

Table of Contents

Preface	viii
Artificial Life in a Challenged World	1
<i>Harold Fellermann, Jaume Bacardit, Ángel Goñi-Moreno and Rudolf M Füchslin</i>	
<hr/> Short Abstracts of Keynote Presentations <hr/>	
Biology on sample size of more than one	4
<i>Kate Adamala</i>	
Financial Networks in the Brave New World.....	5
<i>Stefano Battiston</i>	
Symbiotic Simulations for Decision Support	6
<i>Armando Geller</i>	
Microbial Life for Robotics – towards artificial life	7
<i>Ioannis Ieropoulos</i>	
Artificial Life and the Human Predicament	8
<i>Barry McMullin</i>	
Engineering Plants for Farming and Pharming	9
<i>Nicola Patron</i>	
On the interplay of self-replication and self-reproduction in protocells.....	10
<i>Roberto Serra</i>	
<hr/> Special session: Artificial Life and Society <hr/>	
Artificial Democratic Life. Re-engineering the autonomy of the social, a research program	11
<i>Xabier E. Barandiaran</i>	
Can Bio-Inspired Swarm Algorithms Scale to Modern Societal Problems?.....	13
<i>Darren Chitty, Elizabeth Wanner, Rakhi Parmar and Peter Lewis</i>	
Towards Low-Carbon Conferencing: Acceptance of Virtual Conferencing Solutions and Other Sustainability Measures in the ALIFE Community	21
<i>Harold Fellermann, Alex Penn, Rudolf Füchslin, Jaume Bacardit and Angel Goni Moreno</i>	
Suppleness and Open-Endedness for Social Sustainability	28
<i>Hiroki Sayama</i>	
Cities, a conceptual framing for a synthetic perspective.....	30
<i>Jesús M. Siqueiros-García</i>	
<hr/> Special session : Hybrid Life <hr/>	
Quantifying affordances through information theory	32
<i>Miguel Aguilera, Iñigo Arandia-Romero and Manuel Heras-Escribano</i>	
The dark room problem in predictive processing and active inference, a legacy of cognitivism?	40
<i>Manuel Baltieri and Christopher Buckley</i>	
NukaBot: Research and Design of a Human-Microbe Interaction model.....	48
<i>Dominique Chen, Hiraku Ogura and Young Ah Seong</i>	

Hybrid Synthetic Approach to Animal Interaction	50
<i>Hiroyuki Iizuka, Yosuke Nakamoto and Masahito Yamamoto</i>	
An artificial life approach to studying niche differentiation in soundscape ecology	52
<i>David Kadish, Sebastian Risi and Laura Beloff</i>	
Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network	60
<i>James Knight, Daniil Sakhapov, Norbert Domcsek, Alex Dewar, Paul Graham, Thomas Nowotny and Andrew Philippides</i>	
Hybrid Variational Predictive Coding as a Bridge between Human and Artificial Cognition	68
<i>André Ofner and Sebastian Stober</i>	
Sensorimotor contingency modulates visual awareness of virtual 3D objects	70
<i>Keisuke Suzuki, David Schwartzman, Rafael Augusto and Anil Seth</i>	
<hr/> Special session: Autonomous evolution, production and learning in robotic eco-systems <hr/>	
Heuristics as Decision-making Habits of Autonomous Sensorimotor Agents	72
<i>Erasmus Batta and Christopher Stephens</i>	
From embodied interaction to compositional referential communication: A minimal agent-based model without dedicated communication channels	79
<i>Jorge I. Campos and Tom Froese</i>	
Void Reduction in Self-Healing Swarms	87
<i>Neil Eliot, David Kendall, Alun Moon, Michael Brockway and Martyn Amos</i>	
The ARE Robot Fabricator: How to (Re)produce Robots that Can Evolve in the Real World	95
<i>Matthew Hale, Edgar Buchanan, Alan Winfield, Jon Timmis, Emma Hart, Agoston Eiben, Mike Angus, Frank Veenstra, Wei Li, Robert Woolley, Matteo De Carlo and Andy Tyrrell</i>	
Reinforcement Learning Agents acquire Flocking and Symbiotic Behaviour in Simulated Ecosystems	103
<i>Peter Sunehag, Guy Lever, Siqi Liu, Josh Merel, Nicolas Heess, Joel Leibo, Edward Hughes, Tom Eccles and Thore Graepel</i>	
<hr/> Artificial Chemistry <hr/>	
Building a survivable protocell for a corrosive digital environment	111
<i>David H. Ackley</i>	
Implementing MetaChem framework in Object Orientation	119
<i>Penelope Faulkner Rainford, Angelika Sebald and Susan Stepney</i>	
A Meta-Atom Based Sub-Symbolic Artificial Chemistry	127
<i>Isaac Watson, Angelika Sebald and Susan Stepney</i>	
<hr/> Artificial Life in Social Sciences <hr/>	
Pathways to Good Healthcare Services and Patient Satisfaction: An Evolutionary Game Theoretical Approach	135
<i>Zainab Alalawi, The Anh Han, Yifeng Zeng and Aiman Elragig</i>	
On the Expected Number and Distribution of Equilibria in Multi-player Evolutionary Games	143
<i>Hong Duong and The Anh Han</i>	
Entropy-Based Team Self-Organization with Signal Suppression	145
<i>David King, Lukas Esterle and Gilbert Peterson</i>	
All in Good Team: Optimising Team Personalities for Different Dynamic Problems and Task Types	153
<i>Soo Ling Lim and Peter Bentley</i>	

Only Two Types of Strategies Enforce Linear Payoff Relationships Under Observation Errors in Repeated Prisoner’s Dilemma Games.....	161
<i>Azumi Mamiya and Genki Ichinose</i>	
Emergence of Coordination with Asymmetric Benefits via Prior Commitment.....	163
<i>Bianca Ogbo Ndidi and The Anh Han</i>	
Being a leader or being the leader: The evolution of institutionalised hierarchy	171
<i>Cedric Perret, Emma Hart and Simon Powers</i>	
Generating urban morphologies at large scales.....	179
<i>Juste Raimbault and Julien Perret</i>	
Fairness in Multiplayer Ultimatum Games Through Moderate Responder Selection	187
<i>Fernando Santos and Daan Bloembergen</i>	

Biological Systems

Investigating the Origins of Cancer in the Intestinal Crypt with a Gene Network Agent Based Hybrid Model.....	195
<i>Arturo Araujo, Hanzhao Zhang, Albert Rübber and Peter J Bentley</i>	
A Hexagonal Cell Automaton Model to Imitate Physarum Polycephalum Competitive Behaviour	203
<i>Abubakr Awad, Wei Pang, David Lusseau and George Coghill</i>	
A simplified model of chromatin dynamics drives differentiation process in Boolean models of GRN	211
<i>Michele Braccini, Andrea Roli, Marco Villani, Sara Montagna and Roberto Serra</i>	
A reaction-diffusion model for simulating the oscillatory expansion of biofilms	218
<i>Taishi Mikami, Munehiro Asally, Takeshi Kano and Akio Ishiguro</i>	
Analyzing Evolution of Avian Influenza using detailed Genotypic and Antigenic Models and Phylogenetic Simulation	220
<i>Liam Mosley and Dhananjai Rao</i>	
Anomalies in the Behaviour of a Modularity Inducing Problem Domain.....	228
<i>Zhenyue Qin, Tom Gedeon and Bob McKay</i>	
Optimizing Radiation Therapy Treatments by Exploring Tumour Ecosystem Dynamics in – silico.....	236
<i>Stephan Scheidegger and Harold Fellermann</i>	
Horizontal Gene Transfer Leads to Increased Task Acquisition and Genomic Modularity in Digital Organisms	243
<i>Michael Wisner, Rosangela Canino-Koning and Charles Ofria</i>	

Complex Dynamical Systems

Attractor Landscapes and Information Processing by Convective Obstacle Flows.....	245
<i>Stuart Bartlett and Yuk Yung</i>	
On Sexual Selection in the Presence of Multiple Costly Displays.....	247
<i>Clifford Bohm, Acacia Ackles, Charles Ofria and Arend Hintze</i>	
Steering the Growth of Adaptive Self-Preserving Dissipative Structures.....	255
<i>Matthew Egbert, Yan Kozhitskiy and Nathaniel Virgo</i>	
Complex Systems and Artificial Life: A Decade’s Overview.....	263
<i>Thomas McAtee and Claudia Szabo</i>	
Inferring a Graph’s Topology from Games Played on It.....	271
<i>Douglas Moore and Sara Imari Walker</i>	

Inferring Swarm Models Using a Single Monitoring Robot	278
<i>Suppanat Ruangdech, Martin Homer and Sabine Hauert</i>	
Convolutional Neural Networks for Cellular Automata Classification	280
<i>Eric Silverman</i>	
Biochemical networks display universal structure across projections and levels of organization	282
<i>Harrison Smith, Hyunju Kim and Sara Walker</i>	
Toward the Self-Organisation of Emergency Response Teams Based on Morphogenetic Network Growth	284
<i>Nicolas Toussaint, Emma Norling and René Doursat</i>	
Towards Complex Artificial Life	292
<i>Lance Williams</i>	
<hr/>	
Emergence of Innovation and Cooperation	
<hr/>	
Environmental Harshness and Fitness Improving Innovations	300
<i>Peter Andras</i>	
Reproductive division of labor in a colony of artificial ants	308
<i>Peter Bae and Chris Marriott</i>	
Exogenous Rewards for Promoting Cooperation in Scale-Free Networks	316
<i>Theodor Cimpanu, The Anh Han and Francisco C. Santos</i>	
Simplified Modeling of the Evolution of Skills in a Spatially Resolved Environment	324
<i>Rudolf M. Füchslin, Johannes J. Schneider, Thomas Ott and Richard Walker</i>	
Promoting Cooperation through External Interference	331
<i>The Anh Han, Long Tran-Thanh, Simon Lynch, Theodor Cimpanu and Francisco C. Santos</i>	
Factors that Affect the Evolution of Complex Cooperative Behavior	333
<i>Padmini Rajagopalan, Kay Holekamp and Risto Mäkkiläinen</i>	
<hr/>	
Evolution	
<hr/>	
Towards modelling social habits: an organismically inspired evolutionary robotics approach	341
<i>Manuel G Bedia, Manuel Heras-Escribano, Diego Cajal, Miguel Aguilera and Xabier E. Barandiaran</i>	
Speciation under changing environments	349
<i>Kevin Dubois, Sylvain Cussat-Blanc and Yves Duthen</i>	
To evolve or not to evolve? That is the question	357
<i>Alex Ellery and Gusz Eiben</i>	
Evolutionary rates of information gain and decay in fluctuating environments	365
<i>Nicholas Guttenberg</i>	
Modulating Interaction Times in an Artificial Society of Robots	372
<i>Yara Khaluf and Heiko Hamann</i>	
Acclivation of Virtual Fitness Landscapes	380
<i>Ben Kovitz, David Bender and Marcela Poffald</i>	
Continuous Long-Term Evolution of Genetic Programming	388
<i>W B Langdon and W Banzhaf</i>	
The impact of environmental history on evolved robot properties	396
<i>Karine Miras and A.E. Eiben</i>	
The Cost of Big Brains in Groups	404
<i>Geoff Nitschke, Alex Furman and Danielle Nagar</i>	

Graph Product Representation of Organism-Environment Couplings in Evolution 412
Hiroki Sayama

Maximization of Transfer Entropy leads to Evolution of Functional Differentiation of Swarms 414
Hiromitsu Suganuma, Yuji Kawai, Jihoon Park and Minoru Asada

Neural Networks

Evolving Recurrent Neural Network Controllers by Incremental Fitness Shaping 416
Kaan Akinci and Andy Philippides

Self-Replication in Neural Networks 424
Thomas Gabor, Steffen Illium, Andy Mattausch, Lenz Belzner and Claudia Linnhoff-Popien

The role of ambient noise in the evolution of robust mental representations in cognitive systems 432
Douglas Kirkpatrick and Arend Hintze

Probabilistic Program Neurogenesis 440
Charles Martin and Praveen Pilly

Self-optimization in a Hopfield neural network based on the *C. elegans* connectome 448
Alejandro Ehecatl Morales Huitrón and Tom Froese

DNN Architecture for High Performance Prediction on Natural Videos Loses Submodule's Ability to Learn Discrete-World Dataset 454
Lana Sinapayen and Atsushi Noda

Evolution of metamemory ability by artificial neural networks with neuromodulation 461
Yusuke Yamato, Reiji Suzuki and Takaya Arita

Perception

Robotic models of obstacle avoidance in bats: assessing the benefit of acoustic gaze scanning in complex environments 463
Carl Bou Mansour, Elijah Koreman, Dennis Laurijssen, Jan Steckel, Herbert Peremans and Dieter Vanderelst

A-life Evolution with Human Proxies 465
Vadim Bulitko, Kacy Doucet, Daniel Evans, Hope Docking, Mac Walters, Marilene Oliver, Julian Chow, Shelby Carleton and Natali Kendal-Freedman

Stability of Cooperation in Societies of Emotional and Moody Agents 467
Joe Collenette, Katie Atkinson, Daan Bloembergen and Karl Tuyls

Effects of Visual Sensory Range on the Emergence of Cognition in Early Terrestrial Vertebrates: An Agent-Based Modeling Approach 475
Can Gurkan, Leif Rasmussen and Uri Wilensky

When is an action caused from within? Quantifying the causal chain leading to actions in simulated agents 477
Bjørn Erik Juel, Renzo Comolatti, Giulio Tononi and Larissa Albantakis

Measuring properties of movement in populations of evolved 3D agents 485
Maciej Komosinski and Konrad Miazga

Philosophy, Language, Art and Education

Edge of Chaos: Artificial Life based interactive art installation 493
Vasilija Abramovic and Ruairi Glynn

Autopoiesis in digital learning design: Theoretical implications in education 495
Claudio Aguayo

How to reduce a genome? ALife as a tool to teach the scientific method to school pupils	497
<i>Quentin Carde, Marco Foley, Carole Knibbe, David P. Parsons, Jonathan Rouzaud-Cornabas and Guillaume Beslon</i>	
Gathering of the Hive: Investigating the clustering behaviour of honeybees through art and swarm robotics	505
<i>Haakon Haraldsen Roen, Vako Vartkes Varankian, Stefano Nichele and Kristin Bergaust</i>	
Data Standards for Artificial Life Software	507
<i>Alexander Lalejini, Emily Dolson, Clifford Bohm, Austin J. Ferguson, David P. Parsons, Penelope Faulkner Rainford, Paul Richmond and Charles Ofria</i>	
Organic Selection and Social Heredity: The Original Baldwin Effect Revisited	515
<i>Nam Le</i>	
Morphogenetic Vase Forms	523
<i>Andy Lomas</i>	
Spatial Representation of Self and Other by Superposition Neural Network Model	531
<i>Wataru Noguchi, Hiroyuki Izuka, Shigeru Taguchi and Masahito Yamamoto</i>	
Artificial Ant Colonies for Adaptive Rewards in Serious Games	533
<i>Yann Semet, Bruno Marcon, Konstantinos Demestichas, Nikos Koutsouris and Antonio Ascolese</i>	
The Effects of Individual and Social Learning on the Evolution of Cognitive and Communicative Aspects of Language Abilities	541
<i>Hiroto Yonenoh, Reiji Suzuki and Takaya Arita</i>	
<hr/>	
Robot Control	
<hr/>	
Neuroevolution of Humanoids that Walk Further and Faster with Robust Gaits	543
<i>Ben Jackson and Alastair Channon</i>	
The Limits of Lexicase Selection in an Evolutionary Robotics Task	551
<i>Jared Moore and Adam Stanton</i>	
Improve Quadrupedal Locomotion with Actuated or Passive Joints?	559
<i>Jared Moore and Anthony Clark</i>	
Ego-Noise Predictions for Echolocation in Wheeled Robots	567
<i>Antonio Pico Villalpando, Guido Schillaci, Verena Hafner and Bruno Lara</i>	
Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots	574
<i>Jacopo Talamini, Eric Medvet, Alberto Bartoli and Andrea De Lorenzo</i>	
Different Forms of Random Motor Activity Scaffold the Formation of Different Habits in a Simulated Robot	582
<i>Mario Zarco and Matthew Egbert</i>	
<hr/>	
Swarm Behaviour	
<hr/>	
Guiding aggregation dynamics in a swarm of agents via informed individuals: an analytical study	590
<i>Yannick Gillet, Eliseo Ferrante, Ziya Firat and Elio Tuci</i>	
Emergent Escape-based Flocking Behavior using Multi-Agent Reinforcement Learning	598
<i>Carsten Hahn, Thomy Phan, Thomas Gabor, Lenz Belzner and Claudia Linnhoff-Popien</i>	
Engineering Application of Non-Reciprocal-Interaction-Based (NRIB) Model: Swarm Robotic System That Can Perform Spatially Distributed Tasks in Parallel	606
<i>Takeshi Kano, Eiichi Naito, Takenobu Aoshima and Akio Ishiguro</i>	

Extended Artificial Pheromone System for Swarm Robotic Applications	608
<i>Seongin Na, Mohsen Raoufi, Ali Emre Turgut, Tomas Krajník and Farshad Arvin</i>	
Applying Social Network Analysis to Agent-Based Models: A Case Study of Task Allocation in Swarm Robotics Inspired by Ant Foraging Behavior	616
<i>Georgina Montserrat Reséndiz-Benhumea, Tom Froese, Gabriel Ramos-Fernández and Sandra E. Smith-Aguilar</i>	
On information-optimal scripting of actions	624
<i>Bente Riegler and Daniel Polani</i>	
Modeling Fast and Robust Ant Nest Relocation using Particle Swarm Optimization	626
<i>Hideyasu Sasaki</i>	
Collective Event Detection Using Bio-inspired Minimalistic Communication in a Swarm of Underwater Robots	634
<i>Joshua Cherian Varughese, Hannes Hornischer, Ronald Thenius, Franz Wotawa and Thomas Schmickl</i>	
Collective Change Detection: Adaptivity to Dynamic Swarm Densities and Light Conditions in Robot Swarms	642
<i>Mostafa Wahby, Julian Petzold, Catriona Eschke, Thomas Schmickl and Heiko Hamann</i>	
<hr/> Wet Artificial Life and Synthetic Biology <hr/>	
Droplet based synthetic biology: chemotaxis and interface with biology	650
<i>Silvia Holler and Martin Michael Hanczyc</i>	
Autocatalysis in a Hierarchically Organized Inorganic Chemical Network	652
<i>Cole Mathis, Haralampos Miras and Leroy Cronin</i>	
Synthetic Biology in the Brain: A Vision of Organic Robots	654
<i>Ithai Rabinowitch</i>	
Periodic collective behaviors of organic solvent droplets on the surface of aqueous surfactant solutions .	656
<i>Shinpei Tanaka</i>	
A Candidate Self-Propagating System Enriched by Chemical Ecosystem Selection	658
<i>Lena Vincent, H. James Cleaves and David Baum</i>	

Preface

This volume presents the proceedings of the *2019 Conference on Artificial Life* (ALIFE 2019) which took place 29 July – 2 August 2019 in Newcastle upon Tyne, United Kingdom (<https://2019.alife.org/>).

The International Conference on the Synthesis and Simulation of Living Systems (ALIFE) and the European Conference on Artificial Life (ECAL) have been the major meeting of the artificial life (ALife) research community since 1987 and 1991, respectively. Starting with ALIFE 2018 in Tokyo, Japan, these two conferences have merged into the Conference on Artificial Life. This year, the merged ALIFE conference for the first time replaces what would have been an ECAL conference previously. Hosting it in Europe seemed fitting.

The ALIFE 2019 Theme

This year’s ALIFE conference features the special theme “How Can Artificial Life Help Solve Societal Challenges?” Artificial Life has historically been regarded by its adversaries as an academic “hobby” with little application to real life. We believe that these days are past, as in fact, our interdisciplinary and constantly self-innovating discipline brings together a set of skills and perspectives with a unique potential to tackle some of the most pressing societal challenges of our times. Indeed, our complex systems analysis methodologies have application across a very broad range of domains and provide alternative tools to extract actionable insight than more traditional analysis methods. This special theme ran through the conference in the shape of keynote presentations and satellite events that apply Artificial Life principles to research on e.g. social dynamics, cultural evolution and societal learning, human behaviour, and smart cities.

The ALIFE 2019 Programme

We received a total of 154 full paper and abstract submissions. All submissions were reviewed by typically three and in some cases two reviewers. Scientific advisors then performed a topic wide metareview to derive acceptance decisions. As a result, we accepted 108 contributions for publication and oral presentations. Submission of posters is still open at the time of writing. Following the successful model of ALIFE 2018, we selected 18 submissions for plenary lightning talks.

Simultaneously to ALIFE 2019, Newcastle University also held the Designer Biology 2019 Conference (<http://designer-biology.org/>), which focusses on the intersection of bioengineering and synthetic biology. With the growing overlap of Wet Artificial Life and Synthetic Biology, we closely aligned these conferences by hosting shared keynote presentations as well as a joint panel discussion and generally encouraged broad interactions among the two audiences.

The conference programme this year included:

- Eight keynote presentations spanning diverse areas of Artificial Life research, many of which chosen with respect to the conference theme
 - Barry McMullin (Dublin City University, Ireland)
 - Alex Penn (University of Surrey, United Kingdom)
 - Armando Geller (Scensei GmbH, Switzerland)
 - Stefano Battiston (University of Zurich, Switzerland)
 - Ioannis Ieropoulos (University of Bristol, United Kingdom)
 - Roberto Serra (University of Modena, Italy)
 - Nicola Patrón (Earlham Institute, United Kingdom), shared with Designer Biology
 - Kate Adamala (University of Minnesota, United States of America), shared with Designer Biology
- Parallel sessions
 - Complex dynamical systems

- Perception
- Neural networks
- Robot control
- Swarm behaviour
- Artificial chemistry
- Evolution
- Biological systems
- Emergence of innovation and cooperation
- ALife in the social sciences
- Philosophy, language, art and education
- Three special sessions
 - ALife & Society (organized by Alex Penn and Jesus Mario Siqueiros García)
 - Hybrid Life II: Approaches to integrate biological, artificial and cognitive systems (organized by Manuel Baltieri, Keisuke Suzuki, Hiroyuki Iizuka, Olaf Witkowski and Lana Sinapayen)
 - Towards autonomous evolution, production and learning in robotic eco-systems (organized by Emma Hart, Andy M. Tyrrell, Jon Timmis, Alan Winfield and Gusz Eiben)
- A dedicated student session
- Nine satellite workshops
 - Evolution of Human Behaviour: Using Theory to Address Societal Challenges (organized by Cedric Perret, James Borg, The Anh Han, Tom Lenaerts and Simon Powers)
 - The Fourth Workshop on Social Learning and Cultural Evolution (organized by Chris Marriott, Peter Andras, James Borg and Simon Powers)
 - Computational Approaches to Social Dynamics – Data, Modeling, Simulation, and Hybrids (organized by Genki Ichinose, Fernando P. Santos, Francisco C. Santos and Hiroki Sayama)
 - International Workshop on Agent-Based Modelling of Human Behaviour (organized by Soo Ling Lim, Peter J. Bentley, JoEllyn Prouty McLaren and Randall S. Peterson)
 - Chemistry and Artificial Life Forms (organized by Jitka Čejková, Geoff Cooperand and Richard Löffler)
 - Life at the Nexus of Microbial & Synthetic Ecology (organized by Alex Penn and Erik Hom)
 - 2nd Development of Neural Networks Workshop (organized by Julian Miller, Sylvain Cussat-Blanc, Dennis Wilson)
 - Process Modeling and Self-Organization: Methods and Applications (organized by Francis Heylighen, Peter Dittrich and Tomas Veloz)
 - Methodology in Artificial Life (organized by Emily Dolson, Iñaki Fernández Pérez, Penny Faulkner Rainford and Arturo Araujo)
- Five tutorials
 - Intelligent Systems for Smart Cities (organized by Enrique Alba)
 - Simulating Complex Systems with FLAME GPU (organized by Paul Richmond and Mozhgan Kabiri Chimeh)
 - Introduction to Avida-ED (organized by Michael Wiser)
 - Introduction to Artificial Gene Regulatory Networks (organized by Sylvain Cussat-Blanc and Wolfgang Banzhaf)
 - Cartesian Genetic Programming (organized by Julian Miller)
- A panel discussion shared with the Designer Biology Conference on how the two disciplines relate themselves to current societal challenges

ALIFE 2019 as a Demonstrator for Sustainable Conferencing

In line with our theme of facing societal challenges, ALIFE 2019 attempted to revisit the way academic conferences are run in order to reduce the ecological impact of this scientific event. Sustainability implications were considered at all stages of the conference organization.

Our primary aim was not to deliver a one-off event that would meet self-assigned sustainability criteria, but rather to engage with the community in order to develop a longer term agenda for the ALIFE community. This was initiated through an online survey that assessed the acceptance of diverse sustainability measures. The survey – results of which are included in these proceedings – not only provided us with information about which measures would be accepted by the community, but also with a plethora of actionable suggestions for our endeavour.

As the major ecological impact of scientific conferencing is associated with travel, we early on took the decision to open the conference up for remote participation by means of video-conferencing solutions. As a result, the majority of ALIFE 2019 presentations have been live streamed and recordings of talks have been made available through the conference website. Viewers of these live streams were given the opportunity to take part in discussions using online platforms (sli.do). Participants were allowed to submit prerecorded video presentations, that were then discussed using telecommunication technology. A similar remote presentation model was offered to poster presenters. To increase the reach of these activities, we encouraged groups at different universities in Japan, USA and Mexico to organize local seminars to act as remote hubs of the ALIFE 2019 conference.

For local delegates our agenda lead to sustainably sourced, plastic free catering, featuring menus that followed recommendations of the recent EAT-Lancet Commission report on “Healthy Diets from Sustainable Food Systems” – primarily reducing the amount of red meat and increasing the proportion of vegetarian food options. Similarly, all conference merchandize was selected to be plastic free, low-waste and made from recycled material where possible.

While it is too early to see how our endeavours might impact the long-term way of working in the ALIFE community, we were very happy to see our own institution engage in this pilot project. Implementations we have trialed at ALIFE 2019 will now be considered by Newcastle University to develop best practices and guidelines for its future sustainable conference organization.

About the Editors

Harold Fellersmann holds a PhD in Applied Systems Science from the University of Osnabrück, Germany, and is currently a Lecturer for Complex Biosystems at the School of Computing, Newcastle University. His research interests include complex systems, systems and synthetic biology, minimal cell research and DNA nanotechnology.

Jaume Bacardit has received a BEng and MEng in Computer Engineering and a PhD in Computer Science from Ramon Llull University, Spain in 1998, 2000 and 2004, respectively. He is currently Reader in Machine Learning at Newcastle University in the UK. Bacardit’s research interests include the development of machine learning methods for large-scale problems, the design of techniques to extract knowledge and improve the interpretability of machine learning algorithms and the application of these methods to a broad range of problems, mostly in biomedical domains.

Ángel Goñi-Moreno is a Lecturer in Synthetic Biology at Newcastle University (UK) – as part of the Interdisciplinary Computing and Complex Biosystems (ICOS) group. He studied Computer Engineering at the Technical University of Madrid (UPM), Spain, from which he also earned a MSc in Artificial Intelligence and a PhD in Synthetic Biology. After UPM (2010) he joined Martyn Amos’ group at Manchester Metropolitan University (UK) as a postdoc; then moved back to Madrid (2013) to join the Victor de Lorenzo’s lab at the National Center for Biotechnology (CNB-CSIC). In 2016 he moved to the UK and took his position at Newcastle, in which he leads an interdisciplinary team that carries out both theoretical and experimental research in synthetic and systems biology.

Rudolf Fuchsli holds a PhD in Physics and is a Professor of Applied Complex Systems Sciences at the School of Engineering, Zurich University for Applied Sciences. His research interests include applications of complex systems sciences to medicine, social systems and evolutionary engineering. He is co-director of the European Centre for Living Technology in Venice, Italy.

Acknowledgments

ALIFE 2019 would not have been possible without the help of many people. We would like to thank Prof. Julie Sanders, Deputy Vice-Chancellor of Newcastle University, for her continued support of our sustainability agenda. The University's audio-visual team and technical support team, namely Colin Fahey and John Snowdon, are acknowledged for their support in realizing our complex AV requirements. Kelly Ramsay from the University's conference team was a big help for all local organization. We thank Ana Vásquez for taking the lead on all graphic and web design for ALIFE 2019 and Pawel Widera for his continuous support of the conference server.

Scientific Advisory Committee

Peter Andras	Keele University
Will N Browne	Victoria University of Wellington
Jitka Čejková	University of Chemistry and Technology Prague
Martyn Dade-Robertson	Newcastle University
Pietro Speroni di Fenizio	independent scientist
Tom Froese	National Autonomous University of Mexico
Emma Hart	Edinburgh Napier University
Inman Harvey	University of Sussex
Nathan Labhart	University of Zurich
Alex Penn	University of Surrey
Sergi Valverde	Universidad Pompeu Fabra
Olaf Witkowski	Tokyo Institute of Technology

Programme Committee

Eric Aaron	Nicholas Guttenberg	Sebastian Risi
Dave Ackley	Heiko Hamann	Andrea Roli
Andy Adamatzky	Martin Hanczyk	Jonathan Rouzaud-Cornabas
Miguel Aguilera	Taichi Haruna	Bruno A. Santos
Carlos Alquézar	Inman Harvey	Hiroki Sayama
Martyn Amos	Yasuhiro Hashimoto	Guido Schillaci
Peter Andras	Salima Hassas	Thomas Schmickl
Takaya Arita	Helmut Hauser	William Sellers
Nathanael Aubert-Kato	Arend Hintze	Tomohiro Shirakawa
Joshua Auerbach	Thomas Hinze	Ben Shirt-Ediss
Jaume Bacardit	Julien Hubert	Eric Silverman
Manuel Baltieri	Cristian Huepe	Lana Sinapayen
Wolfgang Banzhaf	Paul Humphreys	J. Mario Siqueiros-García
Stuart Bartlett	Tim Hutton	Pietro Speroni di Fenizio
Randall Beer	Hiroyuki Iizuka	Michael Spranger
Guillaume Beslon	Eduardo Izquierdo	Pasquale Stano
Martin Biehl	Klaus Jaffe	Adam Stanton
Tim Blackwell	Colin Johnson	Kathleen Steinhofel
Christian Blum	Genaro Juarez Martinez	Susan Stepney
Josh Bongard	Yoshihiko Kayama	Keisuke Suzuki
James M. Borg	Hiroki Kojima	Reiji Suzuki
Amine Boumaza	Sam Kriegman	Yasuhiro Suzuki
Markus Brede	Alexander Lalejini	Adam Svahn
David Breen	Mathieu Lefort	Masanori Takano

Christopher Buckley
Timoteo Carletti
José M Cecilia
Jitka Čejková
Alastair Channon
Sung-Bae Cho
Insook Choi
Luís Correia
Sylvain Cussat-Blanc
Luisa Damiano
Thomas Dandekar
Peter Dittrich
Gordana Dodig Crnkovic
Emily Dolson
Matthew Egbert
Arantza Etxeberria
Harold Fellermann
Christoph Flamm
Tom Froese
Rudolf Marcel Fuchslin
Carlos Gershenson
Keyan Ghazi-Zahedi
Jean-Louis Giavitto
Kyrre Glette
Ángel Goñi-Moreno
Sean Grimes

Soo Ling Lim
Tiong Hoo Lim
Taivo Lints
Joseph Lizier
Daniel Lobo
Herve Luga
George Magoulas
Chris Marriott
Philip Mckinley
JJ Merelo
Stefan Meyer
Dusan Misevic
Youichiro Miyake
Stefano Nichele
Geoff Nitschke
Stefano Nolfi
Erfan Noury Qarajalar
Charles Ofria
Yuta Ogai
Mizuki Oka
Joshua Payne
Alexandra Penn
Andrew Philippides
Fiona Polack
Simon Powers
Mikhail Prokopenko

Ivan Tanev
Tim Taylor
Jason Teo
Christof Teuscher
Guy Theraulaz
Jim Torresen
Vito Trianni
Jon Umerez
Tatsuo Unemi
Neil Vaughan
Mario Villalobos
Marco Villani
Sebastian von Mammen
Anyá Vostinar
Yifei Wang
James Whiting
Michael Wisner
Lance Williams
Michael Wisner
Olaf Witkowski
Borys Wrobel
Luis Zaman
Mario Zarco
Hector Zenil
Xiaoge Zhang

The ALIFE 2019 Organizing Committee

Harold Fellermann
(General Chair)
Rudolf Fuchslin
(Programme Chair)
Ángel Goñi-Moreno
(Sustainability & Submissions)
Jaume Bacardit
(Proceedings Chair)
Ben Shirt-Ediss
(Technical Chair)
Andrew Lawson
(Local Chair & Communications)
James McCull
(Local Organization)
Ann Kirtley
(Local Organization)

Artificial Life in a Challenged World

Harold Fellermann^{1,*}, Jaume Bacardit¹, Angel Goñi-Moreno¹ and Rudolf M. Fuchslin²

¹School of Computing, Newcastle University, Newcastle Upon Tyne, UK

²School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland

*harold.fellermann@newcastle.ac.uk

“Life clearly does more than adapt to the Earth. It changes the Earth to its own purposes. Evolution is a tightly coupled dance, with life and the material environment as partners. From the dance emerges the entity Gaia.”

James Lovelock

Life in a Challenged World

Our world is in distress. At an alarmingly increasing pace, we are witnessing the effects of global warming and climate change: record temperatures, droughts, melting of glacial and polar ice and large scale natural disasters are only few of the many clear indicators of the burden that human life puts on our Planet’s climate systems (Masson-Delmotte et al., 2018). Rapid climate change and the loss of natural habitats to make space for humanity’s growing resource needs are causing biological mass extinctions of unprecedented scale (Ceballos and Ehrlich, 2018). With a global human population size predicted to plateau at around 11bn people within the next 50 to 80 years (United Nations, Department of Economic and Social Affairs, Population Division, 2017) and a growing number of threshold countries adopting consumerist life styles, we can only expect the strain on our ecosystem to intensify.

Global challenges do not stop at our door steps. Despite huge gains in global economic output, there is evidence that our current social, political and economic systems are exacerbating inequalities, rather than reducing them (Dabla-Norris et al., 2015), which is an increasing cause of political unrest. The global financial crisis of 2008 revealed significant weaknesses in the financial system and the vulnerabilities of a single interconnected global market (Rose and Spiegel, 2012; Battiston et al., 2016). The scale of the employment challenge is vast, with an estimated 200 million people unemployed globally. Even comparably modest societal goals such as race and gender equality are far from being achieved.

For the majority of humans, the Anthropocene might soon no longer be a comfortable spot to live in. While some au-

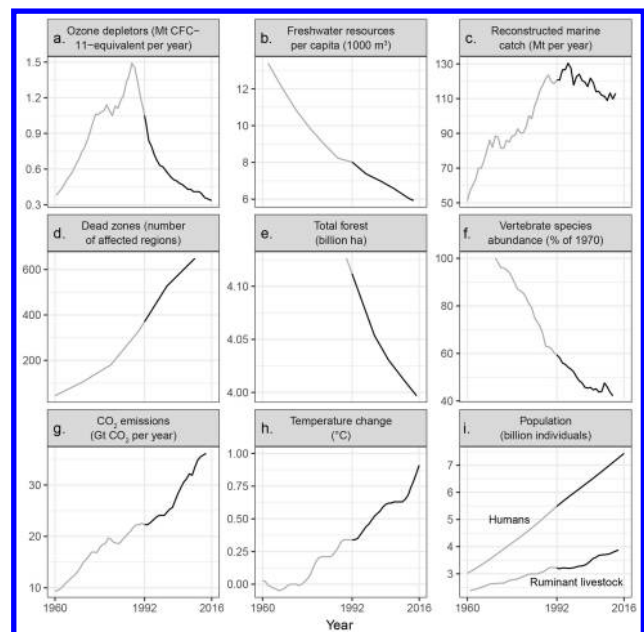


Figure 1: Trends of diverse environmental indicators over the last 50 years. None has reversed since the “World Scientists’ Warning to Humanity” had been issued in 1992. Yet, the reversal of ozone depletor concentrations (panel a) is a proof by example that concerted policy making can in principle overcome ecological problems on a global scale. Reproduced with permission from Ripple et al. (2017).

thors raise the question whether a collapse of our global civilization can be avoided (Ehrlich and Ehrlich, 2013), other scientists worry whether humanity will inevitably push Earth beyond limits where it would be unable to sustain life in the manner that we know at all (Ripple et al., 2017).

Can Artificial Life Help Tackle Societal Challenges?

Historically, the discipline of Artificial Life has often been regarded by its adversaries as an academic leisure, where scientists investigate dreamt up worlds with little to no ap-

plication to real life. We believe this view to be a gross misconception that our community should set straight: Artificial Life is the study of Life driven by investigating its first principles. How could we possibly remain unattached or shy away, if the subject at the very heart of our studies is at stake? Yet, it is not immediately obvious how our community can confront the challenges that humanity is currently facing. What unique skills and perspectives can the Artificial Life community bring to the table?

As organizers of an ALIFE conference on the theme “How Can Artificial Life Help Solve Societal Challenges” we do in fact believe that our interdisciplinary and constantly self-innovating discipline brings together a set of skills and perspectives with a unique potential to tackle some of the most pressing societal challenges of our times.

Firstly, we believe that more than many other disciplines, ALIFE embraces systems thinking at its core. Rather than perceiving our current societal challenges as isolated problems that can be solved independently using linear thinking, we recognize them as interconnected symptoms of a strained system of systems that about to be pushed out of their local area of stability. The formal application of systems thinking to global challenges dates almost 50 years back, when Meadows et al. (1972) developed Systems Dynamics to investigate the interplay of feedback loops in our intertwined socio-economical and ecological systems. Their basic conclusion – that humanity inevitably reaches the carrying capacity of planet Earth – still holds to date, and has been restated with slightly updated facts ever since.

However, such nonlinearities do not come easy to our human minds, and conclusions from these studies have not yet affected policy making and societal norms and habits to the extent necessary. We believe one of the core competencies of ALIFE – mathematical modelling and simulation – to be a vital tool, for understanding the nonlinear nature of our current crisis, as well as for educating the public and policy makers. Model based decision support systems and participatory modelling techniques are key elements to this process (Gilbert and Bullock, 2014; Sayama and Dionne, 2015).

Importantly, the predicament we are facing is not just ecological in nature, nor societal, nor economic, nor bound to any other single discipline – it is truly trans-disciplinary. Few other scientific communities have interdisciplinarity embodied as deeply and drastically as ours (Aicardi, 2010): not only do we engage without reservations in discourses with academics from various departments and faculties, ALIFE also has close bonds with the arts, music, game development and design. Maybe we should reflect on this strength of ours, and provide more domain-segmented academic research areas with our 30 year experience in crossing disciplinary boundaries (Dini et al., 2011).

Thirdly, we believe that ALIFE’s ‘central dogma’ that life is an emergent complex adaptive system that exhibits dynamical hierarchies and open-ended evolution is another

indispensable ingredient when addressing our current problems. The ALIFE community is aware of the complex dynamics that unfold in self-organized, self-regulating multi-agent societies. We know that neither biological systems, societies, nor markets react proportional in response to stimuli and are thus notoriously hard to steer by means of top-down regulations. Yet, ALIFE understands that sudden phase transitions are not only found in our current climate crisis, but that social learning and cultural revolutions equally proceed in series of punctuated equilibria that can open up for unexpected jumps into the next adjacent possible.

The industrialized world currently experiences the rapid growth of a new cultural grassroots movement that turns its back on established consumerism to instead adopt a low-carbon, zero-waste life-style. Such movements, paired with the loud voices of young adults striking all over the world for policy makers to finally take actions, might ignite a spark of hope among the desperate predictions that scientists have produced. But what determines the success or failure of such cultural revolutions? Which factors most strongly affect their speed of uptake? What influences how such movements ultimately shape national and global policy making? We believe that these and similar questions can be readily addressed and turned into actionable recommendations using our ALIFE toolbox. At the same time, we can reach out to these groups from a scientific point of view.

The sheer amount of societal themes discussed at this year’s conference – from social learning, human and cultural evolution, social dynamics and simulation to financial market dynamics and policy making – demonstrates how heavily our community is indeed long concerned with this general agenda. And what better tools to draw from than the union of dynamical systems theory, game theory and agent-based modelling, artificial intelligence and machine learning, evolutionary and unconventional computing that together constitute a good part of the ALIFE methodology?

A main concept of ALIFE is not the study of life as it is (with all its evolved facets and structures that sometimes only make sense in the context of this evolutionary history) but the analysis of deliberately simplified life-like systems. Two main intellectual challenges have to be mastered: First, principles of life found in our real world have to be mapped into artificial systems. Thereby, depending on the research question, only a small subset of important principles may be taken into account in order to highlight specific aspects. Second, we have to answer to what extent results found in artificial systems can be mapped onto the behavior or structures of the real world. The first challenge certainly appeals to the creativity of the ALIFE community and resulted in a rich variety of approaches. The second challenge turns out to be harder and certainly requires an even closer collaboration with those experts who know the real world. However, ALIFE will only achieve its full potential if both challenges are mastered.

Perhaps most importantly, compared to many other academic disciplines, the ALIFE community is a notoriously free-spirited crowd. In our studies, we typically do not let hair-splitting details stop us from pushing pie-in-the-sky ideas. This capability of ALIFE to be imaginative and to think out of the box might perhaps be the most important quality that we can offer to the scientific community and society at large, by embedding our critical creative thinking and our free spirit in all potential solution fronts. Be it in the form of alternative societies and e-democracy (Speroni di Fenizio and Paterson, 2010; Aragón et al., 2017), living technology that incorporates the core features of life (Bedau et al., 2009; Armstrong, 2009), bioremediation and terraforming programmes based on artificial cells (Solé, 2015), or even a coming form of hybrid, biological-technical life forms that could form the basis of a post-human era.

Artificial Life is understood as “*The study of life as it could be, rather than life as we know it.*” At times where life as we know it is threatened to cease to exist, we urge the ALIFE community to take this credo to the next level: Let us dream up what our very own life as human species could be, and how we might be able to recreate the harmonious dance that life and the material environment ought to perform.

References

- Aicardi, C. (2010). *Harnessing non-modernity: a case study in artificial life*. Doctoral, UCL (University College London).
- Aragón, P., Kaltenbrunner, A., Calleja-López, A., Pereira, A., Monterde, A., Barandiaran, X. E., and Gómez, V. (2017). Deliberative Platform Design: The Case Study of the Online Discussions in Decidim Barcelona. In Ciampaglia, G. L., Mashhadi, A., and Yasseri, T., editors, *Social Informatics*, Lecture Notes in Computer Science, pages 277–287. Springer International Publishing.
- Armstrong, R. (2009). Systems Architecture: A New Model for Sustainability and the Built Environment using Nanotechnology, Biotechnology, Information Technology, and Cognitive Science with Living Technology. *Artificial Life*, 16(1):73–87.
- Battiston, S., Farmer, J. D., Flache, A., Garlaschelli, D., Haldane, A. G., Heesterbeek, H., Hommes, C., Jaeger, C., May, R., and Scheffer, M. (2016). Complexity theory and financial regulation. *Science*, 351(6275):818–819.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., and Rasmussen, S. (2009). Living Technology: Exploiting Life’s Principles in Technology. *Artificial Life*, 16(1):89–97.
- Ceballos, G. and Ehrlich, P. R. (2018). The misunderstood sixth mass extinction. *Science*, 360(6393):1080–1081.
- Dabla-Norris, E., Kochhar, K., Suphaphiphat, N., Ricka, F., and Tsounta, E. (2015). Causes and Consequences of Income Inequality : A Global Perspective. Technical Report SD-NEA201513, International Monetary Fund.
- Dini, P., Iqani, M., and Mansell, R. (2011). The (Im)possibility of Interdisciplinarity: Lessons from Constructing a Theoretical Framework for Digital Ecosystems. *Culture, Theory and Critique*, 52(1):3–27.
- Ehrlich, P. R. and Ehrlich, A. H. (2013). Can a collapse of global civilization be avoided? *Proceedings of the Royal Society B: Biological Sciences*, 280(1754).
- Gilbert, N. and Bullock, S. (2014). Complexity at the social science interface. *Complexity*, 19(6):1–4.
- Masson-Delmotte, V., Zhai, P., Pörtner, H.-O., Roberts, D., Skea, J., Shukla, P.R., Pirani, A., Moufouma-Okia, Péan, C., Pidcock, R., Connors, S., Matthews, J.B.R., Chen, Y., Zhou, X., Gomis, M.I., Lonnoy, E., Maycock, Tignor, M., and Waterfield, T. (2018). *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. World Meteorological Organization, Geneva, Switzerland.
- Meadows, D., Meadows, D., Randers, J., and Behrens III, W. (1972). *The limits to growth*. Potomac Associates.
- Ripple, W. J., Wolf, C., Newsome, T. M., Galetti, M., Alamgir, M., Crist, E., Mahmoud, M. I., and Laurance, W. F. (2017). World Scientists Warning to Humanity: A Second Notice. *BioScience*, 67(12):1026–1028.
- Rose, A. K. and Spiegel, M. M. (2012). Cross-country causes and consequences of the 2008 crisis: Early warning. *Japan and the World Economy*, 24(1):1–16.
- Sayama, H. and Dionne, S. D. (2015). Studying Collective Human Decision Making and Creativity with Evolutionary Computation. *Artificial Life*, 21(3):379–393.
- Solé, R. (2015). Bioengineering the biosphere? *Ecological Complexity*, 22:40–49.
- Speroni di Fenizio, P. and Paterson, D. (2010). Don’t Vote, Evolve! In Tambouris, E., Macintosh, A., and Glassey, O., editors, *Electronic Participation*, Lecture Notes in Computer Science, pages 13–25. Springer Berlin Heidelberg.
- United Nations, Department of Economic and Social Affairs, Population Division (2017). *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables*.

Biology on sample size of more than one

Kate Adamala
University of Minnesota
kadamala@umn.edu

Abstract

All of biological research is done on a single sample: that of modern, terrestrial life. In the quest to engineer synthetic living systems, we seek to expand that sample size, enabling investigation of properties of lineage agnostic, synthetic organisms.

Synthetic minimal cells are liposomal bioreactors that have some, but not all properties of live cells. Creating artificial living systems allows us to diversify the chassis of biological studies and provide novel opportunities for bioengineering. We can begin to answer biggest questions about healthy and diseased natural cells and ask whole new set of questions about the nature of life. Engineering synthetic cells with fundamentally different physical and chemical properties, we can compare behaviors and begin drawing broad conclusions about basic rules of biological life.

Synthetic cells are fully definable, enabling studies of natural processes with level of detail previously unavailable. In synthetic system, there is less noise from underlying endogenous activity of the cell, and every interesting process can be isolated and studied independently. Synthetic cells provide new chassis for biological studies, for broadening understanding of our own type of biology, and for investigating alternatives to the single known life form.

Financial Networks in the Brave New World

Stefano Battiston
University of Zurich
stefano.battiston@uzh.ch

Abstract

There is growing academic and societal interest on the relation between the surge of AI and the future of work and services. Some experts project utopian visions where AI enhance human life and helping us to achieve a more sustainable world. Others project dystopian visions where AI dominates human life suppressing freedom and democracy.

These discussions do not adequately account for the crucial role played by the structure of the ownership relationships. Who will own the robots? Who will own the physical devices and the technologies to make them work?

In particular, developed economies have committed to move towards a more sustainable and circular production system. This process could greatly benefit from AI but may require a transformation of the notion of ownership. Consumers may have to purchase services rather than owning goods (i.e. the right to use a car rather than owning the car itself).

Ownership takes place through chains of contracts, which give rise to complex financial networks. In one direction, financial funds flows to make possible the realisation of projects and technologies. In the other direction, revenues, control but also financial risk flow towards the owners. Understanding the properties of financial networks is key to understand and manage the trajectory that human society we will take in the next decades in relation to the surge of AI.

Symbiotic Simulations for Decision Support

Armando Geller

Scensei (Switzerland) GmbH, Zeltweg 74, 8032 Zurich, Switzerland
armando@trovero.io

Abstract

Our socio-technical environment is becoming increasingly complex all the while we have less and less time for making decisions. The advent of computers in general and AI in particular has helped us to deal with complexity and time constraints. Virtually no industry and sector remains untouched by this development: The private sector, government, non-profits; manufacturing, advertisement, logistics, healthcare, defense; finance, compliance, customer care, human resources and many more. Why is it then that, I wonder, we still have very little satisficing solutions to our worlds most pressing problems as for example stated in the sustainable development goals: Climate change, poverty, sustainable cities and so forth. One reason for this, I believe, is grounded in the false trust in the abundance of data and ubiquity of computational power. More data and brute force doesnt necessarily mean better insights and thus decisions.

Sustainable and robust decisions depend on our understanding why things happen and our ability to think in plausible futures. Causality and scenarios can be generated through the symbiosis of human brain ingenuity and computational simulations. This implies that humans embrace digitalization as an opportunity and invest in the further development of cognitive humanmachine interaction.

I will suggest and critically discuss a framework for constructing and applying a computational decision support framework that furthers this vision. Technically the framework hinges on data fusion, simulation and insight generation. Successful application for decision-making, however, relies on consequent stakeholder integration, which requires building trust in a simulations underlying causal model from which openness to internalize insights in organizational decision making processes derives. I will present the concrete steps of building a computational simulation designed for decision support against the background of two case studies representing pressing social problems, along with the process of taking the client on this journey and how it helped her to improve business critical decisions and thus outcomes.

The quintessence is as simple as it remains futuresque: Tomorrows successful organization represents itself and the environment it operates in in a form of holodeck, enabling it and its employees to play through and train for the future that will challenges us.

Microbial Life for Robotics towards artificial life

Ioannis Ieropoulos

Bristol BioEnergy Centre, Bristol Robotics Laboratory,
University of the West of England, Bristol, BS16 1QY, UK
Ioannis2.Ieropoulos@uwe.ac.uk

Abstract

This talk will present results from the practical implementation of MFCs in a range of applications. The presentation will show the chronological evolution of the technology, starting from the earlier implementation in robotics to the more recent development in sanitation and as a robotic chemostat for maintaining steady state conditions in microbial communities. The talk will have a focus on EvoBot, a robotic chemostat that has been developed as part of the EU FP-7 EVOBLISS project (611640), which was funded under the Evolving Living Technologies Programme. The work combined scientific approaches from robotics, artificial intelligence, chemistry, and microbiology and the talk will demonstrate how the integration of these otherwise disparate disciplines was used to produce i) a generally useful, expandable and customizable technical platform for the artificial evolution of new materials and applications based on a real-time feedback robotic workstation and ii) a specific improved technology, namely a microbial fuel cell, that incorporates natural as well as artificial macro-, micro-, and nanoscale elements for improved function. EvoBot was used with the scientific objective to investigate the possibility of optimizing artificial chemical life, microbial ecosystems, and nanoparticles and their physiochemical, dynamic environments using robot facilitated, artificial evolution. The main conceptual synergy of EVOBLISS was to embody the principles of living technology at various scales in order to probe a system's ability to evolve within and between scales. The talk continues with a description of the multiple by-products that can be produced by the core MFC technology and concludes with the case for microbial fuel cells as a platform technology for multiple a range of environments including sanitation, renewable energy generation, production of value-added products via elemental recycling and wastewater treatment.

Artificial Life and the Human Predicament

Barry McMullin
Dublin City University, Ireland
barry.mcmullin@dcu.ie

Abstract

It is well established in the scientific literature that global human civilization faces multiple imminent, and potentially existential, crises. The most comprehensive survey is perhaps that of the *Planetary Boundaries* framework (Rockström et al., 2009; Steffen et al., 2015). The unfolding of these challenges will be very complex, and the trajectory ahead is certainly still open to significant human management and moderation. Nonetheless, it is clear that we are no longer dealing with “problems” that might be “solved”; rather, this is a *predicament* — an uncertain, dynamic, and at least partially chaotic, disruption in global human development (Gilding, 2012). A predicament calls not for “solution”, but for engagement, and continuous refinement of response. The purpose of this contribution is to explore how some specific concepts, tools and techniques of Artificial Life have already helped shape our understanding of this predicament; and may offer some distinctive supports in moulding our future responses.

The use of computational tools to model complex biological, evolutionary, ecological and social dynamics is a foundational technique in the ALife field. Indeed, computational thinking and modelling was at the heart of the *systems dynamics* approach to socio-ecological modelling pioneered by Forrester (1982). This provided the basis for the famous (or infamous?) *Limits to Growth* (LTG) project of the Club of Rome (Meadows et al., 1972). This was the first substantive attempt to computationally model the socio-ecological dynamics of global human society and assess whether ecological impacts would be likely to limit the growth of human material activities within any practically foreseeable timeframe. While the model was necessarily crude, the robust result was that — in the absence of effective control measures to the contrary — serious limits would become apparent within the first half of the 21st century.

In the almost 50 years since its original publication, the world has tracked remarkably close to the “standard run” of the LTG study (Turner, 2014). In fact, multiple lines of investigation now strongly suggest that aggregate human activity has already reached a state of significant *overshoot* beyond safe or sustainable ecological limits. Overshoot is a qualitatively distinct regime for the design and operation of any adaptive or mitigating interventions (Catton, 1982). Effective societal responses to date have been significantly impaired by a lack of wide understanding of this harsh ecological reality. This gap in understanding facilitates the comforting — but erroneous — notion that it is prudent to delay difficult responses until after impacts are manifest. But delay is precisely one of the

principle mechanisms that actually *causes* overshoot, and undermines the capability to damp the subsequent “crash”. This presents both a need and an opportunity for Artificial Life practitioners to use their skills and their tools to help catalyse much wider societal understanding of the nature of ecological overshoot and mediate the desperately needed reflections on how to achieve the necessary collective, systems-level, responses (cf. Bullock, 2016).

Separately and in conclusion, the presentation will briefly consider the meta-question of the ecological footprint of scholarly activity itself: and what, if any, obligations scholarly communities (such as ISAL, the International Society for Artificial Life) might have to reconsider their established practices in the face of planetary scale ecological emergency (e.g. Wilde and Nevins, 2015).

References

- Bullock, S. (2016). Alife as a Model Discipline for Policy-Relevant Simulation Modelling: Might Worse Simulations Fuel a Better Science-Policy Interface? In *Proceedings of the Artificial Life Conference 2016*, pages 28–29, Cancun, Mexico. MIT Press. <http://tinyurl.com/y4z54rjs>
- Catton, W. R. (1982). *Overshoot*. University of Illinois Press. <https://books.google.ie/books?id=jCKXpv-E5HsC>
- Forrester, J. W. (1982). *Principles of Systems*. MIT Press. <https://books.google.ie/books?id=llqaPwAACAAJ>
- Gilding, P. (2012). *The Great Disruption*. Bloomsbury. https://books.google.ie/books?id=ULyEOLm2_ZIC
- Meadows, D. H. et al. (1972). *The Limits to Growth*. Universe Books. <http://goo.gl/XxgxpF>
- Rockström, J. et al. (2009). Planetary boundaries: exploring the safe operating space for humanity. *Ecology and Society*, 14(2): art. 32. <http://goo.gl/A0Q6IB>
- Steffen, W. et al. (2015). Planetary boundaries: Guiding human development on a changing planet. *Science*, 347(6223):1259855. <http://www.sciencemag.org/content/347/6223/1259855>
- Turner, G. (2014). Is global collapse imminent? Technical Report 4, Melbourne Sustainable Society Institute. <http://goo.gl/Gxc7kD>
- Wilde, P. and Nevins, J. (2015). Petition on Flying Less and Reducing Academia’s Carbon Footprint. Technical report. <http://tinyurl.com/j2jxt4a>

Engineering Plants for Farming and Pharming

Nicola J Patron

Engineering Biology, Earlham Institute, Norwich Science Park, Norfolk, NR4 7UZ
nicola.patron@earlham.ac.uk

Abstract

Although unable to flee predators or sub-optimal growth conditions, plants have the incredible ability to continue normal life after losing whole organs. They can also alter the expression-levels of thousands of genes, remodelling growth and metabolism and deploying an extensive molecular armoury in response to threats. These abilities provide us with food but also present a potential platform for the rapid production of complex molecules from water and light. Until recently, however, we lacked the tools and data necessary for complex engineering of plant systems. The application of engineering principles to plant biology has enabled us to establish platforms for high-throughput, automated, experimentation at nanoscales. We are combining these approaches with genome editing technologies and comparative genomics to investigate how regulatory functions are encoded in plant DNA and to engineer plants with new traits and functions. In recent work, we have shown that genome editing can be used to make plants with different carbohydrate structures, paving the way for the production of more nutritious crops. Currently, we are learning where to make precise changes to regulatory regions in order to rewire the control networks that coordinate large-scale responses to environmental signals. Beyond foods, we are developing plants as photosynthetic platforms for biomanufacturing. We are interested not just in making human therapies but also in manufacturing a greater range of products to improve the sustainability of agriculture.

On the interplay of self-replication and self-reproduction in protocells

Roberto Serra

Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia
European Centre for Living Technology (ECLT), Venice
Institute for Advanced Study (IAS), University of Amsterdam
rserra@unimore.it

Abstract

Modern biological cells are endowed with effective mechanisms which control their division, ensuring that it does not take place before the duplication of the genetic material has been completed. It is unlikely that similar sophisticated mechanisms were in place in primitive protocells, which were much simpler than their present-day descendants. So a major question concerns the way in which reproduction of the whole protocell might take place together with replication of its genetic molecules, absent any kind of high-level control.

This might happen if the rate of duplication of the genetic material and that of fission of the protocell are the same, i.e. if the two processes are synchronized. This possibility can be studied using simplified models of reaction networks (among replicators), assuming that one or more replicators can affect the growth and fission rates of their lipid container. Surprisingly enough, such synchronization does not necessarily require a careful assembly of reactions with very specific reaction rates. On the contrary, it turns out to be a property which emerges spontaneously in a broad set of models, with different parameters, different reaction networks and even different protocell architectures. Note that synchronization, while being a widespread property, is not always achieved for all the models and reaction types. The conditions for emergent synchronization will be discussed, reviewing previous work and showing some new results.

These results are based upon dynamical models which assume that the reactions are known a priori. On the other hand, in models of the origin of life it is often assumed that not all the important chemicals are there since the very beginning, but that some of them are synthesized at later stages. The appearance of new chemicals makes new reactions possible, which may in turn lead to the synthesis of new chemicals, etc. Dealing with this kind of problems requires the choice of a particular model of the replicators and of their interactions; in this paper the random binary polymer model proposed by S. Kauffman, where the replicators are polymers which can undergo cleavage or condensation, will be considered. This model allows, in principle, the appearance in time of polymers of increasing length. Another aspect which has to be taken into account, in order to properly model these phenomena, is that new chemical species may be initially present in very low concentrations, which require a stochastic treatment like the one allowed by the well-known Gillespie algorithm.

The random binary polymer model can give rise in time to collectively autocatalytic sets, which are able to self-replicate; if some chemicals which belong to the core or to the periphery of these sets are coupled to the growth of the lipid container, this may lead to emergent synchronization. However, the interactions can be quite complicated and the overall behaviour can be counterintuitive. Some examples of dynamical behaviours which have been observed in simulations will be presented and discussed, with particular emphasis on features which are always, or frequently, observed. It will be argued that studying the dynamical interaction of autocatalytic sets with the growth and splitting dynamics of the lipid container is crucial to understand the possibility that a population of protocells undergo sustainable growth and evolution.

Artificial Democratic Life. Re-engineering the autonomy of the social, a research program

Xabier E. Barandiaran^{1,2}

¹ IAS-Research Center for Life, Mind and, Society, UPV/EHU, University of the Basque Country, Spain.

² Department of Philosophy, Faculty of Labor Relations and Social Work, UPV/EHU, University of the Basque Country, Spain.
xabier.academic@barandiaran.net

Abstract

By *Artificial Democratic Life* we mean the design and deployment of artificial (digital) infrastructures aimed at enhancing or improving social democratic life. Artificial Life, as a discipline and as a community, has much to contribute to the contemporary challenge of redesigning democracy in the network era, in understanding and designing democracy as a form of life: one that evolves into increasingly higher complexity and diversity while preserving homeostatic invariants and designing the infrastructures capable to resiliently enhance it. We identify some opportunities and specific challenges that can be faced using Alife simulation techniques and conceptual resources.

Introduction: democracy and the living

New constraints and opportunities often give rise to the emergence of new forms of life or their radical transformation. Such is the case of the emergence of administrative institutions with the invention of writing, the emergence and autonomization of economic life with capitalism or that of academic life through meetings, letters and journals and, more recently, the internet. This last infrastructure, the internet, has made possible a profound transformation of many human and societal forms of life. And democracy is waiting its turn. In the era of Artificial Intelligence and Algorithmic Governance the issue of how to build public digital infrastructures for democratic life becomes a design challenge that can greatly benefit from Alife simulation techniques and conceptual resources.

The conceptual relationships that researchers have established between living phenomena and democracy are many. At the most abstract level “causal democracy” has been vindicated as a model for understanding developmental and living phenomena (Oyama, 2000), at the molecular level cellular transcription networks have been characterized as democratic dynamics (Bar-Yam et al., 2009), collective behaviour is often characterized using democracy related terms like “quorum” sensing and “consensus” in bacteria (Miller and Bassler, 2001) or directly as “democracy” in honey-bees (Seeley, 2010). In turn, concept such as that of neuronal assemblies (Varela, 1995) or dendritic democracy (Husser,

2001) have been widely used, and as early as in 1941 Sherrington declared the nervous system to be “a million-fold democracy whose each unit is a cell” (Sherrington, 1941). More generally, the concepts of “self-organization” and “autonomy” have been central to the characterizations of the living, strongly associated with notions of self-governance, self-rule-making (Kauffman, 1993; Varela, 1979). Artificial Life models have played a very important role on the modelling and conceptualization of how patterns of life can emerge out of decentralized and self-organized processes (Bedau, 2003), that is, understanding how living patterns are democratic-like. It is time to seek out how Artificial Life can inform and improve Democracy.

Artificial constitution of democratic societies

Our societies are not anymore a collection of rational agents, individually or collectively acting according to specific cognitive strategies with a mere instrumental relationship to artifacts. The french sociologist Bruno Latour (2005) challenges us to conceive of societies as actor-networks where cars, mobiles or pencils operate, interconnected with humans, all creating the collective network of interactions we call “society”. The increasing mediation of digital devices in our social life makes this artificial constitution of society the more apparent and the design of social interfaces has become the primary mode of social production (Yaneva, 2009).

Decentralized Autonomous Organizations as defined by smart contracts running over blockchain technologies (like Ethereum) (DuPont, 2017) or new direct participatory democracy platforms like Decidim.org (Barandiaran et al., 2017, 2019) (currently in use in hundreds of cities) are but two examples of how new technologies are boosting democratic life. But they face multiple design challenges to deliver the experience of effective and complex democracy they aim to make possible.

Artificial Democratic Life: the very idea

Following Bedau (2003) the goals of Artificial Life “include modelling and even creating life and life-like systems, as well as developing practical applications using intuitions

and methods taken from living systems.” If we now consider democracy as a form of (political) life we can envision the very idea of *Artificial Democratic Life* as the design and deployment of artificial infrastructures aimed at enhancing or improving social democratic life understood as a decentralized, egalitarian, and participatory decision making and commitment generating system of interactions.

Some open challenges that Artificial Democratic Life could help with include:

- *Equilibrium between the satisfaction of multiple interests.* Designing mechanisms for decision making that maximize the satisfaction of a maximum number of citizens. For an example see the work of Fain et al. (2016)
- *Facilitation of complex growth patterns in communicative discourse.* Designing deliberative mechanism (e.g. nested comments in a forum) that improve deliberative depth, quality and effectiveness. For an example see the work of Aragon et al. (2017)
- *Effective distribution of relevant information for decision making.* Designing collaborative moderation systems to channel relevant information into front pages or personalized profile feeds. For an example see the work of Mills and Fish (2015)
- *Ecological efficiency and diversity preservation in communication networks.* Designing algorithms for suggesting connectivity patterns between social network participant in order to improve diversity, robustness and communication efficiency.

Life have solved variants of these problems at different scales. And Alife has played a central role understanding and modelling them. Discovering and designing democracy as a form of life involves moving beyond its understanding as a complicated optimization problem, a statistical mass effect, a mere complex system or a political representational function of social cognition. Artificial Life should help understand democracy as a form of social life, with its potential for self-organization, its capacity to evolve into increasingly higher complexity, to homeostatically adapt to new circumstances preserving equity and to embody intelligence in a distributed social body.

Acknowledgements

Xabier E. Barandiaran acknowledges funding from project *FFI2014-52173-P* by the Spanish Ministry of Economy and Competitiveness.

References

Aragon, P., Kaltenbrunner, A., Calleja-Lopez, A., Pereira, A., Monterde, A., Barandiaran, X. E., and Gmez, V. (2017). Deliberative Platform Design: The Case Study of the Online Discussions in Decidim Barcelona. In *Social Informatics*, Lecture Notes in Computer Science, pages 277–287. Springer, Cham.

Bar-Yam, Y., Harmon, D., and de Bivort, B. (2009). Attractors and Democratic Dynamics. *Science*, 323(5917):1016–1017.

Barandiaran, X., Calleja, A., Monterde, A., Aragn, P., Linares, J., Romero, C., and Pereira, A. (2017). Decidim: redes politicas y tecnopoliticas para la democracia en red. *Recerca. Revista de pensament i anlisi.*, (21):137–150.

Barandiaran, X. E., Calleja-Lopez, A., and Monterde, A. (2019). Decidim: political and technopolitical networks for participatory democracy. *White Paper*, <https://docs.decidim.org/whitepaper/en/doc-info/>.

Bedau, M. A. (2003). Artificial life: organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, 7(11):505–512.

DuPont, Q. (2017). Experiments in algorithmic governance. A history and ethnography of The DAO, a failed decentralized autonomous organization. In Campbell-Verduyn, M., editor, *Bitcoin and Beyond: Cryptocurrencies, Blockchains, and Global Governance*, pages 157–178. Routledge, 1 edition.

Fain, B., Goel, A., and Munagala, K. (2016). The Core of the Participatory Budgeting Problem. In Cai, Y. and Vetta, A., editors, *Web and Internet Economics*, Lecture Notes in Computer Science, pages 384–399. Springer Berlin Heidelberg.

Husser, M. (2001). Synaptic function: Dendritic democracy. *Current Biology*, 11(1):R10–R12.

Kauffman, S. A. (1993). *The origins of order*. Oxford University Press US.

Latour, B. (2005). *Reassembling the social: an introduction to actor-network-theory*. Clarendon lectures in management studies. Oxford University Press, Oxford ; New York. OCLC: ocm58054359.

Miller, M. B. and Bassler, B. L. (2001). Quorum Sensing in Bacteria. *Annual Review of Microbiology*, 55(1):165–199.

Mills, R. and Fish, A. (2015). A Computational Study of How and Why reddit.com was an Effective Platform in the Campaign Against SOPA. In Meiselwitz, G., editor, *Social Computing and Social Media*, Lecture Notes in Computer Science, pages 229–241. Springer International Publishing.

Oyama, S. (2000). *The ontogeny of information*. Duke University Press.

Seeley, T. D. (2010). *Honeybee Democracy*. Princeton University Press. Google-Books-ID: txMkdG9G5acC.

Sherrington, C. S. (1941). *Man on his Nature*. Macmillan, Oxford, England.

Varela, F. J. (1979). *Principles of biological autonomy*. English. North Holland, New York.

Varela, F. J. (1995). Resonant cell assemblies: a new approach to cognitive functions and neuronal synchrony. *Biological Research*, 28(1):81–95.

Yaneva, A. (2009). Making the Social Hold: Towards an Actor-Network Theory of Design. *Design and Culture*, 1(3):273–288.

Can Bio-Inspired Swarm Algorithms Scale to Modern Societal Problems?

Darren M. Chitty, Elizabeth Wanner, Rakhi Parmar and Peter R. Lewis

Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Aston Triangle, Birmingham, B4 7ET, UK
darrenchitty@googlemail.com

Abstract

Taking inspiration from nature for meta-heuristics has proven popular and relatively successful. Many are inspired by the collective intelligence exhibited by insects, fish and birds. However, there is a question over their scalability to the types of complex problems experienced in the modern world. Natural systems evolved to solve simpler problems effectively, replicating these processes for complex problems may suffer from inefficiencies. Several causal factors can impact scalability; computational complexity, memory requirements or pure problem intractability. Supporting evidence is provided using a case study in Ant Colony Optimisation (ACO) regards tackling increasingly complex real-world fleet optimisation problems. This paper hypothesizes that contrary to common intuition, bio-inspired collective intelligence techniques by their very nature exhibit poor scalability in cases of high dimensionality when large degrees of decision making are required. Facilitating scaling of bio-inspired algorithms necessitates reducing this decision making. To support this hypothesis, an enhanced Partial-ACO technique is presented which effectively reduces ant decision making. Reducing the decision making required by ants by up to 90% results in markedly improved effectiveness and reduced runtimes for increasingly complex fleet optimisation problems. Reductions in traversal timings of 40-50% are achieved for problems with up to 45 vehicles and 437 jobs.

Introduction

The natural world is filled with a wealth of differing animals and ecosystems. Many of these organisms display collective behaviours which they use to overcome problems within their ecosystem such as ants foraging for food or bees communicating locations of nectar. These organisms have inspired many computing algorithms to assist in solving difficult real-world problems. Much of this inspiration comes from the exhibition of collective behaviours whereby thousands of organisms work together for the benefit of a colony, flock or hive. Each organism is simplistic in nature and by itself cannot survive but as part of a collective, problems such as finding sources of food can be solved. Nature has been used as a source of inspiration for the direct design of meta-heuristic algorithms that are moderately successful in solving optimisation problems of human consideration such as routing problems, information management and logistics to

name a few. Examples of bio-inspired collective behaviour algorithms include Ant Colony Optimisation (ACO) (Dorigo and Gambardella, 1997) inspired by how ants forage for food; Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007) based upon the way bees communicate sources of nectar; and Particle Swarm Optimisation (PSO) (Eberhart and Kennedy, 1995) which models the complex interactions between swarms of insects. These algorithms can be grouped under the term *swarm intelligence* through their use of hundreds or thousands of simulated digital organisms.

However, the types of problems that are tackled in nature by these organisms such as finding sources of food can be considered much more simplistic than the complex societal problems facing the human world. In an increasingly digital world whereby the available data is growing considerably alongside inter-connectivity and joined up thinking, the size and complexity of the problems that require solving are increasing rapidly such as with *smart city* planning (Batty, 2013; Murgante and Borruso, 2015). Moreover, unlike the natural world, restrictions exist on modern computers in terms of compute capability and available memory to be able to simulate many thousands of collective organisms.

In regards to the literature of swarm algorithms most implementations of collective behaviour algorithms are applied to relatively small problem sizes. However, there have been some works in the field addressing scalability. For instance, Piccand et al. (2008) found applying PSO to problems of greater than 300 dimensions resulted in failing to find the optimal solution more than 50% of the time. Cheng and Jin (2015b) noted that PSO fails to scale well to problems of a high dimensionality potentially as a result of problem structure. However, the authors employ a social learning implementation whereby many particles act as demonstrators and present promising results on problems of sizes up to 1,000 dimensions. Cheng and Jin (2015a) later propose a modification to PSO whereby instead of using local and global best solutions to update particle positions a pairwise competition is performed with the loser learning from the winner to update their position. The technique demonstrated improved results over PSO on benchmark problems of up to 5,000

dimensions although it was noted this was very computationally expensive. Cai et al. (2015) applies greedy discrete PSO to social network clustering problems with as many as 11,000 variables. For further reading Yan and Lu (2018) provide a review of the challenges of large-scale PSO.

Regarding ACO, Li et al. (2011) noted the scaling issues of the approach proposing a DBSCAN clustering approach to decompose large Travelling Salesman Problems (TSPs) of up to 1,400 cities into smaller sub-TSPs and solve these. Ismkhan (2017) also noted the computational cost and memory requirements and considered the use of additional heuristics or strategies to facilitate the scaling of the technique to larger problems. Improvements such as considering the pheromone matrix as a sparse matrix and using pheromone in a local search operator enabled ACO to be applied effectively to TSPs of over 18,000 cities. Chitty (2017) also noted computational issues with ACO and mitigated them with a non-pheromone matrix ACO approach which only made partial changes to good solutions applying the technique to TSP instances of up to 200,000 cities.

Therefore, it can be ascertained both ACO and PSO have issues in terms of scaling to high dimensional problems, the curse of dimensionality. Consequently, the question explored in this paper is can nature inspired, collective intelligence techniques scale up to the size and complexity of problems that the modern world desires solving? If not, what are the potential limiting causal factors for this and what mitigating steps could be taken? These questions will be investigated using a case study based on ACO to provide an illustration of the problems faced in scaling up a collective behaviour meta-heuristic and the hypothesized causal limitations by applying to a real-world fleet optimisation problem with steadily increasing complexity. The second aspect of this paper will attempt to mitigate ACO for these scalability issues using the novel Partial-ACO approach and enhance the approach further to assist scalability.

Ant Colony Optimisation: An Exemplar Case

A popular swarm based meta-heuristic is based upon the foraging behaviours of ants and known as Ant Colony Optimisation (ACO) (Dorigo and Gambardella, 1997). Essentially, the algorithm involves simulated ants moving through a graph G probabilistically visiting vertices and depositing pheromone as they move. The pheromone an ant deposits on the edges E of graph G is defined by the quality of the solution the given ant has generated. Ants probabilistically decide which vertex to visit next using this pheromone level deposited on the edges of graph G plus potential local heuristic information regarding the edges such as the distance to travel for routing problems. An *evaporation* effect is used to prevent pheromone levels building up too much and reaching a state of local optima. Therefore, ACO consists of two stages, the first *solution construction*, simulating ants, the second stage *pheromone update*. The solution construc-

tion stage involves m ants constructing complete solutions to problems. Ants start from a random vertex and iteratively make probabilistic choices using the *random proportional rule* as to which vertex to visit next. The probability of ant k at point i visiting point $j \in N^k$ is defined as:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (1)$$

where $[\tau_{il}]$ is the pheromone level deposited on the edge leading from location i to location l ; $[\eta_{il}]$ is the heuristic information from location i to location l ; α and β are tuning parameters controlling the relative influence of the pheromone deposit $[\tau_{il}]$ and the heuristic information $[\eta_{il}]$.

Once all ants have completed the solution construction stage, pheromone levels on the edges E of graph G are updated. First, evaporation of pheromone levels upon every edge of graph G occurs whereby the level is reduced by a value ρ relative to the pheromone upon that edge:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad (2)$$

where ρ is the *evaporation rate* typically set between 0 and 1. Once this evaporation is completed each ant k will then deposit pheromone on the edges it has traversed based on the quality of the solution found:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

where the pheromone ant k deposits, $\Delta\tau_{ij}^k$ is defined by:

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & \text{if edge } (i, j) \text{ belongs to } T^k \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $1/C^k$ is the quality of ant k 's solution T^k . This ensures that better quality solutions found by an ant result in greater levels of pheromone being deposited on those edges.

Consideration of the Scalability of ACO

From a computational point of view, implementing an ant inspired algorithm on computational hardware to solve large-scale problems suffers from three potential limitations regarding overall performance. The degree of memory required, the computational costs of simulating thousands of ants and the sheer intractability of the problem itself.

Memory Requirements A key aspect of ACO is the pheromone matrix used to store pheromone levels on all the edges in the graph G . This can require significant amounts of computing memory. For instance, a fully connected 100,000 city Travelling Salesman Problem (TSP) will have ten billion edges in graph G . Using a float data type requiring four bytes of memory will need approximately 37GB of memory to store the pheromone levels, considerably greater than available in standard computing platforms. In the natural world storing pheromone levels is not an issue with

an infinite landscape to store them. A secondary memory requirement arises from ants only updating the pheromone matrix once all ants have constructed their solutions necessitating storing these in memory too. For a 100,000 city TSP a single ant will require 0.38MB of memory using a four byte integer data type. If the number of ants equals the number of vertices an additional 37GB of memory would be required.

An ant inspired algorithm that addresses this memory overhead is Population-based ACO (P-ACO) (Guntsch and Middendorf, 2002) whereby the pheromone matrix is removed with only a population of ant solutions maintained. From this population, pheromone levels are reconstructed for the available edges by finding the edges taken within the population from the current vertex and assigning pheromone to edges based on the solution quality.

Computational Costs A second aspect to consider with ACO is the time it will take to simulate ants through the graph G . At each vertex an ant needs to decide which vertex to next visit. This is performed probabilistically by looking at the pheromone levels, and possibly heuristic information, on all available edges. This requires computing probabilities for all these edges. As an example, take a 100,000 city TSP, at the first vertex an ant will have 99,999 possible edges to take all of which require obtaining probabilities from. Once an ant has made its choice it moves to the chosen vertex and once again analyses all available edges, now 99,998. Thus, for the 100,000 city TSP an ant will need to perform five billion edge comparisons. If a processor is capable of 100 GFLOPS (billion floating point operations per second) and assuming an edge comparison takes one floating point operation it will require at least 0.05 seconds to simulate an ant through graph G . If using a population of ants equivalent to the number of vertices in graph G then to complete one iteration of solution construction would require nearly 90 minutes of computational time. For ants in nature, compute time is not an issue since each ant can act independently although, the actual time it would take real ants to move through a network of this size would still be problematic.

The simulation of ants is inherently parallel in nature and therefore can easily take advantage of parallel computing resources to alleviate the computational costs. In recent years, speeding up ACO has focused on utilising Graphical Processor Units (GPUs) consisting of thousands of SIMD processors. Delévacq et al. (2013) provide a comparison of differing parallelisation strategies for *MAX-MIN* ACO on GPUs. Cecilia et al. (2013) reduced the decision making process of ants using a GPU with an *Independent Roulette* approach that exploits data parallelism and Dawson and Stewart (2013) went a step further introducing a *double spin* ant decision methodology when using GPUs. These works have provided speedups ranging from 40-80x over a sequential implementation, a considerable improvement. Peake et al. (2018) used the Intel Xeon Phi and a vectorized candidate list methodology to achieve a 100 fold speedup. Can-

didate lists are an alternative efficiency method of reducing the computational complexity of ACO whereby ants are restricted to selecting a subset of the available vertices within its current neighbourhood. If none of these vertices are available then the full set are considered as normal. Gambardella and Dorigo (1996) used this approach to solve TSP instances whereby speedups were observed but also a reduction in accuracy due to sub-optimal edges being taken.

Problem Intractability A final scalability issue with ACO involves the amenability of the problem under consideration to be tackled by ACO. The key issue is the probabilistic methodology ACO employs to decide which edge to take next by utilising the pheromone levels on the available edges to influence the probabilities. Computationally, an ant will take the pheromone level on each edge, and if available multiply by the heuristic information, and multiply this by a random value between zero and one. The edge with the largest product is selected as the next to be traversed.

As an example consider a simple decision point whereby an ant has two choices available, one being the correct, optimal selection, the other suboptimal. If the pheromone levels on each edge are equal then there is a 0.5 probability the ant will take the optimal edge. However, consider ten independent decision points each with two possible choices akin to a binary optimisation problem such as clustering a set of items into two groups. Probabilistically this is equivalent to ten coin flips. With equal pheromone on all edges, there is only a 0.5^{10} probability of an ant making the optimal choices, approximately one in a thousand. Conversely, an ant has a 0.999 probability of generating a sub-optimal solution so 1,000 ants would need to traverse graph G to obtain an optimal solution. For a much larger problem of 100,000 decision points this would be $0.5^{100,000}$ requiring $10^{30,102}$ ants to find the optimal solution.

Consequently, pheromone levels are there to help guide the ants to taking the optimal edge. Consider the previous 100,000 decision point example again but with high levels of pheromone on the edge to the optimal choice, say 0.99 vs. 0.01 on the suboptimal edge, then the probability of obtaining the optimal solution will be $0.99^{100,000}$ or approximately 3^{437} ants required, still a significant number. In fact, to get to a manageable number of ant simulations the pheromone on the optimal edges would need to be of the order 0.9999 vs. 0.0001 on the suboptimal edge when only approximately 20,000 ants would need to traverse the network before an ant probabilistically takes the correct edges at each decision point. However, this means the pheromone level would need to be 10,000 times greater on the optimal edge than the sub-optimal edge. Moreover, the pheromone levels would need to build up over time before reaching these levels.

Hence, it can be observed that applying ACO to ever larger problems results in increasingly reduced probabilities of optimal solutions being found unless the pheromone levels become increasingly stronger on the important edges.

Candidate lists, as covered in the previous section, can reduce the number of decisions that ants need to make but with a potential error reducing accuracy and can only use if heuristic information is available to define the neighbourhood. Of course ants in the natural world do not have problems of this magnitude to solve and have the numbers if necessary without any undue computational cost to consider.

An Illustration of the Scalability Issues With ACO

To highlight the potential drawbacks of ACO it will be tested against a difficult set of real-world fleet optimisation problems of steadily increasing complexity. These problems have been supplied by a Birmingham based maintenance company which operates a fleet of vehicles performing services at customer properties within the city. Each vehicle starts from a depot and must return when it has finished servicing customers. Each customer is defined by a location and a job duration predicting the length of time the job will take and in some cases, a time window for when the jobs must be completed. The speed of travel of a vehicle between maintenance jobs is defined at an average 13kph to account for city traffic. There is also a hard start time and end time to a given working day defined as 08:00 and 19:00 hours. Fleet optimisation is essentially the classic Multiple Depot Vehicle Routing Problem (MDVRP) (Dantzig and Ramser, 1959) with Time Windows (MDVRPTW).

The MDVRP can be formally defined as a complete graph $G = (V, E)$, whereby V is the vertex set and E is the set of all edges between vertices in V . The vertex set V is further partitioned into two sets, $V_c = V_1, \dots, V_n$ representing customers and $V_d = V_{n+1}, \dots, V_{n+p}$ representing depots whereby n is the number of customers and p is the number of depots. Furthermore, each customer $v_i \in V_c$ has a service time associated with it and each vehicle $v_i \in V_d$ has a fixed capacity associated with it defining the ability to fulfill customer service. Each edge in the set E has an associated cost of traversing it represented by the matrix c_{ij} . The problem is essentially to find the set of vehicle routes such that each customer is serviced once only, each vehicle starts and finishes from the same depot, each vehicle does not exceed its capacity to service customers and the overall cost of the combined routes is minimised.

The worksheet data supplied by the company has been divided into a series of problems of increasing complexity and size which are described in Table 1. The manner in which the company assigns customer jobs to vehicles is known *a priori* enabling a *ground truth* for the optimisation process. Effectively, the company assigns geographically related jobs to vehicles based on postcode and then orders them such that the vehicle performs the job furthest from its depot first and then works its way back, time windows allowing.

To highlight the drawbacks of ACO in terms of scalability, the *MAX-MIN* Ant System (*MMAS*) (Stützle and Hoos, 2000) will be tested upon these fleet optimisa-

Table 1: Real-world problem scenarios supplied by a Birmingham maintenance company described in terms of the vehicles available, customers to service, the total predicted service time required and the total travel time using the company’s current scheduling.

Problem	Vehicles	Jobs	Total Job Servicing (hh:mm)	Total Fleet Traversal Time (hh:mm)
Week_1	8	77	47:09	31:12
Week_2	8	79	48:24	22:49
Week_3	8	81	48:33	19:54
Fortnight_1	16	156	95:33	54:01
Fortnight_2	16	138	102:01	57:07
Fortnight_3	16	160	96:57	42:43
ThreeWeek_1	24	237	144:06	73:55
ThreeWeek_2	24	217	150:25	79:56
ThreeWeek_3	24	219	150:34	77:01
Month_1	32	298	198:58	99:50
Month_2	32	313	190:26	96:28
SixWeek_1	45	437	267:47	142:46

tion problems. *MMAS* simulates ants through the graph G but, in contrast to standard ACO, only the best found solution provides pheromone updates. Additionally, minimum and maximum levels of pheromone on edges are defined. To solve the fleet optimisation problem the fully connected graph G has vertices relating to the number of vehicles and customer jobs. Ants start from a random vehicle vertex then visit every other vertex once only resulting in a sequence of vehicles beginning from their specified depots followed by the customer jobs they will service before returning to their depot. This representation is shown in Figure 1 whereby V relates to a vehicle and J relates to a job. The first vehicle will undertake jobs 6, 5 and 9, the second jobs 3, 7, and 2 and so forth.

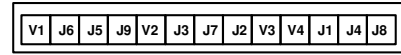


Figure 1: Example solution representation.

Once a new solution has been generated its quality needs to be assessed. This is measured using two objectives, the first of which is to maximise the number of jobs correctly performed within their given time window. The second objective is the minimisation of the total traversal time of the fleet of vehicles. Reducing the number of missed jobs is the primary objective. Hence, comparing two solutions, if the first services more customer jobs than the second then the first solution is considered the better. If though they have equal customer job time serviced then the solution with the lower fleet traversal time is considered the better.

The pheromone to deposit is calculated using these objectives to be optimised. A penalty based function will be utilised for the first objective whereby any customers that have not been serviced due to capacity limitations or missing the time window will be penalised by the predicted job time. The secondary objective is to minimise the time the fleet of vehicles spend traversing the road network between jobs. Solution quality can then be described as:

$$C^k = (S - s^k + 1) * L^k \quad (5)$$

where S is the total amount of time of jobs to be serviced, s^k is the amount of job service time achieved by ant k 's solution and L^k is the total traversal time of the fleet of vehicles of ant k 's solution. Clearly, if ant k has achieved the primary objective of fulfilling all customer demand S then C^k becomes merely the total traversal time of the fleet.

Table 2: Parameters used with the ACO $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ algorithm

Number of Ants	192
Max Iterations	1,000,000
α	1.0
β	1.0
ρ	0.02

A parallel implementation of $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ is tested against the exemplar problems from Table 1 with experiments conducted using an AMD Ryzen 2700 processor using 16 parallel threads of execution. The algorithms were compiled using Microsoft C++. Experiments are averaged over 25 individual execution runs for each problem with a differing random seed used in each instance. The parameters used with $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ are described in Table 2.

The results from these experiments are shown in Table 3 whereby the issue of scalability is abundantly clear. As the size and complexity of the fleet optimisation problems increases, the ability for $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ to find a solution which satisfies all the customer demand reduces. Similarly, $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ cannot obtain solutions with a lower fleet traversal time than the company's own scheduling when the problem size increases. Therefore, it can be considered that these results support the hypothesis that a nature inspired swarm algorithm such as ACO suffers from scalability issues.

Table 3: The $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ results for fleet optimisation in terms of customers serviced, reductions in fleet traversal time over the original scheduling and the execution time.

Problem	Job Time Serviced (%)	Traversal Reduction (%)	Execution Time (mins)
Week_1	100.00 ± 0.00	33.62 ± 3.39	2.23 ± 0.10
Week_2	100.00 ± 0.00	30.70 ± 4.85	2.33 ± 0.10
Week_3	100.00 ± 0.00	31.48 ± 4.68	2.49 ± 0.10
Fortnight_1	100.00 ± 0.00	23.84 ± 7.46	6.56 ± 0.13
Fortnight_2	100.00 ± 0.00	28.64 ± 4.99	6.84 ± 0.10
Fortnight_3	100.00 ± 0.00	25.02 ± 4.49	5.76 ± 0.11
ThreeWeek_1	99.81 ± 0.18	-11.43 ± 7.62	13.09 ± 0.11
ThreeWeek_2	99.95 ± 0.11	7.33 ± 6.56	11.57 ± 0.15
ThreeWeek_3	99.86 ± 0.18	-2.36 ± 5.92	12.09 ± 0.15
Month_1	99.76 ± 0.18	-17.85 ± 3.75	19.75 ± 0.13
Month_2	99.91 ± 0.13	6.24 ± 3.47	21.57 ± 0.15
SixWeek_1	98.46 ± 0.57	-26.97 ± 6.76	39.54 ± 0.26

Addressing the ACO Scalability Issues

Given that the evidence seems to support the hypothesis that ACO methods will struggle to scale to larger, increasingly complex problems the next step is to attempt to address the underlying reasons behind the poor performance. As has been previously discussed, a key problem is the degree of decision making required to form solutions vs. the probabilistic nature of ACO. Therefore, it can be theorized that if the degree of decision making is reduced, ACO may well scale better. A novel modification to the ACO algorithm

known as Partial-ACO (Chitty, 2017) provides a mechanism to achieve this. Essentially, this technique minimises the computational effort required and the probabilistic fallibility of ACO by ants only considering *partial* changes to their solutions rather than constructing completely new solutions. In contrast to standard ACO algorithms, Partial-ACO operates in a population based manner much the same as P-ACO. Essentially, a population of ants is maintained each of which represent a solution to the given problem. Pheromone levels are constructed from the edges taken within this population of solutions with their associated qualities which are relative to the best found solution. Partial-ACO also operates in a pure steady-state manner to preserve diversity. An ant only replaces its own *best* solution with a new solution if it is of better quality. Hence, each ant maintains a *local memory* of its best yet found solution.

This l_{best} memory enables an ant to consequently only partially change this solution to form a new solution. To partially modify its locally best found solution an ant simply picks a random point in the solution as a starting point and a random sub-length of the tour to preserve. The remaining aspect of the tour is rebuilt using standard ACO methodologies in a P-ACO manner. This process is illustrated in Figure 2. To highlight the computational advantage of this technique, consider retaining 50% of solutions for a 100,000 TSP problem. In this instance only 50,000 probabilistic decisions now need to be made and only 1.25 billion pheromone comparisons would be required, a reduction of 75%. An overview of the *Partial-ACO* technique is described in Algorithm 1.

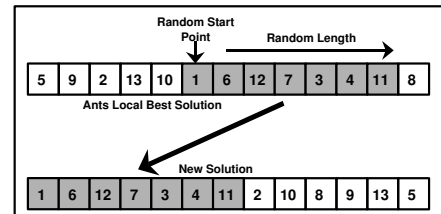


Figure 2: An illustration of the *Partial-ACO* methodology.

Algorithm 1 *Partial-ACO*

- 1: **for** each ant **do**
- 2: Generate an initial solution probabilistically
- 3: **end for**
- 4: **for** number of iterations **do**
- 5: **for** each ant k **do**
- 6: Pick uniform random start point from l_{best} solution
- 7: Select uniform random length of l_{best} to preserve
- 8: Copy l_{best} points from start for specified length
- 9: Complete remaining aspect probabilistically
- 10: If new solution better than l_{best} then update l_{best}
- 11: **end for**
- 12: **end for**
- 13: Output best l_{best} solution (the g_{best} solution)

Evaluating the Partial-ACO Approach

To test the hypothesis that reducing the degree of decision making that ants need to perform will enable them to scale to larger problems, Partial-ACO will be tested upon the same problems as previously. The parameters used for the implementation of Partial-ACO are described in Table 4. Note the lower number of ants in contrast to $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$. The original Partial-ACO work found a low number of ants was highly effective. To ensure the same number of solutions are evaluated, Partial-ACO will use six times more iterations.

The results for the MDVRP fleet optimisation problem are shown in Table 5 whereby it can be observed that now in all problem instances, reductions in the fleet traversal times are achieved by Partial-ACO over the commercial company's methodology. In fact, in many cases the improvement in the reduction in fleet traversal time is significantly better than that from $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$, especially regarding the larger, more complex, problems. Disappointingly though, the Partial-ACO technique was also unable to service all the customer jobs for the larger problems. In terms of execution timings, Partial-ACO is slightly slower than $\mathcal{M}\mathcal{M}\mathcal{A}\mathcal{S}$ when evaluating the same number of solutions. This is caused by the requirement to construct the edge pheromone levels at each point as an ant moves through the graph G .

Table 4: Parameters used with the Partial-ACO algorithm

Number of Ants	32
Max Iterations	6,000,000
α	3.0
β	1.0

Table 5: The Partial-ACO results for fleet optimisation in terms of customers serviced, reductions in fleet traversal time over the original scheduling and the execution time.

Problem	Job Time Serviced (%)	Traversal Reduction (%)	Execution Time (mins)
Week_1	100.00 \pm 0.00	32.29 \pm 3.77	4.69 \pm 0.49
Week_2	100.00 \pm 0.00	22.39 \pm 7.84	4.76 \pm 0.45
Week_3	100.00 \pm 0.00	28.75 \pm 5.55	4.90 \pm 0.50
Fortnight_1	99.98 \pm 0.10	23.12 \pm 6.30	9.87 \pm 0.43
Fortnight_2	100.00 \pm 0.00	27.27 \pm 5.49	10.18 \pm 0.38
Fortnight_3	100.00 \pm 0.00	29.70 \pm 6.64	9.09 \pm 0.48
ThreeWeek_1	100.00 \pm 0.00	17.13 \pm 4.47	16.42 \pm 0.71
ThreeWeek_2	99.91 \pm 0.21	20.64 \pm 2.36	14.54 \pm 0.78
ThreeWeek_3	99.84 \pm 0.27	18.00 \pm 7.84	14.95 \pm 0.49
Month_1	99.82 \pm 0.18	19.60 \pm 4.09	23.59 \pm 0.83
Month_2	99.81 \pm 0.22	20.25 \pm 3.60	24.73 \pm 1.08
SixWeek_1	97.48 \pm 0.66	11.50 \pm 5.78	41.71 \pm 0.47

Enhancing Partial-ACO

Although the results of the Partial-ACO approach seemed promising they did not significantly enforce the premise that ants are less effective with higher degrees of decision making. Analysing the Partial-ACO methodology, it could be postulated that modifying a continuous subsection of an ant's locally best found tour could present problems in that individual points within the solution cannot be displaced a great distance. They are confined to a local neighbourhood as to how they could be reorganised.

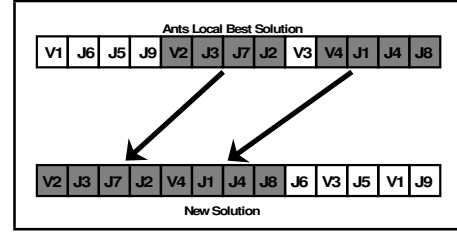


Figure 3: An illustration of the *Enhanced* Partial-ACO methodology whereby two vehicle schedules are preserved.

An enhancement to Partial-ACO is proposed which will facilitate the movement of points in a given ant's locally best solution. To achieve this, it is proposed that instead of one continuous segment of an ant's solution being preserved and the remaining part probabilistically regenerated as is the norm, a number of separate blocks throughout the solution are preserved instead. In this way a point at one end of a given solution could be moved to points throughout the solution. This should help prevent the ants becoming trapped in local optima. This methodology is actually well suited to the fleet optimisation problem since each vehicle can be considered as a stand-alone aspect of the solution. Each preserved block could in fact be a vehicle's complete job schedule. Before attempting to construct a new solution an ant can simply decide randomly which vehicle schedules to preserve and then use the probabilistic behaviour of moving through the graph G to assign the remaining customer jobs to the remaining vehicles as normally. Figure 3 demonstrates the principle whereby it can be observed that two sections representing vehicle schedules are preserved by an ant from its l_{best} solution with the rest built up probabilistically.

Table 6: The enhanced Partial-ACO results for fleet optimisation regards customers serviced, reductions in fleet traversal time over the original scheduling and the execution time.

Problem	Job Time Serviced (%)	Traversal Reduction (%)	Execution Time (mins)
Week_1	100.00 \pm 0.00	34.75 \pm 5.92	5.08 \pm 0.86
Week_2	100.00 \pm 0.00	38.60 \pm 4.14	5.15 \pm 0.81
Week_3	100.00 \pm 0.00	36.00 \pm 5.05	5.76 \pm 1.00
Fortnight_1	100.00 \pm 0.00	49.19 \pm 0.57	10.75 \pm 0.34
Fortnight_2	100.00 \pm 0.00	50.18 \pm 0.49	11.25 \pm 0.22
Fortnight_3	100.00 \pm 0.00	47.24 \pm 0.80	10.21 \pm 0.15
ThreeWeek_1	100.00 \pm 0.00	46.55 \pm 1.40	18.78 \pm 0.14
ThreeWeek_2	100.00 \pm 0.00	42.61 \pm 0.92	17.48 \pm 0.14
ThreeWeek_3	100.00 \pm 0.00	44.21 \pm 1.20	17.57 \pm 0.16
Month_1	100.00 \pm 0.00	34.80 \pm 1.94	26.22 \pm 0.35
Month_2	100.00 \pm 0.00	36.05 \pm 0.92	26.96 \pm 0.20
SixWeek_1	100.00 \pm 0.00	10.09 \pm 4.16	42.36 \pm 0.17

To evaluate the enhancement to Partial-ACO, it will be tested against the same problems as previously using the same parameters as described in Table 4. The results are shown in Table 6 and when contrasted to those in Table 5 it can be seen that significant improvements have been made over the standard Partial-ACO approach. Now, for all problem instances including the most complex, all the customer jobs have all been serviced. Furthermore, significantly improved reductions in the fleet traversal times have

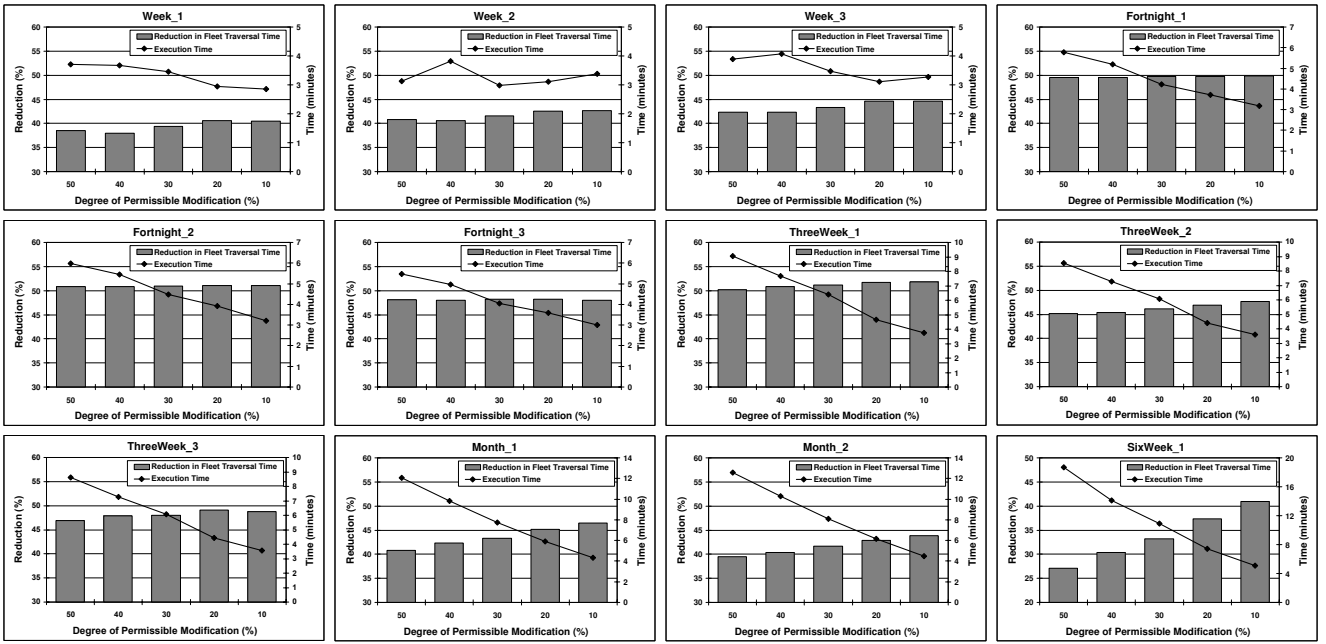


Figure 4: The average reductions in fleet traversal timings over the commercial company’s methodology as the degree of permissible ant modification is reduced for each problem instance. Execution timings are also displayed on the same graph.

been achieved. In fact, as much as an additional 29% reduction in fleet traversal time for the *ThreeWeek_1* problem instance. With regards execution timings, block based Partial-ACO is slightly slower which is caused by the overhead of assembling blocks of retained solution rather than one continuous section. Consequently, from these results it can be inferred that when using solution preservation with Partial-ACO, smaller random blocks should be preserved rather than a continuous section to obtain improved results.

Reducing the Degree of Modification

The enhanced Partial-ACO approach has provided a significant improvement over standard ACO techniques such as *MMAS*. However, recall that the original hypothesis supporting the development of Partial-ACO was that potentially, collectively intelligent meta-heuristics could fail to scale well to larger problems because of the degree of decision making that is necessitated. This hypothesis seems to be borne out by the results achieved by Partial-ACO to some degree. However, it is possible to test this hypothesis to a greater extent by reducing the degree of permissible modification an ant can make. Currently, an ant will randomly preserve any amount of its locally best solution and will modify the rest using the ACO probabilistic rules, approximately 50% of the solution on average. To avoid a large aspect of redesign, a maximum degree of modification could be imposed on an ant changing its locally best solution. This will firstly have the benefit of increasing the speed of Partial-ACO but also, if the hypothesis is correct, lead to improved optimisation. As such, the previous experiments will be re-run using a maximum modification limit ranging from 50%

of the solution down to 10% in increments of 10%. The improved block preserving version of Partial-ACO will be used and additionally, to prevent ants becoming trapped in local optima with a small random probability (0.001) an ant can modify its locally best found solution to any degree.

The results from reducing the degree of permissible modification of ants locally best solutions are shown in Figure 4. These describe the reductions in fleet traversal times over the commercial company’s own scheduling and execution timings. The percentage of customer jobs serviced is not shown as in all cases 100% of jobs were serviced. A clear trend can be observed for considerably improved reductions in fleet traversal times whilst reducing the degree of permissible modification. This further reinforces the hypothesis that due to the probabilistic nature of ants, the degree of decision they are exposed to must be reduced in order for the technique to scale. Moreover, the larger the problem, the more pronounced the effect as evidenced by the month and six week long problem instances. Remarkably, for the largest problem, reducing ants decision making by 90% yields the best results with a four fold improvement in solution quality fully enforcing the hypothesis that ants significantly benefit from reduced decision making. A further added benefit from reduced decision making of ants is faster execution times. Not only does the Partial-ACO approach provide improved reductions in fleet traversal times but can also achieve these reductions much faster by reducing the probabilistic decisions that ants need to make. In fact, from these results, it can be stated that Partial-ACO is more accurate, much faster and more scalable than standard ACO as a consequence of the reduced decision making of ants within the algorithm.

Conclusions

This paper has posed the hypothesis that although algorithms inspired by the collective behaviours exhibited by natural systems have been effective for simplistic human level problems, they may fail to problems of much greater complexity. Evidence supporting this hypothesis is provided by applying Ant Colony Optimisation (ACO) to a range of increasingly complex fleet optimisation problems whereby degrading results are observed as complexity rises. A theory postulated is that the degree of decision making required by ants to construct solutions becomes too great. Given a small probability of an ant choosing poorly at each decision point, the greater decisions required to construct a solution and available choices, the greater probability of reduced solution qualities. Consequently, this paper applies the Partial-ACO approach to reduce the decision making of ants. Indeed, the Partial-ACO approach provided much improved results for a complex fleet optimisation problem enabling ACO to scale to much larger problems with reductions of over 50% in traversal times achieved with the subsequent savings in fuel costs for the given company and similarly significant reductions in city traffic and hence vehicular emissions.

In fact remarkably, for the larger problems, reducing ants decision making by up to 90% yielded the best results. Consequently, this reinforces the posed hypothesis that for collective behaviour algorithms to scale effectively, the degree of decision making should be minimised as much as possible. However, further studies need to be performed with bio-inspired algorithms besides ACO such as Particle Swarm Optimisation (PSO) and Artificial Bee Colony (ABC) and to consider problem areas other than fleet optimisation to provide better supporting evidence to the hypothesis posed by this paper.

Acknowledgement

Carried out for the System Analytics for Innovation project, part-funded by the European Regional Development Fund.

References

- Batty, M. (2013). Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279.
- Cai, Q., Gong, M., Ma, L., Ruan, S., Yuan, F., and Jiao, L. (2015). Greedy discrete particle swarm optimization for large-scale social network clustering. *Information Sciences*, 316:503–516.
- Cecilia, J. M., García, J. M., Nisbet, A., Amos, M., and Ujaldón, M. (2013). Enhancing data parallelism for ant colony optimization on GPUs. *Journal of Parallel and Distributed Computing*, 73(1):42–51.
- Cheng, R. and Jin, Y. (2015a). A competitive swarm optimizer for large scale optimization. *IEEE transactions on cybernetics*, 45(2):191–204.
- Cheng, R. and Jin, Y. (2015b). A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291:43–60.
- Chitty, D. M. (2017). Applying ACO to large scale TSP instances. In *UK Workshop on Computational Intelligence*, pages 104–118. Springer.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.
- Dawson, L. and Stewart, I. (2013). Improving ant colony optimization performance on the GPU using CUDA. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1901–1908. IEEE.
- Delévacq, A., Delisle, P., Gravel, M., and Krajecki, M. (2013). Parallel ant colony optimization on graphics processing units. *Journal of Parallel and Distributed Computing*, 73(1):52–61.
- Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66.
- Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE.
- Gambardella, L. M. and Dorigo, M. (1996). Solving symmetric and asymmetric tsps by ant colonies. In *Proceedings of IEEE international conference on evolutionary computation*, pages 622–627. IEEE.
- Guntsch, M. and Middendorf, M. (2002). A population based approach for ACO. In *Workshops on Applications of Evolutionary Computation*, pages 72–81. Springer.
- Ismkhan, H. (2017). Effective heuristics for ant colony optimization to handle large-scale problems. *Swarm and Evolutionary Computation*, 32:140–149.
- Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471.
- Li, X., Liao, J., and Cai, M. (2011). Ant colony algorithm for large scale tsp. In *2011 International Conference on Electrical and Control Engineering*, pages 573–576. IEEE.
- Murgante, B. and Borruso, G. (2015). Smart cities in a smart world. In *Future City Architecture for Optimal Living*, pages 13–35. Springer.
- Peake, J., Amos, M., Yiapanis, P., and Lloyd, H. (2018). Vectorized candidate set selection for parallel ant colony optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1300–1306. ACM.
- Piccard, S., O'Neill, M., and Walker, J. (2008). On the scalability of particle swarm optimisation. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2505–2512. IEEE.
- Stützle, T. and Hoos, H. H. (2000). Max–min ant system. *Future generation computer systems*, 16(8):889–914.
- Yan, D. and Lu, Y. (2018). Recent advances in particle swarm optimization for large scale problems. *Journal of Autonomous Intelligence*, 1(1):22–35.

Towards Low-Carbon Conferencing: Acceptance of Virtual Conferencing Solutions and Other Sustainability Measures in the ALIFE Community

Harold Fellermann¹, Alexandra S. Penn², Rudolf M. Fuchsli³, Jaume Bacardit¹ and Angel Goñi-Moreno¹

¹School of Computing, Newcastle University, Newcastle Upon Tyne, UK

²Centre for Environmental Strategy, University of Surrey, Guildford, UK

³School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland
harold.fellermann@newcastle.ac.uk, angel.goni-moreno@newcastle.ac.uk

Abstract

The latest report from the Intergovernmental Panel on Climate Change (IPCC) estimated that humanity has a time window of about 12 years in order to prevent anthropogenic climate change of catastrophic magnitude. Green house gas emission from air travel, which is currently rising, is possibly one of the factors that can be most readily reduced. Within this context, we advocate for the re-design of academic conferences in order to decrease their environmental footprint. Today, virtual technologies hold the promise to substitute many forms of physical interactions and increasingly make their way into conferences to reduce the number of travelling delegates. Here, we present the results of a survey in which we gathered the opinion on this topic of academics worldwide. Results suggest there is ample room for challenging the (dangerous) business-as-usual inertia of scientific lifestyle.

Background

The latest report from the Intergovernmental Panel on Climate Change (IPCC) estimated that humanity has a time window of about 12 years in order to prevent anthropogenic climate change of catastrophic magnitude (IPCC, 2018) and calls for drastic actions in order to halve CO₂ emissions by no later than 2030. In light of these reports, a growing number of scientists rethinks their way of working and attempts to implement a low-carbon agenda (Nevins, 2014; Nathans and Sterling, 2016; Cobb et al., 2018).

As analyzed by Achten et al. (2013), professional mobility constitutes about 75% of the carbon emissions of a representative PhD project, with 35% of total emissions being associated with conference attendance alone.

Global impact of flying is immense—and it is rapidly increasing. Recent reports establish aviation impact as high as 2.1% of global CO₂ emissions (Girling et al., 2018)—roughly equivalent to Germany’s total emissions. Moreover, this figure does not consider the enhanced impact of greenhouse gas emissions at high altitude (Lee et al., 2009).

Universities are also dealing with the problem of aviation (Nature editorial, 2015). For example, the University of British Columbia has recently published a comprehensive analysis of emissions due to flying—where conferences

are highlighted as the biggest contributor to the academic environmental footprint (Wynes and Donner, 2018). University of California Los Angeles applies a tax to academic flights—although this measure “doesn’t go too far” (Hasan, 2018). In addition to this, the University of Basel is considering mandatory train travel within some radius (Leybold-Johnson, 2019). However, university-sustainability policies do not necessarily tackle this issue and may “unintentionally encourage academic staff to fly more rather than less” (Glover et al., 2017).

Already almost a decade ago, a multi-hub conference (*i.e.*, talks/audience distributed among different locations) paved the way for low-carbon conferencing strategies (Krumdieck and Orchard, 2011).

Here, we present the result of a community survey on the acceptance of different measures for implementing low-carbon conferencing with particular emphasis on virtual conferencing solutions. The survey has been performed for the Artificial Life (ALIFE) community. Results may be considered in the future to adapt the ALIFE Conference to a low-carbon path.

Results

Our online survey consisted of 28 questions that assessed respondents general conferencing habits, their experience with and evaluation of virtual technologies to substitute for presentational conferencing, as well as acceptance of potential measures that conference organizers might put in place to reduce the ecological impact of their event. The survey was open to unrestricted anonymous participation over a period of four weeks and had been announced through the mailinglist and Twitter feeds of the International Society of Artificial Life (announce@isal.groups.io), as well as the personal professional networks (email and Twitter) of the ALIFE 2019 organizers.

Survey participants

At the time of writing, the survey was completed by 170 respondents. Figure 1 shows the profile of respondents. Most of these work in academia (academic or student), 67% are

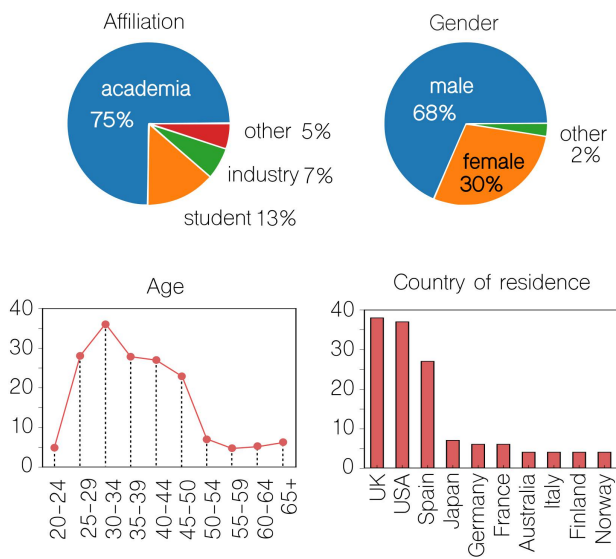


Figure 1: Demographics of survey respondents: professional affiliation, gender, age, and top-most countries of residence (80% of total).

male and the majority is within 30 and 50 years of age.

Since the survey was openly accessible, we devised several questions in order to determine respondents' affiliation with the Artificial Life community. Participants were asked whether they are interested in Artificial Life (very much: 68, somewhat: 48, not at all: 49), are working in Artificial Life (fully: 26, partly: 63, not at all: 77), have attended ALIFE or ECAL conferences (once: 10, 2-5 times: 32, 6-9 times: 14, more than 10 times: 5, never: 105), have published in the Artificial Life journal (once: 12, 2-5 times: 32, 6-9 times: 10, more than 10 times: 6, never: 107), or whether they are members of the International Society for Artificial Life (yes: 27, no: 130, do not know: 11). Since all of these are exceedingly correlated, we simply chose the answer to the question "Do you consider yourself as working in the field of Artificial Life" as an indicator of the community affiliation. With this, our survey includes 89 respondents (52%) from the ALIFE community and 81 respondents (45%) from other research areas, which includes 5 respondents who did not answer the respective question.

Conference attendance

Participants reported to attend an average of three to four conferences per year, one or two of which typically being intercontinental. All in all, conference attendance appears to follow a Poisson distribution, *i.e.* we do not observe a heavy tail distribution in our data sample (Figure 2).

Participants were asked to weigh their reasons for attending conferences (Figure 3). This establishes the priorities of the respondents, which eventually need to be considered

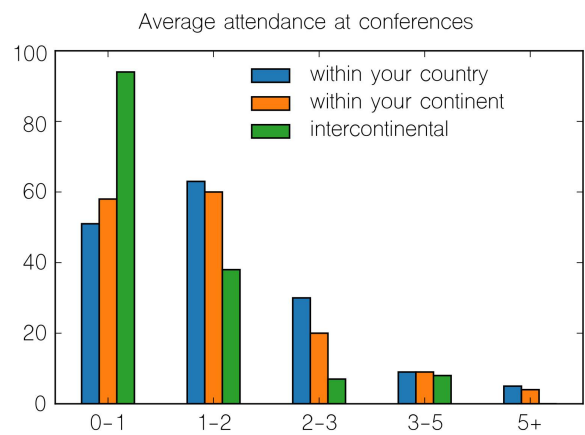


Figure 2: Average number of conferences attended per year.

when devising sustainability measures, particularly regarding inclusion of virtual remote participation. Presenting own work, discovering new work, and catching up with work done by colleagues were named as the three top priorities. Employment considerations such as finding potential future employers or employees were regarded less relevant. Additional comments allowed respondents to name further reasons, of which the following received the most mentions: Keeping in touch with friends and colleagues; boosting academic reputation, *e.g.* through invited talks; being provided with an environment that allows to completely focus on science and develop new ideas; travelling to new places, especially when paired with vacations.

Participants were then asked to think about the last conference they intended to attend, but eventually decided against. The main reasons for not attending were, in decreasing order: too much travel involved (85 mentions), travel budget exceeded (85 mentions), nothing to present (55), and competing professional (53 mentions) or personal commitments (42 mentions). Noteworthy, additional comments completed this list with five mentions of climate impact as a motif for not attending the conference in question.

If the conference in question would have allowed for virtual remote participation, survey respondents reported that they would have made use of this opportunity in order to: follow broadcast contributions (136 mentions), present own work as talk or poster (99 mentions), and discuss presentations in comments or forums (85 mentions). Comparably fewer people stated that they would have engaged in plenary or one-to-one video conversations (53 and 47 mentions, respectively). Only 29 respondents would have advertised job openings in forums or chat rooms. However, this number might reflect the relative scarceness of job opportunities more than the respondents' willingness to advertise those.

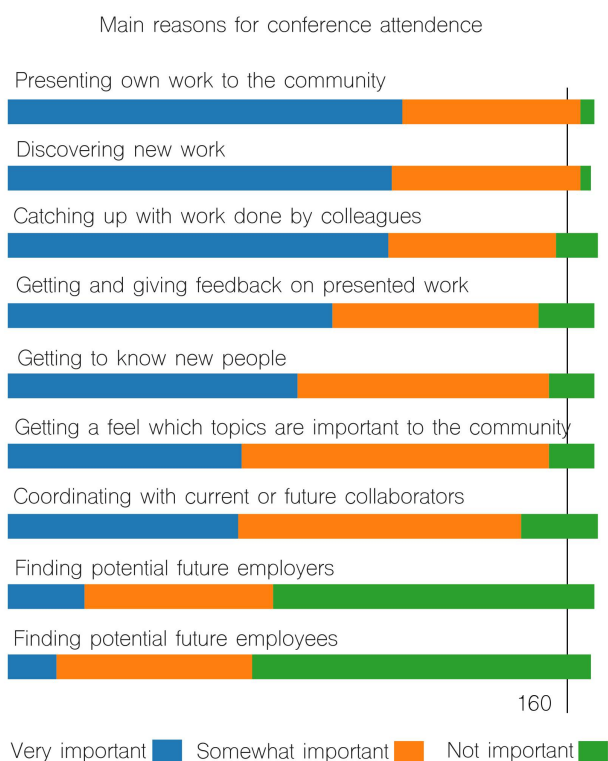


Figure 3: Main reasons to attend scientific conferences.

Evaluation of virtual conferencing solutions

The majority of survey respondents (66%) have never attended a conference that was delivered virtually or had a significant portion of remotely delivered content. Yet, 62% reported that they did participate in webinar series and the vast majority (92%) reports that they occasionally or regularly follow video lectures and/or recorded talks related to their field of work.

The majority of respondents (69 %) indicated that they would only be interested in attending a conference virtually at significantly reduced fee, while 14% were willing to pay a slightly reduced fee, and another 16% would only participate virtually if it were for free.

Participants were asked to evaluate (from strong advantage to strong disadvantage) the effects of virtual remote participation on different aspects of conferencing (Figure 4). The top advantages are money and time saving, the ability to reach a wider audience and to accommodate for conflicting professional or personal duties. All these advantages are deemed about equally strong. Additional comments added the following benefits: Reduced climate impact (31 mentions); no travel inconvenience or visa problems (5 mentions); longer availability of recorded talks (5 mentions); improved accessibility for delegates with disabilities (4 mentions).

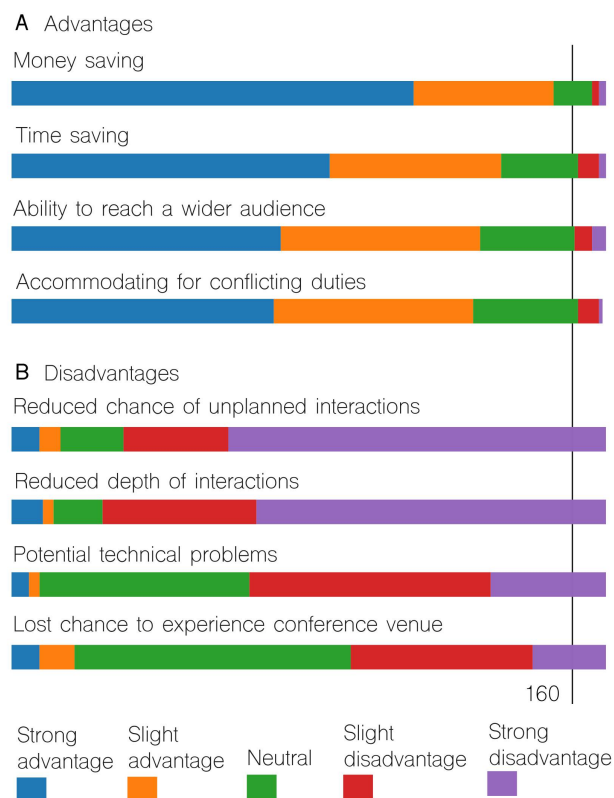


Figure 4: Advantages and disadvantages of virtual conferencing.

The top disadvantages are the lost chance of unplanned interactions, and reduced depth of interaction. Potential technical problems are deemed as less significant, as is the lost chance to experience the conference location. Additional comments added the following concerns: virtual respondents miss out on networking opportunities, including after hour activities (11 mentions); harder for virtual participants to fully commit to the event (10 mentions); remote presentations being less engaging for presential delegates (3 mentions); Another important concern was that broadcasting and recording of talks discourages presentation of work-in-progress and unpublished material, as well as open discussion about presented work.

The survey then asked respondents to evaluate how well virtual conference solutions can deliver the aspects of scientific conferences (from “much better” to “much worse”). Results are shown in Figure 5. As expected, virtual solution are deemed to perform well (in some cases even better than presential conferencing) when it comes to delivering technical content (seeing what is important to the community, discovering new work, presenting own work), but fall short when it comes to supporting social interaction (finding potential future employers or employees, coordinating with

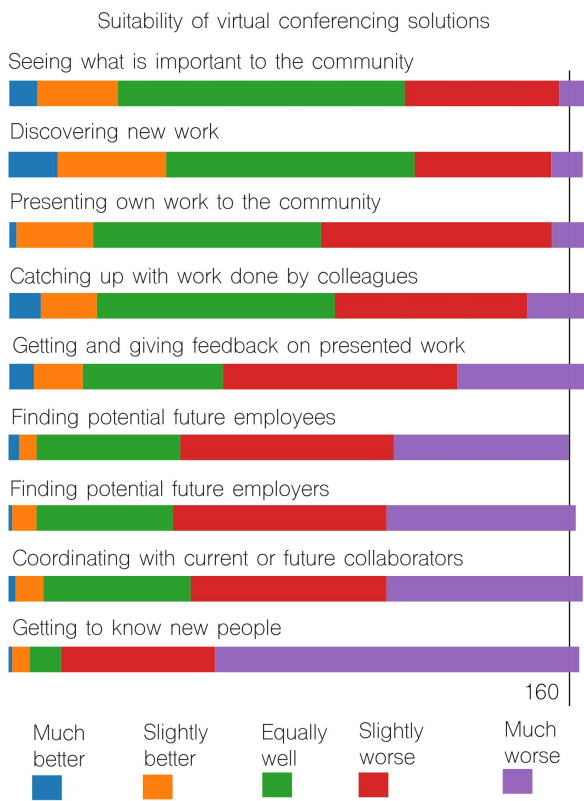


Figure 5: How well can virtual technologies deliver different aspects of conferences compared to presentational participation?

collaborators, and getting to know new people).

Interestingly, when crossing these answers with the ones to the question on the importance of various conferencing aspects, it can be concluded that the functions that are seen as most important (see Figure 3) are also deemed to be delivered equally well or only slightly worse by virtual conferencing technologies compared to presentational conference attendance.

The survey also included several open questions about respondents' experience with virtual conferencing technologies and asked what they saw that worked particularly well, what could have been improved where things went wrong, and where they see the biggest barriers that would need to be overcome in order to make virtual conferencing an equal or superior option. Feedback from these questions generally fell into three categories: technical, organizational, and social.

On the technical side, the software platform used for broadcasting and/or video conferencing is crucial as it has been responsible for many of the reported problems. If remote participants are to interact virtually with the presentational audience, virtual conferencing solutions should not require installation of proprietary clients, particularly if they

are not equally well supported on all major operating systems. Internet connection, both video and audio equipment, but especially the latter, need to be of sufficient quality to not impede the conversation. Similarly, the choice of lighting can significantly influence virtual conferencing experience. Solutions need to be supervised by well trained technical support, and organizers should provide alternative ways of access and/or backup recordings if part of the technology becomes unreliable. If all requirements are met, several survey respondents feel that virtually delivered or received presentations can actually be superior to presentational ones, as the former is also often affected by poor sound quality and visuals in the venue.

Regarding modes of presentation, respondents showed some divide as to whether live presentation or pre-recorded presentation leads to a better delivery of a remote talk. Live presentation is generally reported to be more engaging, but more susceptible to technical issues. In both cases, delivery of a talk to the camera is generally unsympathetic to people's audience-oriented delivery style, and the lack of audience feedback renders remote delivery less engaging. In a live setup, this might be addressed by streaming a feed of the audience to the presenter. However, lack of engagement is ultimately not a technical problem but is often due to the speaker himself not feeling present at the event. Several respondents advocate for pre-recorded presentations followed by live Q&A sessions.

For virtual live interaction, special focus needs to be placed on enabling interaction between presentational and virtual participants, as interactions can easily appear unsatisfying due to time lags and reduced communication bandwidth of virtual participation solutions. While some survey respondents advocate for questions and answers to be moderated by the chair (for example using technologies such as sli.do), other respondents prefer the opportunity for a direct dialogue with the presenter. A particular difficulty is to include virtual participants in workshops, where the focus lies on discussion rather than delivery of presentations. On the other hand, change between different interaction formats can also support the exchange of ideas and current affairs: Discussions that start in a Q&A session or virtual chat can readily be followed up in specialized discussion forums, and virtual break-out sessions can reconvene in plenary.

Independent of virtual participation, most respondents valued the recording and streaming of presentations, both for their potential to reach a wider audience as well as for their longevity. The latter being especially valuable as conferences with parallel tracks do not allow for participants to follow the whole programme. On the flip-side, wide and long-lasting availability of conference contributions can make participants less willing to present work in progress and unpublished results, which can have a detrimental impact on the quality of the discussion.

In line with the numeric evaluation shown in Figure 5,

the most mentioned biggest barrier that would need to be overcome in order to make virtual conferencing an equal or superior option to presential conferencing is the lack of serendipitous interactions, unplanned face-to-face conversations, and informal networking activities. This was most often expressed as the lack of “virtual hallway tracks”, “virtual lunches” and after hour activities. No respondent hinted towards any existing technology with a potential to enable such interactions to a meaningful extent. As a consequence of falling short in this important conferencing function, several respondents expressed their worry that opening traditional conferences for virtual participation might introduce a two-class scientific society, where virtual participants (those who are unable or unwilling to fly more, presumably mainly early career scientists) are disadvantaged compared to presential attendants.

A recurrent suggestion that might compensate for the lack of social interactions in virtual conferences was to host scientific events in several parallel local venues, which are then linked up using virtual conferencing technology. This form of conferencing is able to create an opportunity for full immersion, social networking and direct physical interaction, while still reducing the amount of long distance air travel.

Acceptance of sustainability measures

Finally, we asked participants to evaluate different measures that conference organizers might install in order to reduce the ecological impact of their event (Figure 6). Most respondents strongly or slightly welcomed the inclusion of virtual remote talks and poster presentations, despite all difficulties expressed in the previous sections. In general, respondents demonstrate a higher acceptance of *optional* measures and incentives (such as registration discounts for train travel), whereas *mandatory* measures (such as exclusively vegetarian and sustainably sourced catering) received lower sanctioning, particularly when connected with increased registration fees. Carbon offsetting was ranked equally low (with the optional measure being slightly more accepted than the mandatory one), likely due to its controversial benefit. Installing presential participation quota meets the lowest acceptance by survey respondents, particularly when correlated with travel distance or when quota are installed per research group. In general, all but these last two sustainability measures received more supportive than opposing evaluations.

Participants were more inclined to accept hard presential participation quota when virtual participation is notably discounted (Figure 7). 61% of the respondents would welcome enforced presential participation limits if connected with a significant registration discount and 72% would welcome the measure if virtual presentation were at no cost.

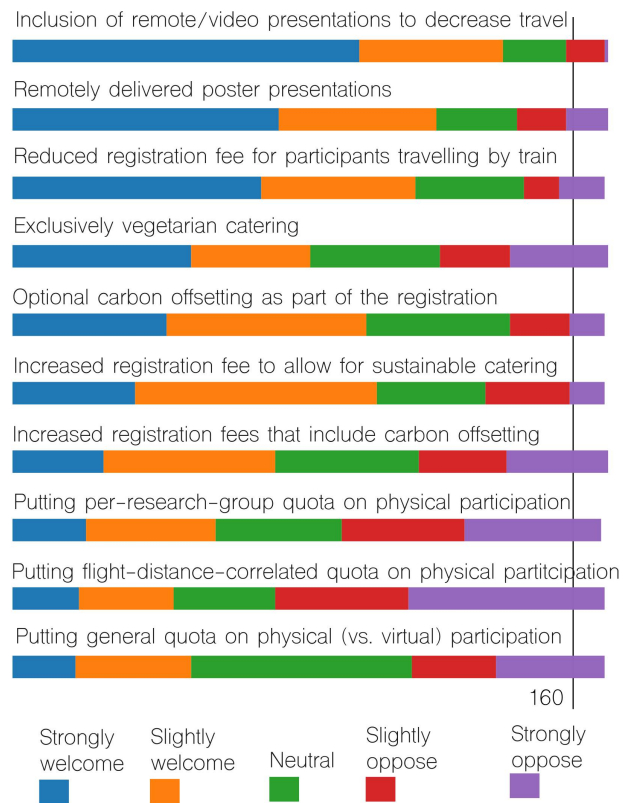


Figure 6: Acceptance of different sustainability measures for scientific conferences

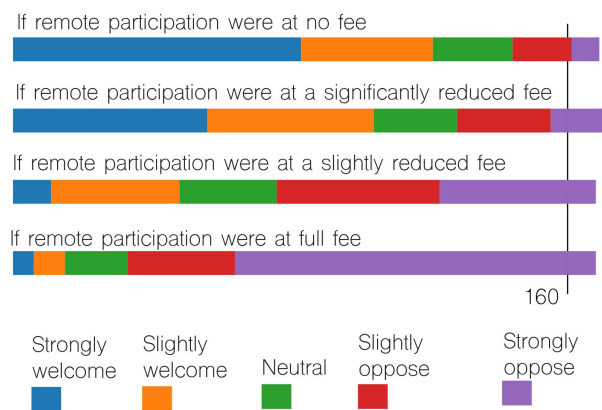


Figure 7: Acceptance of hard physical attendance quota for different registration schemes

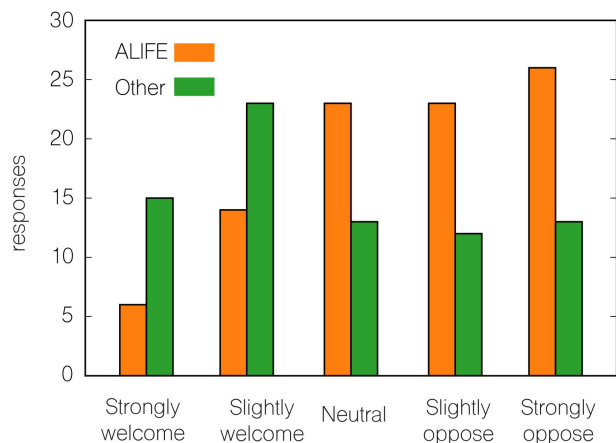


Figure 8: Different acceptance of flight-distance-correlated presential participation quota within the ALIFE community and other respondents.

Difference between the ALIFE community and other survey respondents

We have analyzed all survey results with respect to differences between responses from self-declared ALIFE community members against the rest of the respondents. For almost none of the questions could a significant difference be noticed.

The only exception is that ALIFE community members show a stronger opposition than other respondents towards conference organizers imposing flight-distance-correlated presential participation quota (Figure 8). This might be due to a number of reasons: Firstly, ALIFE is a small community that holds essentially one international conference per year. If people from certain origins are discouraged to participate presentially, they cannot be offered an alternative closer to their country of residence. Secondly, because the community is relatively small, there might be a stronger desire to physically co-locate all its members – something that is unfeasible for larger scientific communities. Thirdly, ALIFE is also a very diverse community. Co-locating its members might help to keep the community coherent and prevent the loss of exchange between subfields of Artificial Life. Lastly, we cannot exclude a sampling bias due to the way survey respondents had been recruited. If this were the case, it is nevertheless interesting that such bias shows most strongly when it comes to this particular sustainability measure.

Discussion

We have presented results from a community survey that assessed the acceptance of virtual conferencing technologies and other sustainability measures in the ALIFE community as well as the general scientific community. In summary, our results show that scientists are rather welcoming of almost all suggested measures that help us to lower the eco-

logical impact of academic conferences. Measures are more accepted when they are presented as incentives rather than compulsion.

The most positive feedback was on the inclusion of remote participation by means of virtual talks and poster presentations, as well as making these presentations accessible via streaming and/or broadcasting. This is very encouraging, as it is arguably the most effective measure to reduce the ecological cost associated with conferencing, *i.e.*, the impact of air travel.

At the same time, there is strong concern that virtual conferencing techniques can only deliver the scientific content that is presented, but fail to cover other important aspects of conferencing, primarily the opportunity for personal networking and the opportunity to fully engage in the conference. There is the associated fear that the spreading of virtual conferencing will deprive early career researchers from their professional opportunities. Quoting one of the survey respondents: “The young need to travel: for their own sake and for the sake of the elder.” While we fully agree with this sentiment, we point out that opportunities for travel are already constrained by boundary conditions such as travel budget limitations, competing professional or personal commitments such as child care obligations, or disabilities. Virtualization has the potential to make scientific conferences more inclusive by lowering these barriers. In a time, where a growing number of academics experiences a personal conflict between their ecological ethos and a perceived *duty* to travel, virtual conferencing technologies can empower academics to decide more freely *how often* and *how far* to travel, without sacrificing their engagement with the community entirely.

The enforcement of presential participation quota has received the lowest acceptance among respondents, both within the ALIFE community as well as the wider scientific audience. This is unfortunate as it is the only proposed measure by which conference organizers can *guarantee* that their event does not exceed some given ecological impact target. Yet, the survey results reveal that academics are more receptive for such enforcements, if alternative remote participation comes with significant monetary incentives or is offered for free.

The results of this survey will be taken forward, and will be closely considered in the organization of ALIFE 2019. As this year’s ALIFE conference will explore the theme “how Artificial Life can help solve societal challenges”, we are eager to trial a variety of solutions toward low-carbon conferencing in the hope to make a noticeable real-world difference. We understand that changing the habits of a community is an undertaking that can only show success, if done in line with the communities sentiments and agenda. Therefore, we regard our efforts merely as starting point of a long-term process that needs to be performed in discussion with the ALIFE community, the International Society of Artificial Life

(ISAL), as well as future conference organizers. The presented survey is the start of this discussion.

Acknowledgements

We wish to express our gratitude to all participants of our online survey for the time taken, the depth of their considerations, and the many ideas they have provided.

References

- Achten, W. M. J., Almeida, J., and Muys, B. (2013). Carbon footprint of science: More than flying. *Ecological Indicators*, 34:352–355.
- Cobb, K. M., Kalmus, P., and Romps, D. M. (2018). AGU should support its members who fly less.
- Girling, J., Liese, P., Dance, S., Gerbrandy, G.-J., Kyllönen, M., Eickhout, B., and dOrnano, M. (2018). Co2 emissions from aviation. Technical report, European Parliament.
- Glover, A., Strengers, Y., and Lewis, T. (2017). The unsustainability of academic aeromobility in australian universities. *Sustainability: Science, Practice and Policy*, 13(1):1–12.
- Hasan, M. (2018). New university rules encourage scientists to avoid air travel. <https://www.wired.com/story/climate-scientists-take-the-train/>. Accessed: 2019-03-09.
- IPCC (2018). Summary for policymakers. In *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. World Meteorological Organization, Geneva, Switzerland.
- Krumdieck, S. and Orchard, S. (2011). Signs of change national networked e-conference: Highlighting emerging sustainability and social business. *Social Business*, 1(1):37–58.
- Lee, D. S., Fahey, D. W., Forster, P. M., Newton, P. J., Wit, R. C., Lim, L. L., Owen, B., and Sausen, R. (2009). Aviation and global climate change in the 21st century. *Atmospheric Environment*, 43(22-23):3520–3537.
- Leybold-Johnson, I. (2019). Should academics be taking the plane for short trips? https://www.swissinfo.ch/eng/university-policies_should-academics-be-taking-the-plane-for-short-trips-/44726904. Accessed: 2019-03-09.
- Nathans, J. and Sterling, P. (2016). Point of view: How scientists can reduce their carbon footprint. *ELife*, 5:e15928.
- Nature editorial (2015). A clean, green science machine. *Nature News*, 519(7543):261.
- Nevins, J. (2014). Academic jet-setting in a time of climate destabilization: Ecological privilege and professional geographic travel. *The Professional Geographer*, 66(2):298–310.
- Wynes, S. and Donner, S. D. (2018). Addressing greenhouse gas emissions from business-related air travel at public institutions: a case study of the University of British Columbia. Technical report, Pacific Insitute for Climate Solutions.

Suppleness and Open-Endedness for Social Sustainability¹

Hiroki Sayama

Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, USA
Waseda Innovation Laboratory, Waseda University, Japan
sayama@binghamton.edu

Abstract

One of the research questions in ALife that could contribute greatly to social sustainability issues is how dynamic meta-states of a complex system may be sustained through continual adaptive changes, or *suppleness* (Bedau, 1998). The idea of sustainability by suppleness is fundamentally different from conventional ideas of sustainability by robustness or resilience, and it is directly linked to *open-endedness*, a topic that has recently attracted significant attention in the ALife community (Taylor et al., 2016). Understanding and implementing mechanisms of suppleness and open-endedness may provide novel perspectives of many of today's socio-economic, socio-ecological and socio-technological problems that call for new strategies to cope with inevitable environmental/contextual changes. This short essay provides a non-exhaustive list of research questions on this topic and encourages ALife researchers to play a leading role in this interdisciplinary collaboration endeavor.

Sustainability of a dynamical system is often conceptualized and characterized as either robustness or resilience in the literature. Robustness implies persistent stability, rigidity and outright tolerance of a system against perturbations. A robust system's state remains at or near a stable fixed point without much deviation from it (Fig. 1(a)). The mechanism that realizes sustainability of a robust system is a strong, quick negative feedback which eliminates any small perturbation before it grows and takes the system away from its normal state. There is little room for adaptation in a robust system.

Resilience implies a more elastic, longer-term stability than that implied by robustness. A resilient system's state may undergo large drifts from the normal one occasionally, but temporary accommodation and gradual recovery eventually bring the system back to normal (Fig. 1(b)). The mechanism that realizes sustainability of a resilient system is elasticity (i.e., capability of temporary accommodation) and restoration (i.e., capability of gradual recovery). Like robustness, resilience is based on goal-oriented negative feedback, which is much slower and more moderate than those of robust systems.

As seen above, robustness and resilience are, in a sense, similar – robust systems may show resilience at a finer scale, while resilient systems may show robustness at a coarser scale. The distinction between the two is partly based on the scale of observation and control being used.

The sustainability of real biological or ecological systems, however, is not necessarily based on either robustness or resilience. Although those systems can definitely be robust and resilient, their states never remain at a single place in a long term. Instead, they continue to change dynamically due to complex interaction and evolutionary adaptation of system components. What is sustained over a long period of time in those systems is not a specific normal state, but the ability to demonstrate a unique, dynamic, emergent meta-state, such as “alive”, “active”, “productive”, and so on. Some of those meta-states may be readily quantifiable and measurable, while others are more qualitative and may be hard to quantify.

By shifting our focus from the system's specific desired

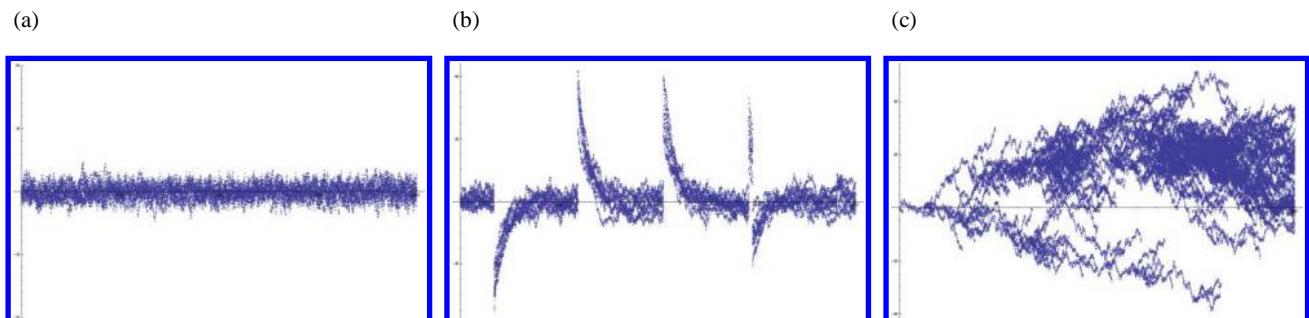


Figure 1: Illustrative visualizations of three different forms of sustainability. (a) Robustness. (b) Resilience. (c) Suppleness. Horizontal axes show the progress of time, while vertical axes show the system's states represented by the aggregate states of its components.

¹ This work is largely based on the author's earlier white paper submitted to the US National Science Foundation (<https://bit.ly/2HmwWkz>), with revisions and addition of discussions on the linkage to open-endedness.

state to its dynamic meta-state, we may consider yet another characterization of sustainability, which we call *suppleness*. Bedau (1998) defined supple adaptation as “an automatic and continually creative evolutionary process of adapting to changing environments”. We can conceptualize suppleness as a novel form of sustainability fundamentally different from robustness or resilience discussed above. It may inform us about the sustainability of our social systems from a different perspective.

Suppleness implies the maintenance of a dynamic meta-state of a system by continual adaptive changes of the system (and its components), where the system’s actual state drift greatly and may not even come back to the original place in its state space (Fig. 1(c)). There is no strong negative feedback to a normal state in action, while inputs are continuously provided to the system from an ever-fluctuating environment. Since there is no reference point to converge towards, the dynamics of a supple system are necessarily open-ended, which may result in divergent, exploratory, or even creative processes unless the environment is too selective. Understanding and creating such open-ended processes is now widely recognized as one of the major challenges in the ALife community and the AI community at large (Taylor et al, 2016; Stanley, 2018).

The concept of suppleness and its significance for the maintenance of dynamic meta-states of complex systems have already been discussed in various forms. The Red Queen principle discussed in evolutionary biology (Van Valen, 1973) is a classic example, which illustrated the need of continual adaptation for a dynamic evolutionary system to survive in co-evolutionary environments.

Many of today’s socio-economic, socio-ecological and socio-technological problems arise in rapid environmental/contextual changes, such as global warming, economic volatility and political instability. In the presence of such large-scale changes, an attempt to bring the system back to its original state, which is the central idea of sustainability by robustness or resilience, may no longer be a viable option from a practical viewpoint. Instead, these problems call for new strategies to cope with inevitable environmental changes.

Here we argue that the concept of suppleness (and open-endedness) discussed above will have significant implications for today’s social problems that are all situated in rapidly changing environments. Meanwhile, applications of the concept of suppleness to social sustainability issues has not fully developed yet in either science or practice. There are a number of open questions to be investigated. Here are some examples:

1. How can suppleness be modeled theoretically/mathematically? How to formally define sustainability by suppleness?
2. What are appropriate, meaningful, and/or useful ways of characterizing dynamic meta-states sustained in a supple system?
3. What are the environmental conditions for suppleness to be an effective strategy for the system’s sustainability? Does our current natural and social environment meet those conditions?
4. What are potential risks of being supple?

5. Are all biological systems supple, or are there exceptions?
6. Among a variety of different social, political and economic systems human being has developed in its history, which are suppler than others? Are/were they functioning effectively or not?
7. Can suppleness help improve creativity, decision making and problem solving by human individuals or groups?
8. Has the recent development of IT and social media made our society suppler or not?
9. Rapid concentration of resource and information in a small number of entities is ongoing in many scenes of our society. Is it for or against social suppleness?
10. Is there any fundamentally different form of social mechanisms (policies, technologies, customs, etc.) that could significantly enhance social suppleness?
11. How can we make the financial systems suppler to prevent further financial crises from happening?
12. How can we make the social infrastructure (e.g., power grid, water supply, transportation, the Internet) suppler?
13. What are the implications of suppleness for global climate change?
14. How can we improve the current political decision making mechanisms using the concept of suppleness?
15. Is the current form of science and engineering supple enough to promote new discovery and innovation? If not, what should be done?

These research questions are by no means intended to be exhaustive. Readers are encouraged to explore other directions of inquiry and come up with more questions and answers.

We believe ALife and other related disciplines have a lot to offer to the research on suppleness and open-endedness for social sustainability. Models and simulations of nonlinear dynamical systems, evolutionary systems and agent-based systems will likely play an essential role in driving this research. Equally important will be evolutionary biology and mathematical biology, where numerous models of evolutionary adaptation have already been developed and studied extensively. Moreover, insight into practical social applications will be obtained from management and organizational sciences that study innovation, creativity, decision making and social networks. ALife researchers can and should play a leading role in promoting such interdisciplinary collaboration endeavor.

References

- Bedau, M. A. (1998) Four puzzles about life. *Artificial Life* 4: 125-140.
- Stanley, K. O. (2018) Answering the call of open-endedness. In *Artificial Life Conference 2018 Proceedings* (pp. 7-7). MIT Press.
- Taylor, T. et al. (2016) Open-ended evolution: perspectives from the OEE workshop in York. *Artificial Life*, 22(3), 408-423.
- Van Valen, L. M. (1973) A new evolutionary law. *Evolutionary Theory* 1: 1-30.

Cities, a conceptual framing for a synthetic perspective

Jesús M. Siqueiros-García¹

¹IIMAS-Unidad Mérida, UNAM, México
jmario.siqueiros@iimas.unam.mx

Abstract

Cities are pervasive and they are some of the most powerful ecosystems of the Anthropocene. They have been around for almost 10000 years, and by the year 2050 most humans will be living in a city. Although it is known that cities have an impact to different scales, from the very local to the whole Earth System, its implications are far from being understood. Studying cities as organismic systems has been a productive strategy and favourable to complex adaptive systems analyses. However, the organismic view is to a great extent metaphorical, focusing exclusively on human activity, instead I argue for an approach that actually considers life processes as constitutive to them. In this extended abstract I suggest a conceptual framing for a synthetic approach to cities in which life processes are paramount for their understanding. Specifically, I will focus on two aspects: 1) the human-teleological component of cities and 2) the role of life processes organisationally closing the city, and bringing forth a self-generated unity and identity and the conditions for its own evolution. I believe that due to the increasing interest of the ALife community in tackling social issues, ALife unique insights and methods can be of great value in understanding cities and dealing with the social-ecological challenges they pose. A definition of cities from a synthetic perspective can help the ALife community to put into action its epistemic arsenal.

Introduction

Cities are equally fascinating and highly challenging processes to academics and policy makers. Their awe is well justified because cities display complex dynamics, affecting their inhabitants –humans and non-human, with an immense impact on their surroundings and contributing to global change at the level of the biosphere (Mills, 2007). However, given the current environmental conditions, understanding cities should also help to steer them towards more sustainable pathways and to be resilient once they're on track (Romero-Lankao et al., 2016).

At present, it is quite common to approach cities from a systems and integrative perspective (Bretagnolle et al., 2009). For example Sustainability science in general conceptualize cities as social-ecological systems with complex adaptive dynamics. It is important to acknowledge those dynamics as traversing the human-ecological dimensions, be-

cause it re-frames the we understand our relation with the biosphere. It also points directly to the need to scientifically investigate this “epistemic object” under new light with the double objective to understand it, as well as to steer it. However progressive this vision, it still falls short since the system is seen as a composite of the social with the biophysical –with some exceptions such as Alberti (2016)-. Also, although it acknowledges it, it struggles in integrating the human intentional and goal-directed behaviour as constitutive to the city as a social-ecological system. The ALife community is no stranger to either Life and teleology. With its epistemic and methodological emphasis on *synthesis*, it seems a good candidate to suggest an integrative view of cities that can properly address such complications. In this sense, the concept of urban system here advanced is that of an *ecosystems mediated by human-built material –green, blue and grey¹- life-worlds that are functionally integrated (organisational closure) by multi-scale life processes –e.g. primary production, nutrient cycling, and niche construction among others.*

Human material life-worlds, niche construction and the city organisational closure by life-processes

No doubt, cities are a collection of purpose and designed material transformations that cannot be neglected in any definition of an urban system. As any other organism, humans build their niches. Humans transform their material world into blue and green, and in the case of cities, into grey infrastructure. Those transformations can be intentional –with added unintentional consequences-, and through processes of reflexivity, design itself is object of transformations and intentions.

¹In very simple terms, green infrastructure refers to ‘nature-based’ transformations to the urban landscape, such as parks, tree rows, and vertical gardens. Blue refers to water-based infrastructure like city lakes and constructed wetlands. Green-blue infrastructure networks –as sometimes are referred- are important because they conserve ecosystem values and functions. Gray infrastructure would include things like buildings, roads, pipes, power networks.

Yet, human-environment interactions are not unidirectional. Such relationships are both ways in which the environment contributes substantially to the organism-environment dynamics. The environment imposes a set of constraints and it offers a space of possibility, both defined by its materiality, from its physics to its geo-bio-ecological processes. This is a relationship known as niche construction: the organism modifies the environment that in turn will modify it back (Laland et al., 2016).

The interaction implied in the process of niche construction grounds the idea that an *environment is not simply the physical properties of things as now conceived by physical science. Instead, they are ecological, in the sense that they are properties of the environment relative to an animal or organisms in general* (Reed and Jones, 1982). When the interaction is meaningful to an organism, then we are speaking of a life-world. In other words, a life-world is the world that is meaningfully experienced by an organism and its community. *Cities are ecosystems that are possible as they revolve around a human-built material life-world.* This life-world, due to its materiality transcends individuals & generations in a way that it becomes part of the process known as niche inheritance, that scaffolds the reproduction of practices, traditions, and culture.

The material life-world becomes the axis without which the urban ecosystem wouldn't be possible. That is because human induced transformations to the environment become an opportunity for other organisms to create their own niche, in this sense, niche inheritance also transcends humans and affects other organisms.

As humans develop their life-world, other organisms do as well. Such life-worlds can be conceptualized as niches. In a word, a niche is more than a habitat it is also the way an organism lives it. A niche then, is formed by the network of opportunities for action –also known as *affordances*- that an environment offers to an organism in relation to the organisms exploratory modes and skills, therefore a niche is always meaningful to an organism (Reed and Jones, 1982; Heras-Escribano and De Pineado-García, 2018).

I suggest that niches also bundle together forming a web. As organisms modify their environment they create opportunities for other organisms niche construction processes, e.g., the cables of urban energy supply –a human modification to the environment- affords squirrels to move all over the city. The central idea is that niches, by connecting multiple levels of organization, they channel the flows of energy, matter and information and sustain and integrate the system as a whole.

According to Mossio and Moreno (2010), organisational closure is about how a system assures the conditions for its own production. In the case of ecosystems in general, organisational closure is attained through constant production of a web of niches. The ongoing production of such web can be seen as a constitutive *life process* that generates the conditions for the continuation of its own unity and identity, as

for its components and system level evolution.

Conclusions

Human-built material life-worlds –green, blue and grey- and closure by life-processes bring cities to life, . . . literally. There is an asymmetry that has to be acknowledged openly: when compared to the actions of other organisms, human material life-worlds impose the dominant constraints on flows of energy, matter & information that sustain the city as an ecosystem; that is why these are among the most important ecosystems in the *Anthropocene*. On the other hand, cities would be sterile and inert if it were not because of its dynamicity brought by the complex web of interdependent life forms, creating and connecting niches.

I believe that the ALife community has too much to offer to tackle pressing societal and environmental problems, such as those present in urban contexts. In Langton's line of thought, I believe that social, cultural, economic or political phenomena are a continuation of Life. The operationalization of this idea is highly relevant if we are interested as a community in ALife becoming part of the solution. This is why I am interested in advancing such a primitive and highly speculative definition of cities with the hope of bringing closer to the ALife community the problem of cities as meaningful complex geo-eco-systems.

Acknowledgements

I would like to thank Ximena González for her comments and observations.

References

- Alberti, M. (2016). *Cities that think like planets : complexity, resilience, and innovation in hybrid ecosystems*. University of Washington Press.
- Bretagnolle, A., Pumain, D., and Vacchiani-Marcuzzo, C. (2009). *The Organization of Urban Systems*, pages 197–220. Springer.
- Heras-Escribano, M. and De Pineado-García, M. (2018). Affordances and landscapes: Overcoming the nature-culture dichotomy through niche construction theory. *Frontiers in Psychology*, 8.
- Laland, K., Matthews, B., and Feldman, M. W. (2016). An introduction to niche construction theory. *Evolutionary ecology*, 30:191–202.
- Mills, G. (2007). Cities as agents of global change. *Int. J. Climatol.*, 27:1849–1857.
- Mossio, M. and Moreno, A. (2010). Organisational closure in biological systems. *Hist. Phil. Life Sci.*, 32:269 – 262.
- Reed, E. and Jones, R. (1982). *Reasons for Realism: Selected essays of James J. Gibson*. Lawrence Erlbaum: Hillsdale, NJ.
- Romero-Lankao, P., Gnatz, D., Wilhelmi, O., and Hayden, M. (2016). Urban sustainability and resilience: From theory to practice. *Sustainability*, 8:1224.

Quantifying affordances through information theory

Miguel Aguilera^{1,2}, Iñigo Arandia-Romero² and Manuel Heras-Escribano¹

¹ IAS-Research Center for Life, Mind and, Society, UPV/EHU, University of the Basque Country, Spain.

² ISAAC Lab, Aragón Institute of Engineering Research, University of Zaragoza, Zaragoza, Spain.
sci@maguilera.net

Abstract

Affordances are directly perceived environmental possibilities for action. Born within ecological psychology, they have been proposed to be one of the main building blocks to explain cognition from an embodied and situated perspective. Despite the interest, a formal definition of affordances in information theory terms that would allow to exploit their full potential in models of cognitive systems is still missing. We explore the challenge of quantifying affordances by using information-theoretical measures. Specifically, we propose that empowerment (i.e., information quantifying how much influence and control an agent has over the environment it can perceive) can be used to formally capture information about the possibilities for action (the range of possible behaviors of the agent in a given environment), which in some cases can constitute affordances. We test this idea in a minimal model reproducing some aspects of a classical example of body-scaled affordances: an agent passing through an aperture. We use empowerment measures to characterize the affordance of passing through the aperture. We find out that empowerment measures yield a similar transition to the one found in experimental data in humans in the specialized literature on ecological psychology. The exercise points to some limitations for formalizing affordances and allows us to pose questions regarding how affordances can be differentiated from more generic possibilities for action.

Introduction

Ecological psychology and affordances

According to ecological psychology, affordances are possibilities for action that we directly perceive in our environments. These possibilities for action emerge from the relations between the capacities of agents and elements of the environment. The term ‘affordance’ was coined by J. J. Gibson and refers to the complementarity of organism and environment (Gibson, 2014, 119).

Ecological psychology is an embodied, situated, and non-representational approach to cognition that presents itself as an alternative to behaviorism and cognitivism. The way ecological psychology traditionally analyzes perception is not by focusing on how nervous systems work, but on how organisms explore the environment so as to find opportunities for acting (the above-mentioned affordances). In this view,

affordances are key objects of perception from an embodied and situated approach to cognitive science, since they are relations between the bodily capacities of agents and certain environmental elements (Chemero, 2009; Heras-Escribano, 2019). These relations can sometimes be mathematically quantified or measured in agent-related units (Warren, 1984; Warren and Whang, 1987), which means that it is possible to offer a scientific account of these relations that can be useful for other embodied and situated approaches (the enactive approach, dynamical systems theory, etc.).

While affordances are defined as ‘possibilities for action’, not all possibilities for action are affordances. For example, since affordances need to be perceived, activities like abstract thinking or calculating are not considered as such. On the other side, not all perception is of affordances: colours or sensations are not affordances either. Thus, affordances are opportunities for action in the sense that certain aspects of the environment that we perceive are in relation to our movement or to our bodily features, like a bottle to grab, a doorknob to turn, or a door to pass through. In the following sections, we will call ‘possibilities for action’ or ‘range of possible actions’ to those movements that the agent can perform (exploration, deambulation, etc.). Note that this range of movements may include those that the agent performs when taking advantage of a directly perceived affordance.

Affordances are usually classified either as action-scaled or body-scaled affordances (Fajen et al., 2009). Action-scaled affordances are those opportunities for acting that depend on the behavior of the agent. For example, how fast an organism moves in a particular location determines the possibilities for acting for that organism (e.g., whether the organism is capable of avoiding moving obstacles). On the other hand, body-scaled affordances relate a particular dimension of the body of an organism to a particular environmental element. Although there are many examples of body-scaled affordances in the literature, here we are going to focus on the contributions of W. Warren and colleagues (Warren, 1984; Warren and Whang, 1987) for describing experimentally the affordance of climbability and that of passing or walking through apertures.

A particular aspect of Warren's contributions is that they explain how we perceive these affordances from an ecological perspective. Based on the active capacities of the organism as a whole, using an agent-related metric, they highlight the relational or complementary aspect of the organism-environment engagement.

Warren (1984) provided one of the first experimental accounts of the body-scaled affordance of *climbability*, one that allows organisms with legs to step on obstacles. He arrived to the conclusion that a step is climbable for a human if its height is less than 0.88 times the height of that human's leg regardless their body size (Heras-Escribano, 2019, 53). This way of describing that a step is climbable for a human emphasizes the agent-environment interaction inasmuch as it does not rely on absolute metric (meters, inches) for explaining in mathematical terms the affordance of climbability, but on agent-related metric; that is, the specific leg size of each agent.

Warren and Whang (1987) used the same agent-related metric for explaining how humans perceive the affordance of walking or passing through apertures. They determined that an agent can perceive the possibility of walking through an aperture if the aperture is at least 1.3 times the human's width for shoulder rotation regardless their body size (Warren and Whang, 1987, p. 381). Again, the authors offer an agent-related metric for quantifying the possibility for acting of a particular agent.

Exploring affordances with information theory

Although there is experimental evidence in favor of the quantification of affordances, there is still an open debate regarding their ontological status.¹ This has certain impact in the scientific practice, since there is no unique or single formal definition of affordances that is widely shared by the scientific community. In particular, a quantitative formal definition of affordances suitable to be applied to modelling simulations is lacking.

Here, we explore the challenge of interpreting affordances formally by characterizing the possibilities for action that

¹There is a long debate regarding the ontological status of affordances. For example, Reed (1996) considered that they were aspects of the environment that exert selective pressure, while Chemero (2009) claims that affordances are relations between the abilities of agents and some environmental aspects. Here we offer a neutral approach to the ontology of affordances, although we could claim that affordances here could be understood as being part of a dispositional pair that include the agent's abilities and certain environmental aspects. This dispositional approach makes it partially compatible with Chemero's (2009) approach because an ingredient of a dispositional approach is a relation as such, but a disposition is more than a relation: it includes a dimension of actualization that alters the organism-environment system, which with time it may lead to unexpected changes that may affect the history of the system (Heras-Escribano, 2019; Heras-Escribano and De Pinedo-Garca, 2018). For further discussions on the ontology of affordances, see (Heras-Escribano, 2019, Chapter 3)).

constitute affordances using measures from information theory. Ecological information is crucial for the notion of affordance, as it allows embodied agents to perceive affordances. Take, for example, the case of vision: from an ecological perspective, light becomes ecological information when it forms a heterogeneous structure or pattern caused by the reflections and reverberations in the environment. Regarding the source of light and the position of the agent, there are differences of intensity in that pattern that, in turn, produce differences in what the organisms perceive (Gibson, 2014, 45-46). This structure of ecological information shows the agent the available affordances of the environment as the agent explores it (Heras-Escribano, 2019, 30-45). Given the complexity of the light patterns and the variety of elements and variables that we can find in our natural environment, it is almost impossible to reproduce in the exact same terms all these aspects of ecological information and affordances in modelling simulations. However, we can reproduce some minimal aspects of it in information-theoretic terms (as when objects get bigger as they get closer to the visual field of the modelled agent, for example).

We aim to explore what an information-theoretic interpretation of affordances would imply, and what possibilities and limitations are entailed by such an interpretation. This exploration will use minimal models reproducing some aspects of well-known experimental setups, which means that what we offer here is a simplification of real life situations that also lacks of several aspects taken into account within the ecological approach. Nevertheless, we think such models are still able to offer both some of the relevant aspects about the proposed problem and important insights about the idea of interpreting affordances in information-theoretic terms.

Gibson claimed that Shannon's information theory (Shannon, 1948) was inadequate to capture ecological information (Gibson, 2014, 231-232). Nevertheless, we understand that what he rejected was a particular version of information theory that included the channel metaphor, because he interpreted that it implied a tacit commitment to representationalism and cognitivism. In addition, since Shannon's information measures are based on correlational regularities between variables (they cannot directly capture causal relations), they seem to be limited for measuring ecological information about causal effects of an agent's actions. However, there are modern formulations of information theoretic measures that are more adequate to describe possibilities for action in an agent. In this vein, we suggest that an interventionist notion of causality in the sense of Pearl (2009) and the notion of causal information flows (Ay and Polani, 2008) are better suited for characterizing ecological information.

Specifically, we propose that there is an information-theoretic measure that is able to capture possibilities for action that constitute affordances: the measure of empowerment. Empowerment (Klyubin et al., 2005; Salge et al.,

2014) is defined as the channel capacity between an agent’s actuators A for a window of time of size n starting at time t , and its own sensors S at time $t + n$

$$\mathfrak{E} = C(A_{[t:t+n-1]} \rightarrow S_{t+n}) \quad (1)$$

$$\equiv \max_{p(a_{[t:t+n]})} I(A_{[t:t+n-1]}; S_{t+n}) \quad (2)$$

where $A_{[t:t+n-1]} = \{A_t, A_{t+1} \dots A_{t+n-1}\}$ and $p(a_{[t:t+n]})$ is a probability distribution of the possible states of $A_{[t:t+n-1]}$. $I(X; Y)$ is the mutual information between X and Y , which is defined in terms of entropy and conditional entropy of the variables

$$I(X; Y) = H(Y) - H(Y|X) = H(X) - H(X|Y) \quad (3)$$

$$H(X) = - \sum_{x \in X} p(x) \log p(x) \quad (4)$$

$$H(X|Y) = - \sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log p(x|y) \quad (5)$$

In a deterministic world, i.e., one where each action leads to one specific outcome, we have that $H(S_{t+n}|A_{[t:t+n-1]}) = 0$ and empowerment can be simplified as:

$$\mathfrak{E} = \max_{p(a_{[t:t+n-1]})} H(S_{t+n}) = \log |S_A| \quad (6)$$

where $S_A = \{s_{t+n} \in S | \exists a_{[t:t+n-1]} \in A : p(s_{t+n}|a_{[t:t+n-1]}) \geq 0\}$ is the set of different sensor states that can be reached from time t to time $t+n$ with all possible combinations of available actions.

In this article, we propose that empowerment measures can be used to measure possibilities for action that compose affordances, and we test this intuition in a model that replicates the main results from experiments of perception of passability through apertures from Warren and Whang (1987). Still, it should be noted that, in general, empowerment measures are defined by a single value quantifying the possibilities available for an agent. Specifically, empowerment directly quantifies the possibilities for action as the average number of bits that an agent’s sensors can perceive as results of its own actions. In contrast, affordances are defined as sets of possibilities related with different objects or parts of an environment. In our comparison with Warren and Whang (1987) this is not a problem, since only one affordance is present and changes in the sensor state of the agent will be related to it. Thus, the mapping between empowerment and affordance becomes straightforward in this case. However, this issue becomes problematic when there is more than one affordance and when the environment has more elements not related to the affordance under study (as the experiments in Figure 4 will show). A more precise relation between empowerment measures and sets of multiple affordances are left as future work, though measures of

context-dependent empowerment (Salge et al., 2014; Klyubin et al., 2008) could be used to define sets of possibilities for action related to specific objects in an environment.

Model

Warren and Whang (1987) describe a classical example of body-scaled affordance by using the ability of an agent to perceive whether or not it can pass through an aperture. This ability is related to the capacity of a subject to be sensitive to the relationship of its own body to the objects in its environment. In this work, experimental evidence shows a transition in the perception of ‘passability’, which is related to the width of an aperture relative to the body of an agent. This transition takes place in humans approximately when an aperture is at least 1.3 times the shoulder width of a person. One of the results of the study, adapted in Figure 2 (top), represents the rate in which subjects with *small* and *large* body sizes judged an aperture of width W as passable (impassable in the original data). The data (Figure 2, top left) shows that each group displays a transition for a different value of W . Nevertheless, when the aperture width is normalized by the subjects’ shoulder width, then Figure 2 (top right) displays an invariant transition for both groups around $W/S = 1.3$. This result supports the hypothesis that scale-invariant critical points govern the perception of such affordances.

Furthermore, the authors hypothesize that subjects use eye-height information related to the ground as a source of intrinsic information about their own size to directly perceive the affordance of an aperture from a distance. In an experiment using an Ames room, in which the effective eye-height is distorted, Warren and Whang (1987) show that the self-perceived eye-height is determinant for characterizing the location of the transition in Figure 2 (top).

A minimal model of passability

In this paper we design a minimal model to represent the task described by Warren and Whang (1987), with the aim to quantify the information about the affordances available to an agent by using empowerment measures. In previous work, Slocum et al. (2000) modelled a minimal agent which was evolved to pass through openings wide enough to accommodate its body while avoiding too narrow openings. Agents used proximity sensors in the form of rays that produce a sensor input when intersecting with an object.

Inspired by this work, we present a model (Figure 1) with an agent that can move inside a room with a wall that shows an aperture of width W on one side. The agent is shaped as a rectangular prism of height H and width S , and its length and width are equal. We test agents with different widths, but we always maintain a human-like height-width ratio of $H/S = 4$. In contrast with the model by Slocum et al. (2000), we do not model the neural system of the agents, since it is not necessary for empowerment measures.

Instead, we design different actions $a(t)$ available for the agent: move forward, move back, move left, move right or stand still. The agent moves in discrete time, moving a distance $s = 0.2$ in the appropriate direction each step. The agent cannot move through walls. As we are interested in studying the perception of the affordance, we restrict the analysis to the inside of the room and assume that the agent never crosses the aperture.

The agent projects seven sensory rays from its upper front, with angles evenly distributed in the horizontal plane between $[-\frac{\pi}{4}, \frac{\pi}{4}]$ (Figure 1, main). The sensors receive a binary input, being active when the ray intersects with a wall, and inactive otherwise. In the example shown in Figure 1, where the three sensors in the right pass through the aperture (0) and the remaining four sensors in the left intersect the wall (1), the sensor state is (1, 1, 1, 1, 0, 0, 0). If the agent moves left, the right sensors will change their state from 0 to 1, until all rays collide with the wall. If it moves right, the left sensors will change from 1 to 0, while the right sensors will switch from 0 to 1 when they hit the right side of the wall. The lateral plane (Figure 1, right box), shows that sensors are projected from the upper front edge of the agent to the ground of the aperture, to represent the effect of the eye-height of the agent in its sensory input, retrieving intrinsic information necessary for performing the task as reported by Warren and Whang (1987). In practice, this implies that the distance traveled by the ray will be of $\sqrt{d^2 + H^2}$, where d is the horizontal distance between the agent and the aperture. Thus, note that the available sensory configurations (and therefore the values of empowerment) will change depending on the height of the agent, since for any agent objects will always be seen from a distance larger than H . This is included as a way for the agent to have perceptual information about its own size in its patterns of sensor activation. Warren reported height information as critical for perceiving passability, and we will see here that it has an important effect.

Empowerment of the agent $\mathcal{E}(n)$ is computed as follows: an agent starts from a specific location and it is allowed to move n steps. Then, for all the positions available within a distance of n steps the sensory input received at each final location is recorded. Then, following Equation 6, the empowerment of the agent in a deterministic world is equal to the logarithm of the size of the set of perceived inputs.

Results

In order to explore the relationship of empowerment with the size of the agent and the affordances of the environment, we perform two experiments. In the first experiment, we reproduce the results reported by Warren and Whang (1987). Measuring the empowerment of the agent for different sizes of the aperture and different body-sizes of the agent we find a scale-invariant transition very similar to the one shown by Warren and Whang (1987). In the second experiment we

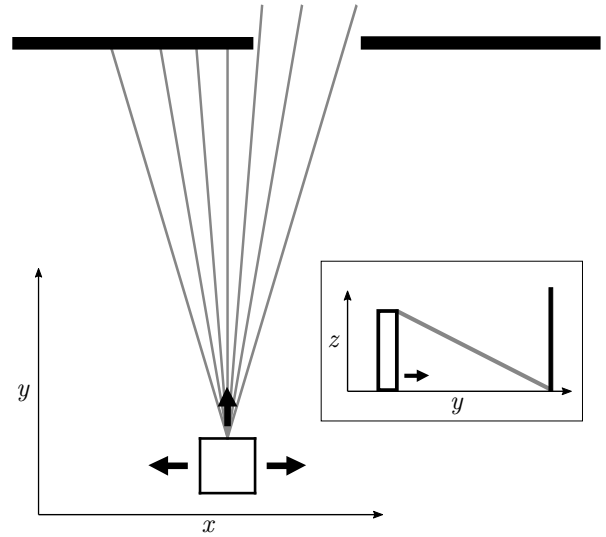


Figure 1: Schema of the agent. An agent interacts with an environment consisting on a wall with an aperture. The agent can perceive using binary sensors shaped as a set of rays arranged in a beam. Each step, the agent can move forward, left or right, or stand still. In the case of the vertical agent (right plot), the rays are not perpendicular to the floor but they are emitted from the top of the agent's body at height H .

compute the empowerment field of an agent in an environment with two apertures, one larger than the agent and thus passable, and another one smaller that the agent cannot go through. In this way, the empowerment can illustrate how regions of sensorimotor viability and possibilities for action arise depending on the agent and its environment.

Transition in the perception of passability

First, we explore the case of an agent moving in a finite room with an aperture. In this case, we assume that the number of steps n is large enough to explore the whole room ($n \rightarrow \infty$) and that the positions of the agent are restricted to all the positions available within the room. In this way, we compute the value of empowerment $\mathcal{E}(n \rightarrow \infty)$ assuming the agent can reach all positions in the room. We consider that the agent is on a square room with sides of size 8, and that the side in front has an aperture in the middle of width W .

We compute the empowerment of 5 different agents with width S equal to 0.5, 0.75, 1, 1.25 and 1.5, for aperture widths in the range $[S/4, 4S]$. Empowerment is obtained from the sensory inputs the agent can perceive from all positions of the room. The result is displayed in Figure 2 (bottom).

As we can observe, the value of empowerment \mathcal{E} presents a transition at different values of W (Figure 2, bottom left). However, if we normalize the values of W by the width of

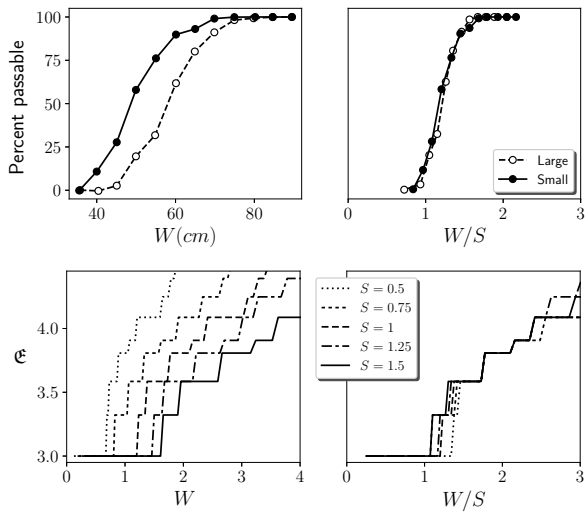


Figure 2: Top: Mean percentage of ‘passable’ judgments in experimental tests with humans in static viewing condition as a function of aperture width (left) and aperture width normalized for shoulder width (W/S , right). Adapted from (Warren and Whang, 1987). Bottom: value of empowerment $\mathcal{E}(n)$ of an agent in a room at enough steps n to explore the whole room ($n \rightarrow \infty$) as a function of aperture width (left) and aperture width normalized for shoulder width (W/S , right). Note that in both cases we can observe an scale-invariant transition for values of W/S slightly larger than 1.

the agent (Figure 2, bottom right) all agents present a transition at the same point (W/S around 1.2), independently of their size².

Affordance spaces as sensorimotor regions of viability

In the previous section, we have measured the empowerment of the agent assuming it can perform an infinite number of movements to explore the whole environment. However, if we restrict the number of available movements, we can compute the value of empowerment for specific positions of the environment in a finite number of steps n , which will depend on perceived sensor states in the locations that are accessible from each initial position. This is useful to characterize the affordances that appear in particular regions of the space that an agent can navigate. For an agent of width $S = 1$ with $n = 15$ steps in a 15×60 room with two apertures of widths 2 and 0.5 separated by 20, we observe the distribution of empowerment values shown in Figure 3 (top). High values of empowerment are obtained only around the aperture that is passable by the agent. We can interpret these regions as

²Note that because of the discrete nature of the model, values of empowerment grows step-wise, and that by transition we refer only to the point where empowerment starts to grow from a flat state).

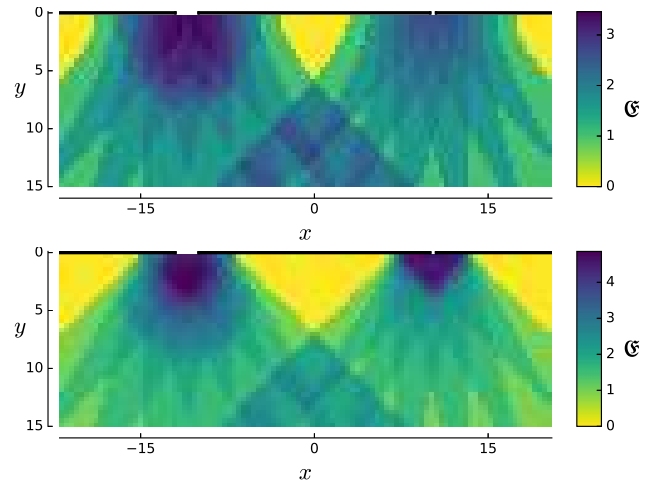


Figure 3: Top: Empowerment of an agent with width $S = 1$ starting at different positions (x, y) for 15 steps in front of a wall with two apertures of different sizes, the left one wider than the agent ($W_L = 2$), and the right one smaller ($W_R = 0.5$). Empowerment around the wide aperture is much larger than around the small one, since they are located at opposite positions of the transition represented in Figure 2, above and below the transition point $W/S = 1$ respectively. Bottom: Same model as in the top figure but with a flat agent ($H = 0$), showing the lack of a clear difference in the empowerment between the wide ‘passable’ aperture and the small ‘not passable’ one. In this case we cannot find a difference between the small and large aperture, showing that the height is crucial for the agent to get information about its width.

spaces of viability in which sensorimotor possibilities exist.

To explore what is the effect of the height of the agent, we compare those results with the empowerment of a ‘flat’ agent (an agent with $H = 0$). Note that in such agent, objects are perceived to be at a different distance (since the agent perceives objects to be at a distance $\sqrt{d^2 + H^2}$), and thus the agent’s sensor will lose any information about the size of the agent. As we can see in Figure 3 bottom, in this case the viability space is ill-defined, and thus the agent is in a state of maladaptation (to the behaviour of traversing an aperture). In this case, both the large and the small aperture show a similar value of empowerment. In contrast, the area of viability in the model with height larger than zero is correctly defined, with large empowerment correlating with the possibility for exploiting the affordance of passing through the aperture.

If we try to extend our minimal model to more complex environments, the relationship between empowerment and possibilities for action is not as easy as in the previous cases. If we analyze a wall with two nearby apertures, we can see that the empowerment measure increases when the sensors

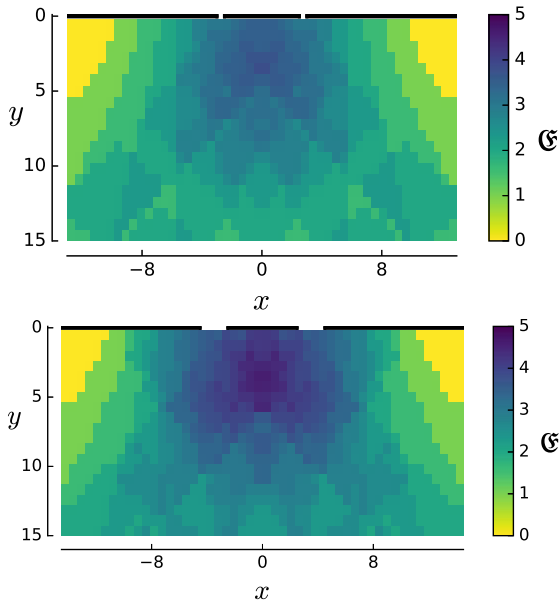


Figure 4: Empowerment of a model with the same parameters as in Figure 3 (top) but with two nearby apertures, small and ‘not passable’ in the top ($W = 0.5$), and larger and ‘passable’ in the bottom ($W = 2$). The empowerment measure shows some limitations when considering more complex environments, yielding high values even in the absence of affordances (top) or mixing the contributions of two affordances (bottom).

have access to both apertures. This increment in empowerment, however, does not always correlate with possibilities of action and affordances. This limitation is illustrated in Figure 4. We can see in the top of this Figure 4 that empowerment is large in the center and relatively close to the apertures, but in this environment there is no affordance available for this agent. In contrast, in the bottom panel of Figure 4 both apertures are wider than the agent and thus they generate two affordances, passing through the left or right aperture. In this cases, thus, it is not clear how to interpret the increase of empowerment, which might be because of a richer environment (even if there is no affordance available) or due to the fact of having two affordances available.

Discussion

In the previous sections we have studied the challenge of interpreting affordances using information theory. We have explored this issue by proposing measures of empowerment as a natural form of quantifying affordances in information theory terms. We have applied this idea to a minimal model reproducing some aspects of Warren’s experiment of perception of passability through apertures. As Figure 2 shows, we successfully reproduced the scale-invariant transition in Warren’s experiment with a measure of the empowerment

of the agent. The results are similar since both transitions are body-scaled. This shows that we can use empowerment measures to quantify information about a particular interpretation of the affordance of passing through apertures in information-theoretical terms. Nevertheless, as we claimed before, this interpretation of affordances in information theoretic terms is a simplified simulation of a full-blown scientific, ecological explanation of how we perceive affordances, so it lacks many key aspects from an in vivo scientific explanation, like a realistic model of visual perception.

In addition, we have shown how empowerment (Figure 3) can describe a field of behavioural possibilities, characterizing the area in which a specific affordance exists in information theoretic terms. We could connect this with more general ideas about the maintenance of regions of viability in adaptive systems. In this work we have not considered the behaviour of the neural system of the agent, but we could think of the family of neural systems that are adapted to a particular affordance as the ones that maintain the state of the agent within a region in which the affordance exists.

Still, the results here present some limitations that should be addressed in further work. First, the empowerment measure is a good description of possibilities for action that generate observable results for the agent. Nevertheless, this information theoretic interpretation of affordances is less restrictive than the original one, since not all possibilities for action are affordances. As we observe in Figure 2 (bottom right), even in the situation when the affordance doesn’t exist ($W/S < 1$) there is some basal level of empowerment (because the movement of the agent can still cause sensations of a small aperture). However, when the affordance appears, it generates a transition in the empowerment metric. Our results suggest that affordances could be quantified by identifying these transitions in the space of possibilities for action. Nevertheless, further work could explore whether this also happens in more complex models and how empowerment metric could be refined to capture more specific aspects about affordances that are not shared by general possibilities for action.

In this line, we made explicit this limitation by showing that the relation between empowerment and affordances in our minimal model holds only for simple environments where either there is one affordance or none. When we analyzed the case of walls with two nearby apertures, we found that the empowerment combines contributions from both apertures, making the absolute value of empowerment not an indicative measure of the presence of an affordance or even of the possibility for action. Even if any of the apertures is wide enough for the agent to pass (Figure 4 top), the empowerment increases around the apertures, but in this case the region of high empowerment does not correlate with a viability region where sensorimotor possibilities arise. Instead, both viability regions overlap, and we cannot disentangle the contributions of each individual affordance.

In order to overcome this limitation, some adaptations might be necessary to extend its usefulness to more complex situations than the one analyzed here. Nevertheless, there are refined versions of the measure, e.g. context-dependent empowerment (Salge et al., 2014; Klyubin et al., 2008), which might successfully be used to separate different sources of empowerment.

Our description of affordances shows similarities with some works in the field of robotics that try to simplify action spaces (Guttenberg et al., 2017) and with the concept of ‘intrinsic options’ developed in the field of reinforcement learning. Options in general are defined as ‘closed-loop policies for taking action over a period of time’ (Sutton et al., 1999), but this definition can be restricted to policies with a termination condition that meaningfully affect the world, i. e., intrinsic options (Gregor et al., 2016). Therefore, the main goal of learning intrinsic options is not to predict future observations, but to control the environment. Information theory and empowerment measures have also been applied in this context (Gregor et al., 2016), and theoretical work in this field might be useful to extend the validity of our interpretation and overcome some of the difficulties mentioned above.

When computing empowerment there is an important distinction between open and closed-loop measures. The open-loop measure assumes that the action sequence is selected in advance, with each action depending only on the initial state. However, this is not very realistic for noisy or changing environments. When computing closed-loop empowerment later actions can change depending on the current sensor state, allowing the agent to adapt to modifications in the environment during the action sequence. It is important to note that in the model we analyzed the agent moves in a deterministic world, meaning that each action is associated with just one specific outcome. Therefore, the sequence of actions that the agent follows is known in advance, and only depends on the initial state, making both open-loop and closed-loop empowerment measures equivalent in our model.

Another aspect for future exploration is that affordances are described as specific possibilities for action which the agent can perceive *directly*. In our work we have not explored how an agent can perceive information about its empowerment. Future work could explore how information about the presence of an affordance flows through variables of the neural system and the body of the agent in a similar way as the experiments performed by Beer and Williams (2015). Furthermore, as empowerment is defined in terms of entropy (of a channel capacity between actuators and sensors), future explorations could quantify to what extent the information captured by empowerment metric also flows through different variables of the agent. Such an analysis could rigorously determine whether information processing in the neural system of the agent is involved in the perception of a particular affordance, or whether this information

is directly available through sensorimotor interaction without internal computation.

Acknowledgements

The authors would like to thank Ezequiel Di Paolo for constructive criticism of the manuscript. Miguel Aguilera was supported by the UPV/EHU post-doctoral training program ESPDOC17/17 and partially by project TIN2016-80347-R. Iñigo Arandia-Romero is thankful to Manuel G. Bedia and was supported by funding from the Government of Aragon (Spain) to acknowledge research groups (ISAAC lab, cod T33_17D). Miguel Aguilera and Iñigo Arandia-Romero were partially supported by project TIN2016-80347-R, funded by the Spanish Ministry of Economy and Competitiveness. Manuel Heras-Escribano developed this work thanks to a 2018 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation (The Foundation accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors), the Project FFI2016-80088-P funded by the Spanish Ministry of Science, and the FiloLab Group of Excellence funded by the University of Granada, Spain.

References

- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in complex systems*, 11(01):17–41.
- Beer, R. D. and Williams, P. L. (2015). Information Processing and Dynamics in Minimally Cognitive Agents. *Cognitive Science*, 39(1):1–38.
- Chemero, A. (2009). *Radical embodied cognitive science*. MIT press.
- Fajen, B. R., Riley, M. A., and Turvey, M. T. (2009). Information, affordances, and the control of action in sport. *International Journal of sport psychology*, page 29.
- Gibson, J. J. (2014). *The Ecological Approach to Visual Perception : Classic Edition*. Psychology Press.
- Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational Intrinsic Control. *arXiv:1611.07507 [cs]*. arXiv: 1611.07507.
- Guttenberg, N., Biehl, M., and Kanai, R. (2017). Learning body-affordances to simplify action spaces. *arXiv:1708.04391 [cs]*. arXiv: 1708.04391.
- Heras-Escribano, M. (2019). *The Philosophy of Affordances*. Palgrave Macmillan, Cham.
- Heras-Escribano, M. and De Pinedo-Garca, M. (2018). Affordances and Landscapes: Overcoming the NatureCulture Dichotomy through Niche Construction Theory. *Frontiers in Psychology*, 8.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: a universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 128–135 Vol.1.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2008). Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems. *PLOS ONE*, 3(12):e4018.

- Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge, U.K. ; New York, 2nd edition edition.
- Reed, E. S. (1996). *Encountering the world: Toward an ecological psychology*. Oxford University Press.
- Salge, C., Glackin, C., and Polani, D. (2014). Empowerment: An Introduction. In Prokopenko, M., editor, *Guided Self-Organization: Inception, Emergence, Complexity and Computation*, pages 67–114. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Shannon, C. (1948). A Mathematical Theory of Communication - Shannon - 1948 - Bell System Technical Journal - Wiley Online Library.
- Slocum, A. C., Downey, D. C., and Beer, R. D. (2000). Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention. In *From animals to animats 6: Proceedings of the sixth international conference on simulation of adaptive behavior*, pages 430–439.
- Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211.
- Warren, W. H. (1984). Perceiving affordances: Visual guidance of stair climbing. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5):683–703.
- Warren, W. H. and Whang, S. (1987). Visual guidance of walking through apertures: body-scaled information for affordances. *Journal of experimental psychology: human perception and performance*, 13(3):371.

The dark room problem in predictive processing and active inference, a legacy of cognitivism?

Manuel Baltieri and Christopher L. Buckley

Evolutionary and Adaptive Systems Group – Sussex Neuroscience, Department of Informatics,
University of Sussex, Brighton, UK
m.baltieri@sussex.ac.uk

Abstract

The free energy principle describes cognitive functions such as perception, action, learning and attention in terms of surprisal minimisation. Under simplifying assumptions, agents are depicted as systems minimising a weighted sum of prediction errors encoding the mismatch between incoming sensations and an agent’s predictions about such sensations. The “dark room” is defined as a state that an agent would occupy should it only look to minimise this sum of prediction errors. This (paradoxical) state emerges as the contrast between the attempts to describe the richness of human and animal behaviour in terms of surprisal minimisation and the trivial solution of a dark room, where the complete lack of sensory stimuli would provide the easiest way to minimise prediction errors, i.e., to be in a perfectly predictable state of darkness with no incoming stimuli. Using a process theory derived from the free energy principle, active inference, we investigate with an agent-based model the meaning of the dark room problem and discuss some of its implications for natural and artificial systems. In this set up, we propose that the presence of this paradox is primarily due to the long-standing belief that agents should encode accurate world models, typical of traditional (computational) theories of cognition.

Introduction

The free energy principle (FEP) and predictive processing (PP) are popular frameworks in the cognitive sciences that advocate the use of probabilistic generative models to describe brain processes including perception, action and higher order cognitive functions (Dayan et al., 1995; Rao and Ballard, 1999; Knill and Pouget, 2004; Friston et al., 2006; Hohwy, 2013; Clark, 2015b; Bogacz, 2017; Buckley et al., 2017). In these frameworks, perception is described as a process of inferring the most likely hidden properties of sensory data by minimising the error between actual sensations and those predicted by a probabilistic generative model (Dayan et al., 1995; Rao and Ballard, 1999; Knill and Pouget, 2004). Active inference, a process theory derived from the free energy principle, introduces also a formal description of action as a way for agents to change their sensory input to better fit their predictions. Agents thus actively interact with the environment to produce sensations that generative models can predict. On this view, behaviour

is generated through interactions with the world defined in terms that are consistent with the perceptual accounts of FEP/PP. Motor commands are expressed as predictions instantiated by the same generative model at a proprioceptive level compared with actual proprioceptive input (Friston, 2011; Adams et al., 2013). These two processes, inferring properties of the world and inferring actions needed to meet expectations, close the sensorimotor loop and suggest a deep symmetry between action and perception.

The “dark room problem” (Friston et al., 2012) is presented in the context of an agent whose only goal is to reduce prediction error. Such agent, it is argued, should find the simplest and most predictable state where prediction error can be minimised, i.e., a dark room with no sensory input. This state, however, fails to account for the complexity of the behaviour that the FEP and PP frameworks claim to account for. Here we propose that this paradox arises mostly from the use of “perception centric” views of PP and active inference theories, with agents seen as simply building generative models of their sensory observations capturing the complexity of the environment. This perception centric view can be seen in analogy to, we claim, traditional sense-model-plan-act architectures (as described by Brooks (1991)), emphasising the role played by detailed and precise world models.

In this work we introduce a minimal model of perception centric agents, showing a simple implementation of agents seeking (and finding) “dark rooms”. We will argue that, from the Bayesian perspective proposed in active inference, this is due to the lack of priors that can affect the behaviour of our agents (cf. Baltieri and Buckley (2017, 2019)), with actions entirely driven by external stimuli.

Perception centric PP and the dark room problem

In perception centric approaches to PP, agents can be described as “perception machines” whose job is to capture, encode and possibly predict the richness of their environment, becoming mirrors of their milieu (Huang and Rao, 2011; Spratling, 2016). This creates, we claim, a GOFAI-like reasoning system that allows an agent to simulate so-

phisticated cognitive tasks using an internal (generative) model that, essentially, mirrors the world (see for instance Ha and Schmidhuber (2018) for a recent example of these generative models in machine learning). The only true novelty introduced by PP interpretations is the explicit use of top-down information flows, inspired by predictive coding accounts of cortical activity (Rao and Ballard, 1999). On this view, PP is depicted as a scheme for the construction of accurate and meticulous world models that serve higher purposes such as planning, attention and decision-making. Action is vicariously implemented based on powerful and accurate models of an agent’s milieu that can be seen as almost detached from the world itself (Hohwy, 2013). The external environment is essentially only “used” during the initial construction of internal models, implicitly assuming that it is possible to encode all of the properties needed for planning and that such properties will not change over time. If the goal of an agent is to to minimise the surprisal or, under certain assumptions, a weighted sum of prediction errors of its sensations (Buckley et al., 2017), what is the role of action in implementations of PP and active inference? An agent that builds models of the world by inferring properties that objectively reflect its incoming sensations should, if prediction error minimisation is its only purpose, also only act to minimise such prediction errors.

In this light, the “dark room problem” (Friston et al., 2012; Sims, 2017; Klein, 2018) describes the case where the best way for an active inference agent to minimise its sensory surprisal is to simply act in order to generate a trivial and easily predictable sensory input stream, cf. the oriental Nirvana analogy (Mumford, 1992). In this thought experiment, an agent can access a so-called “dark room”, a place or a state in the world with no sensory stimuli. It is thus argued that an active inference agent simply looking to minimise its sensory surprisal is bound to go to such room, formulate trivial hypotheses on the lack of sensory input and never move again, indefinitely. Staying in a dark room becomes the best outcome for this agent since the lack of sensations is explained away by trivial predictions, giving thus a prediction error which is constantly zero. This example represents a valuable theoretical construct for the discussion of active inference agents in the context of sensorimotor loops, but as already suggested in Friston et al. (2012) it can never be the case for biological systems. Appealing to classical ideas of homeostasis tracing back to, at least, the good regulator theorem (Conant and Ashby, 1970), only agents whose purpose is to exist while having no realistic physiological constraints could find themselves preferring a dark-room-like situation. The living creatures we know of, on the other hand, show different needs that must be satisfied over time, including for example the maintenance of a certain body temperature and several other variables within boundaries (e.g. glucose, calcium and oxygen levels). The variables ensuring an agent’s survival are proposed to be encoded within an agent through

evolution, and used in a set of homeostatic mechanisms that regulate different processes (cf. the “essential variables” in Ashby (1957)) of a system (Seth, 2014). In active inference, these different drives are represented by priors and are crucial for the role they play in top-down predictions of the world. When these predictions are not matched by the sensory input, errors at the sensory level are generated and propagated in a bottom-up fashion to trigger processes of prediction update and action selection. The balance of top-down and bottom-up flows is modulated by precisions (inverse (co)variances), a set of weights for prediction errors that modulate their strength.

To discuss the role of both priors and precisions in the context of sensorimotor loops, in this work we present some initial results from computational simulations of active inference agents performing basic homeostatic control. By focusing on a minimal model of a “Bayesian cruise controller”, similar in spirit to the “Bayesian thermostat” example found in Buckley et al. (2017), we emphasise the role of perception and action in perception centric active inference agents leading to the dark room puzzle.

A Bayesian cruise controller

In this model, a block of mass = 1 kg (our agent) is placed on a surface with some sliding friction. The goal of this agent is to regulate its velocity, which can be perceived through a sensor, towards a desired set-point v_{des} ($v_{des} = 10$ km/h unless otherwise stated). The regulation will be described as a Bayesian inference process, inspired by the free energy principle and implemented in an active inference set up. The details behind the mechanism for velocity regulation will not be specified, since they don’t add any more insight to our proof of concept. We will simply assume that this agent can apply a force that moves the block against the effects of friction which tend to bring the velocity of the block down to zero. The *generative process*, describing the dynamics of the world for our agent, will simply entail the definition of a velocity variable x (here to be interpreted as hidden state rather than as a position/displacement) that exponentially decays over time with a constant rate α due to the effects of friction. We also describe these dynamics as noisy, with a random variable $w \sim \mathcal{N}(0, \sigma_w^2)$, and have an action variable a that represents the force applied by the agent as an input (in states-space formulations terms) to achieve homeostatic control. The generative process is presented in the form of a state-space model as in most implementations of active inference, e.g., Friston (2008); Buckley et al. (2017); Bogacz (2017); Baltieri and Buckley (2019):

$$x' = -\alpha x + a + w \quad (1)$$

To simplify the example, no other exogenous inputs (in a state-space representation sense) are added, cf. Baltieri and Buckley (2019) where we also considered forces such

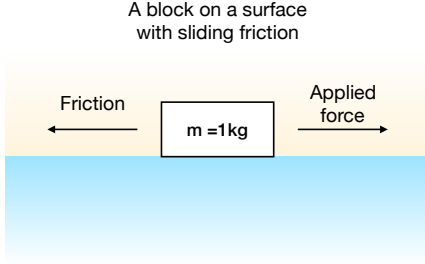


Figure 1: **The agent, a Bayesian cruise controller.** A block of mass = 1 kg, i.e. the agent, is placed on a surface with dynamic friction. The goal of the agent is to reach and maintain a velocity v_{des} .

as wind. To maintain (mathematical) consistency with previous formulations we represent the generative process using a Langevin form where w is weakly autocorrelated in a Stratonovich sense, i.e., not a Wiener process, even if the noise variables are implemented as white noise in our code for simplicity¹, see Friston (2008); Baltieri and Buckley (2019) for discussion. The velocity and accelerations measurements y, y' are given as noisy readings of x, x' with observation noise $z \sim \mathcal{N}(0, \sigma_z^2), z' \sim \mathcal{N}(0, \sigma_{z'}^2)$:

$$y = x + z \quad y' = x' + z' \quad (2)$$

The next step requires the definition of the agent's generative model (Buckley et al., 2017; Bogacz, 2017), including a model of the system's dynamics:

$$x' = -\alpha x + v_{des} + w \quad (3)$$

and of measurements:

$$y = x + z \quad y' = x' + z' \quad (4)$$

One of the major assumptions made in active inference is that the action variable a cannot be observed directly by an agent (i.e., it's not part of its generative model) and not necessary for problems of control (Friston et al., 2010; Friston, 2011), giving rise to a different way of implementing regulation (Baltieri and Buckley, 2018a, 2019). In active inference, one thus assumes that an agent is endowed with a minimal mapping encoding how actions a modify observations y, y' (rather than hidden states x, x') via reflex arcs, as discussed in Friston et al. (2010); Friston (2011). In this case we also use, again for consistency, the notation in generalised coordinates of motion defined in Friston (2008); Buckley et al. (2017) for random variables z, z' . Under Gaussian assumptions for z, z' and w , one can write the above state-space rep-

¹<https://github.com/mbaltieri/BayesianCruiseController>

resentation of the generative model in a probabilistic form:

$$\begin{aligned} p(y|x) &= \mathcal{N}(y, \sigma_z^2) \\ p(y'|x') &= \mathcal{N}(y', \sigma_{z'}^2) \\ p(x'|x, v; \alpha) &= \mathcal{N}(-\alpha x + v_{des}, \sigma_w^2) \end{aligned} \quad (5)$$

and considering the Laplace-encoded variational free energy defined in equation (12) in Baltieri and Buckley (2019), here reported as

$$F \approx -\ln p(\tilde{\psi}, \tilde{x}, \tilde{v}; \theta, \gamma) \Big|_{\tilde{\theta}=\tilde{\mu}} \quad (6)$$

one can see that the probabilistic description of the generative model presented here reflects the likelihood and prior distributions necessary to build the generative density for the definition of the free energy (Buckley et al., 2017). The generative density in equation (6) can be decomposed into

$$p(\tilde{\psi}, \tilde{x}, \tilde{v}; \theta, \gamma) = p(\tilde{\psi}|\tilde{x}, \tilde{v}; \theta, \gamma) p(\tilde{x}, \tilde{v}; \theta, \gamma) \quad (7)$$

and after specifying $\tilde{\psi} = \{y, y'\}, \tilde{x} = \{x, x'\}, \tilde{v} = \{v_{des}\}, \theta = \alpha$ and hyperparameters γ encoding properties about precisions $\pi_z, \pi_{z'}, \pi_w$, one gets

$$p(\tilde{\psi}|\tilde{x}, \tilde{v}; \theta, \gamma) = \{p(y|x), p(y'|x')\} \quad (8)$$

$$p(\tilde{x}, \tilde{v}; \theta, \gamma) = p(x'|x, v; \alpha) \quad (9)$$

The free energy then becomes:

$$\begin{aligned} F(y, \tilde{\mu}_x, \mu_v) &\approx \frac{1}{2} [\pi_z (y - \mu_x)^2 + \pi_{z'} (y' - \mu_{x'})^2 \\ &\quad + \pi_w (\mu_{x'} + \alpha \mu_x - \mu_v)^2 - \ln(\pi_z \pi_{z'} \pi_w)] \end{aligned} \quad (10)$$

with perception $\dot{\tilde{\mu}}_x = D\tilde{\mu}_x - \partial F / \partial \tilde{\mu}_x$, following Friston et al. (2010); Bogacz (2017); Buckley et al. (2017), defined as:

$$\begin{aligned} \dot{\mu}_x &= \mu_{x'} - [-\pi_z (y - \mu_x) + \pi_w \alpha (\mu_{x'} + \alpha \mu_x - \mu_v)] = \\ &= \mu_{x'} + [\pi_z (y - \mu_x) - \pi_w \alpha (\mu_{x'} + \alpha \mu_x - \mu_v)] \\ \dot{\mu}_{x'} &= \mu_{x''} - [\pi_w (\mu_{x'} + \alpha \mu_x - \mu_v)] = \\ &= -\pi_w (\mu_{x'} + \alpha \mu_x - \mu_v) \end{aligned} \quad (11)$$

and action, $\dot{a} = -\partial F / \partial a$ (Friston et al., 2010; Buckley et al., 2017), as:

$$\begin{aligned} \dot{a} &= -[\pi_z (y - \mu_x) \partial y / \partial a + \pi_{z'} (y' - \mu_{x'}) \partial y' / \partial a] \\ &= -[\pi_{z'} (y' - \mu_{x'})] \end{aligned} \quad (12)$$

where we use the fact that an implicit model in terms of reflex arcs (Friston, 2011) is embodied by the agent via

$$\partial y' / \partial a = 1, \quad \partial y / \partial a = 0 \quad (13)$$

These equations, when combined, form an action-perception loop with information inferred from the environment

through perception and control exerted on the world via action. The combination of action and perception is regulated by precision parameters “ π ”, representing weights in the weighted sum of prediction errors, see equation (10). Precisions encode the uncertainty (they are in fact inverse (co)variances) of different variables of a generative model in an agent and effectively regulate the minimisation of variational free energy in equation (11) and equation (12). For the remainder of this work we will specify (*weighted*) *sensory prediction errors* as the errors weighted by sensory precisions π_z or more in general $\pi_{\bar{z}}$ and *process* or (*weighted*) *system prediction errors* as the ones weighted by process or system precisions π_w or $\pi_{\bar{w}}$ if dealing with generalised coordinates of motion (Friston, 2008; Buckley et al., 2017; Baltieri and Buckley, 2019). This distinction will be useful when we emphasise the role of precision weights on the minimisation of variational free energy, producing behaviours influenced by their relative strength.

More in general, precision parameters in a generative model can be unrelated to the actual precisions of the true hidden states, causes and observations of a generative process (i.e. the world dynamics), and in some cases this misalignment is claimed to be necessary for behaviour (Feldman and Friston, 2010; Wiese, 2016). Precisions have also been addressed also in terms of “confidences”, thought to encode how confident an agent is about its estimates of hidden variables. Precisions π ’s are in the most general case dynamic parameters that can change over time allowing for several types of behaviours to emerge depending on different situations and needs of an agent, see for example Feldman and Friston (2010). In this work we assume fixed-valued precisions in order to focus on cases of “precision engineering” (Clark, 2015b) showing their role in the emergence of different behaviours as in, for instance, Baltieri and Buckley (2017). More specifically, we focus on “perception centric” (or passive) agents within the context of active inference, agents that heavily rely on perceptual inference, (over)focusing on estimating hidden properties of their sensory input. This perception centric view will be implemented with agents whose sensory prediction errors dominate system prediction errors, emphasising the bottom-up nature of incoming signals, as described in standard models of predictive processing models for perception Huang and Rao (2011); Spratling (2016). We will also consider the importance of a closed sensorimotor loop, initially focusing on agents that can only perceive their environment without acting, and then introducing the ability for agents to affect the world, once again in a perception centric view of PP.

Just observing, the passive tracker

Passive trackers are agents that can only perceive their world without the ability to modify any of its properties. They are an extreme version of the archetypical case advocated by “perception centric” PP (Huang and Rao, 2011; Spratling,

Table 1: **Agents’ parameters and setups.** The table summarises the parameters used to simulate our two agents, the passive tracker and the active tracker, following the implementation of equation (11) and equation (12).

	π_z	$\pi_{z'}$	π_w	Action
Passive tracker	$\exp(1)$	$\exp(1)$	$\exp(-12)$	$a = \dot{a} = 0$
Active tracker	$\exp(1)$	$\exp(1)$	$\exp(-12)$	$\dot{a} = \partial F / \partial a$

2016), already prioritising the estimation of the causes of observed sensations over adaptive behaviour. Passive trackers over-prioritise perception over action and in fact are implemented following equation (11) for perception, while actions a in equation (12) are not included, i.e., $a = \dot{a} = 0$. They also heavily rely on bottom-up observations over top-down priors, with weighted sensory prediction errors taking a dominant role and driving predictions about incoming data. The larger the ratio between sensory and system prediction errors, the smaller is the role played by prior beliefs. As we can see in Fig. 2, in the simplest case, suitable (although small) priors filter out some of the measurement noise, separating the signal to be inferred (the black line) from the noise due to sensors/receptors. Without action, this agent cannot control its velocity and reach the target velocity ($v_{des} = 10$ km/h), naturally slowing down and eventually stopping following its autonomous dynamics.

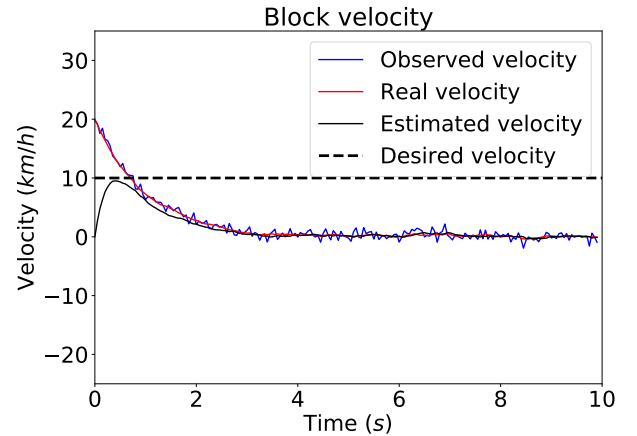


Figure 2: **(The passive tracker) The velocity of the block.** The velocity perceived by the agent (blue line), its best estimate according to weak priors (red) and the block’s true velocity, i.e. without measurement noise (black).

In Fig. 3 we can see that the variational free energy of our agent is (on average) minimised over time (Fig. 3c), driven mainly by the weighted prediction errors on sensory input. Weighted sensory prediction errors vary in the order of 10^1

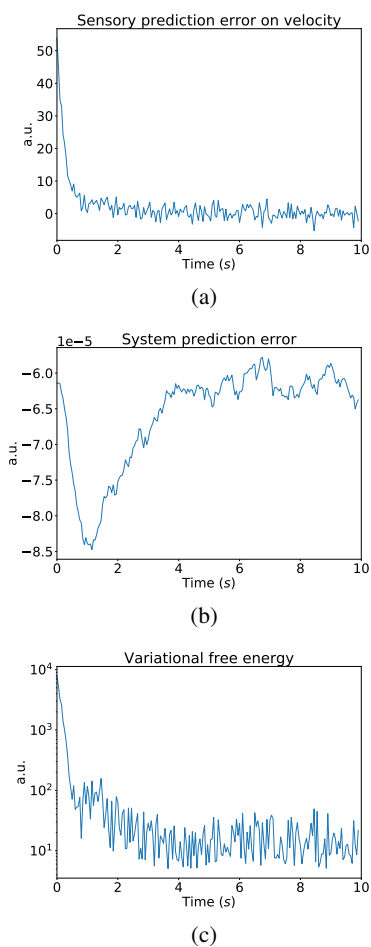


Figure 3: **(The passive tracker) Weighted prediction errors and variational free energy.** The evolution of (A) sensory prediction errors on velocity, (B) system prediction error and (C) variational free energy.

(Fig. 3a and similarly for the acceleration, not reported), while the system error is in the order of 10^{-5} (Fig. 3b).

Acting with no reason, the active tracker

Active trackers are agents that can actively interact with their environment and unlike their passive version, they integrate action via equation (12) to close the sensorimotor loop together with perception, implemented by equation (11). However they are just another (although more elaborate) example of the perception centric description introduced by Clark (2015a,b), a direct consequence of Bayesian brain/predictive coding schemes (Rao and Ballard, 1999; Huang and Rao, 2011; Spratling, 2016) endowed with simple mechanisms for active behaviour and motor control. These agents can impact their environment through motor actions but they only do so to better sample sensations in agreement with their existing predictions, producing a “kind

of self-fulfilling prophecy” (Hohwy, 2013; Clark, 2015a) entirely driven by incoming sensory input. Active trackers don’t use (possibly relevant) priors to estimate their sensations and, as in the case of the passive tracker, are completely enslaved by their observations in a state of pure information gathering. Actions are only produced to cancel sensory prediction errors, to generate more accurate predictions about the world. Effectively, this creates the “dark room problem” for *active* agents exposed in Friston et al. (2012), i.e., agents that “predict”, or rather account for, all their observations, with action simply bound to produce a process of inconclusive behaviour (unless the purpose for a system is to just estimate the hidden properties of its observations, unlike ours!).

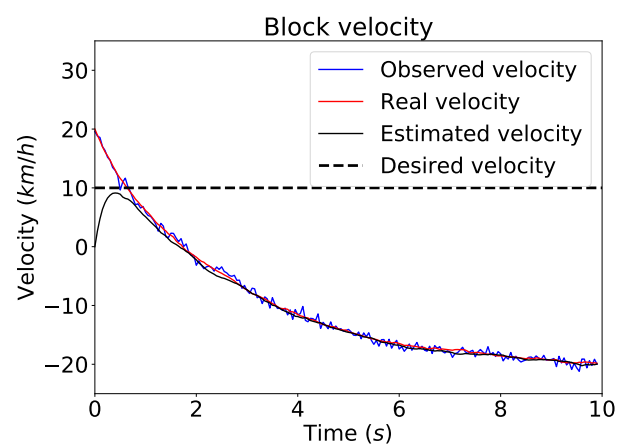


Figure 4: **(The active tracker) The velocity of the block.** The velocity perceived by the agent (blue line), its best estimate according to weak priors (red) and the block’s true velocity, i.e. without measurement noise (black).

The estimate of velocity, Fig. 4, becomes a good description of the real variable in the world as in the case of the passive tracker. In the passive tracker example, however, the block naturally slowed down and eventually stopped (nearly stopped, because of the presence of environmental noise) close to the origin. In the active version of the tracker, the initial sensory prediction error given by the estimate μ_x initialised at 0 triggers an action (see Fig. 5) which will then be constant over time after the prediction error on velocity is minimised, i.e. when the agent can predict its velocity. Having no other drive but to accurately predict its observations, this agent maintains its motor action constant since it has no associated cost. Random initialisations of μ_x give different set-point equilibria to the system, providing different, but still accurate, estimates of the block’s motion after actions bring it into a predictable state more quickly. Similarly to the passive tracker, the agent cannot control its behaviour towards the target velocity, but due to the presence of actions a affecting the environment, it now follows the non-

autonomous dynamics driven by its own actions, generating observations more easily predictable from its perspective.

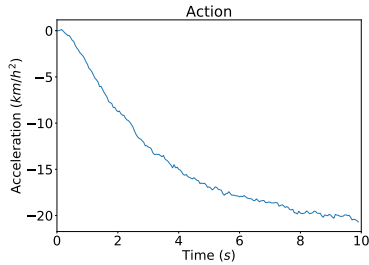


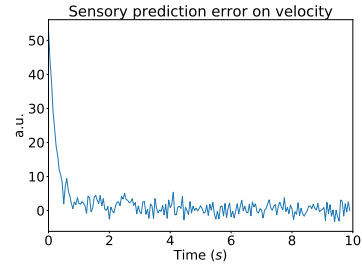
Figure 5: **(The active tracker) The motor action of the agent.** The action induced by the minimisation of variational free energy following active inference given, in this case, a weak prior.

As in the case of the passive tracker, (weighted) sensory prediction errors (Fig. 6a for velocity and the one on acceleration, not reported) exert a much larger influence on the minimisation of variational free energy (Fig. 6c) due to the precision weighting mechanism enforcing their role. The only significant difference between the active and the passive versions is on the process prediction error, cf. Fig. 6b and Fig. 3b, given by the fact that the active tracker gets further away from the “desired” state represented by the prior thanks to its motor actions, while still fulfilling its only goal of better predicting its incoming sensations.

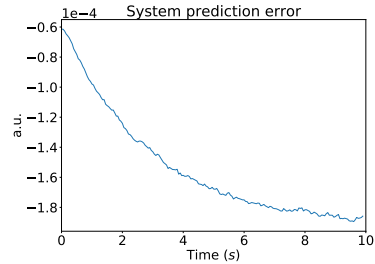
Discussion

In theories derived from the Bayesian brain hypothesis (Knill and Pouget, 2004) and predictive processing (Hohwy, 2013; Clark, 2015b), there is often a strong emphasis on perceptual processes. This is both due to historical reasons that trace these ideas back to work by Helmholtz and related theories of analysis by synthesis (Von Helmholtz, 1867; Neisser, 1967; Gregory, 1970), and to a strong tradition in the cognitive sciences to focus on perception and cognition over action and behaviour (Fodor, 1983; Boden, 2006). The repercussions of this bias in Bayesian theories of the mind are deep and rooted, constantly re-emerging even in the most modern proposals on the Bayesian brain. Following the definition given by Clark (2015a,b), we strongly advocate for a formal distinction between “perception centric” and “action-oriented” Bayesian approaches to cognitive science (see also Engel et al. (2016)), with implications potentially capturing aspects of the more general discussion between traditional and 4E approaches to cognitive science.

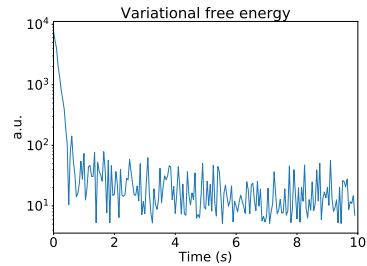
In this work we provided a minimal model of a sensorimotor loop built using active inference and aimed at showing, with an example of homeostatic regulation, some of the possible misunderstandings of the FEP and related theories. Here we focused on an initial account of the “dark



(a)



(b)



(c)

Figure 6: **(The active tracker) Weighted prediction errors and variational free energy.** The evolution of (A) sensory prediction errors on velocity, (B) system prediction error and (C) variational free energy.

room problem” proposed in Friston et al. (2012) following arguments introduced in Mumford (1992). This problem describes the contrast between the rich repertoire of behaviours of real living creatures and the simple mandate of an agent looking to only minimise the surprisal of its sensations as advocated by PP and FEP. In particular, agents minimising their surprisal should, it is claimed, find an easily predictable state and cease to receive any new input, minimising their prediction errors while avoiding new sensations, i.e., a dark room. It was our goal to make the example especially simple, and for this reason the problem of regulation was reduced to a (Bayesian) cruise controller for an agent (i.e. a block) sliding on a surface with dynamic friction. The friction naturally slows the block down, but the agent is endowed with the ability to apply a force over time that allows the block to move and maintain a desired speed. With this example we then explored two cases representing an open

and closed action-perception loop in active inference. The weaknesses of stories without motor actions became soon obvious, but it was nonetheless important to establish the background over which this work is based (see also Bruineberg et al. (2018), where this point is explored in depth).

Alongside the absence or presence of action to define an appropriate sensorimotor loop, we also began investigating the balance of prediction errors. As we can see in equation (10), the expression for the variational free energy under the Laplace approximation is reduced almost entirely to a weighted sum of prediction errors. These errors can be divided into (weighted) sensory and (weighted) process or system prediction errors, the former encoding mismatches between the current best estimates of sensory data and bottom-up (true) sensory data, the latter representing the differences between top-down prior information and the current best estimate of the hidden variables of a system. All these prediction errors are weighted by precision hyperparameters, the inverse (co)variance of observations and hidden dynamics of a system. As stressed in previous work (Baltieri and Buckley, 2017, 2019), these hyperparameters need not encode true properties of the world and can instead be seen as quantifying the uncertainty, or confidence, of an agent’s estimates. Considering that precisions are, in principle, defined over a continuous interval of values, we simplified our initial analysis by imposing high sensory precisions, $\pi_{\bar{z}} = \exp(1)$ and low system precisions, $\pi_{\bar{w}} = \exp(-12)$. Higher precisions drive the minimisation of free energy, enforcing the relative strength of one subset of hyperparameters and relative prediction errors over the other, see equation (11) and (12) (and results in Baltieri and Buckley (2017, 2019)).

We initially studied the passive tracker, representing an extreme version of (almost) purely bottom-up driven perceptual processing. The passive tracker passively engages with new observations, attempting to estimate new observations. The complete lack of prior information however, forces this agent to rely entirely on new observations and so, at best, to track the incoming sensations over time after they have been observed. For this agent, every sensation is essentially “surprising” (in statistical terms) since priors play little to no role in making predictions about incoming data. Sensory prediction errors have a much greater amplitude and are thus driving the minimisation of variational free energy. These agents present in a straightforward way some of the arguments advocated by ideas of analysis by synthesis and the Bayesian brain hypothesis (Knill and Pouget, 2004; Yuille and Kersten, 2006), in particular the necessity of top-down information in the form of priors to disambiguate observations, whose estimates are otherwise entirely enslaved by bottom-up signals. In our example, while top-down information is available to the agent, it is completely overshadowed by the presence of large weighted sensory prediction errors that drive the minimisation of variational free energy.

In this set up, homeostatic regulation requires both a per-

ceptual process of estimation of the world (i.e. the agent’s velocity) and an action selection procedure that allows, at least in principle, an agent to fulfil its “desires”, i.e. targets encoded in the form of a prior. The agent we investigated however, the active tracker, follows the same fate of the passive one, bound to simply attempt to account for its observations. In the active tracker, action simply enacts behaviour that generates more predictable sensory input, in analogy to the dark room problem (Friston et al., 2012). An agent with no strong priors and whose only purpose is thus to predict its sensations should look for a state where sensations are trivially predicted, i.e. a dark room. Considering the block in our set up, the closest state to a “dark room” is any equilibrium of the system reached when action is stationary, since acting is modelled without any associated cost. This agent simply finds the best way to predict its state by bending the world to its predictions and generating predictions that better conform to its sensations.

It has been argued that the presence of strong top-down prior information that *misrepresent* the incoming sensations can generate actions that compensate for sensory prediction errors generated by the misalignment of top-down priors and bottom-up sensations, allowing an agent to fulfil its goals (Wiese, 2016). On this “action oriented” view of PP and active inference (Engel et al., 2016; Clark, 2015b), generative models do not encode veridical information of incoming sensations but on the contrary, describe the desires of an agent with the very purpose of creating mismatch errors that only active behaviour can minimise. The two example agents presented in this work, the “passive tracker” and the “active tracker”, invoke a more traditional notion of generative model as a stand-in for the environment, providing an accurate and objective characterisation of the world an agent traverses. This outlines the connections between “perception centric” descriptions of PP (Huang and Rao, 2011; Hohwy, 2013; Spratling, 2016) and traditional, computational accounts of the mind (Newell et al., 1972; Fodor, 1983) where the necessity of accurate world models is a central tenet of cognitive processes. On the other hand, the presence of strong priors may denote a more “action oriented” perspective of PP and active inference, one where precise models of the world are not only unnecessary but fundamentally detrimental (Clark, 2015a; Wiese, 2016), as seen in our simulations where the agent never reached the desired speed. Agents emphasising the role of priors can (potentially) better represent the need for ideas inspired by 4E (embodied, enactive, embedded and extended) theories in PP, while still advocating for generative models of approximate understandings of the world (Baltieri and Buckley, 2018b) and sensorimotor contingencies and coupled agent-environment systems (Baltieri and Buckley, 2017). The in-depth exploration of an action oriented version of our Bayesian cruise controller with a more central role for priors implemented using different precision weights is, however, left for future work.

Acknowledgements

This work was supported in part by a BBSRC Grant BB/P022197/1.

References

- Adams, R. A., Shipp, S., and Friston, K. J. (2013). Predictions not commands: active inference in the motor system. *Brain Structure and Function*, 218(3):611–643.
- Ashby, W. R. (1957). *An introduction to cybernetics*. Chapman & Hall Ltd.
- Baltieri, M. and Buckley, C. L. (2017). An active inference implementation of phototaxis. In *14th European Conference on Artificial Life 2017, Lyon, France*, pages 36–43.
- Baltieri, M. and Buckley, C. L. (2018a). The modularity of action and perception revisited using control theory and active inference. In Ikegami, T., Virgo, N., Witkowski, O., Oka, M., Suzuki, R., and Iizuka, H., editors, *The 2018 Conference on Artificial Life*, pages 121–128.
- Baltieri, M. and Buckley, C. L. (2018b). A probabilistic interpretation of pid controllers using active inference. In Manoonpong, P., Larsen, J. C., Xiong, X., Hallam, J., and Triesch, J., editors, *From Animals to Animats 15*, pages 15–26. Springer International Publishing.
- Baltieri, M. and Buckley, C. L. (2019). PID control as a process of active inference with linear generative models. *Entropy*, 21(3).
- Boden, M. A. (2006). *Mind as machine: A history of cognitive science*. Clarendon Press.
- Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211.
- Brooks, R. A. (1991). New approaches to robotics. *Science*, 253(5025):1227–1232.
- Bruineberg, J., Kiverstein, J., and Rietveld, E. (2018). The anticipating brain is not a scientist: the free-energy principle from an ecological-enactive perspective. *Synthese*, 195(6):2417–2444.
- Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 14:55–79.
- Clark, A. (2015a). Radical predictive processing. *The Southern Journal of Philosophy*, 53(S1):3–27.
- Clark, A. (2015b). *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press.
- Conant, R. C. and Ashby, W. R. (1970). Every good regulator of a system must be a model of that system. *International journal of systems science*, 1(2):89–97.
- Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The Helmholtz Machine. *Neural computation*, 7(5):889–904.
- Engel, A. K., Friston, K. J., and Kragic, D. (2016). The pragmatic turn: Toward action-oriented views in cognitive science.
- Feldman, H. and Friston, K. J. (2010). Attention, uncertainty, and free-energy. *Frontiers in human neuroscience*, 4:215.
- Fodor, J. A. (1983). *The Modularity of Mind*. MIT Press.
- Friston, K. J. (2008). Hierarchical models in the brain. *PLoS Computational Biology*, 4(11).
- Friston, K. J. (2011). What is optimal about motor control? *Neuron*, 72(3):488–498.
- Friston, K. J., Daunizeau, J., Kilner, J., and Kiebel, S. J. (2010). Action and behavior: A free-energy formulation. *Biological Cybernetics*, 102(3):227–260.
- Friston, K. J., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1):70–87.
- Friston, K. J., Thornton, C., and Clark, A. (2012). Free-energy minimization and the dark-room problem. *Frontiers in psychology*, 3:130.
- Gregory, R. L. (1970). *The intelligent eye*. ERIC.
- Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.
- Hohwy, J. (2013). *The predictive mind*. OUP Oxford.
- Huang, Y. and Rao, R. P. (2011). Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5):580–593.
- Klein, C. (2018). What do predictive coders want? *Synthese*, 195(6):2541–2557.
- Knill, D. C. and Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712–719.
- Mumford, D. (1992). On the computational architecture of the neo-cortex. *Biological cybernetics*, 66(3):241–251.
- Neisser, U. (1967). *Cognitive psychology*. Appleton-Century-Crofts.
- Newell, A., Simon, H. A., et al. (1972). *Human problem solving*, volume 104. Prentice-Hall Englewood Cliffs, NJ.
- Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.
- Seth, A. K. (2014). The Cybernetic Bayesian Brain. In *Open MIND*. Open MIND. Frankfurt am Main: MIND Group.
- Sims, A. (2017). The problems with prediction. In Metzinger, T. K. and Wiese, W., editors, *Philosophy and Predictive Processing*, chapter 23. MIND Group, Frankfurt am Main.
- Spratling, M. (2016). Predictive coding as a model of cognition. *Cognitive processing*, pages 1–27.
- Von Helmholtz, H. (1867). *Handbuch der physiologischen Optik*, volume 9. Voss.
- Wiese, W. (2016). Action is enabled by systematic misrepresentations. *Erkenntnis*, pages 1–20.
- Yuille, A. and Kersten, D. (2006). Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308.

NukaBot: Research and Design of a Human-Microbe Interaction Model

Dominique Chen¹, Hiraku Ogura and Young Ah Seong²

¹Waseda University, Tokyo, Japan

²The University of Tokyo, Tokyo, Japan
dominique@waseda.jp

Abstract

We describe the design, implementation and on going evaluation processes of “NukaBot”, a system built to realize Human to Microbe Interaction, aimed to assist production of fermented food. Our system senses, records and analyzes in real time the fermentation process inside a “*nukadoko*”, a traditional method to produce vegetable pickles in Japan that involves a highly complex network of microbes. The flora of a *nukadoko* mainly consists of lactic acid bacteria, yeasts and gram-negative bacteria. The novelty of our system lies in providing an intuitive user interface that lets its user chat with a virtual persona attributed to the *nukadoko*. The NukaBot thus enables non-specialists to discern the complex dynamics of the microbial communities within a *nukadoko* in daily situations.

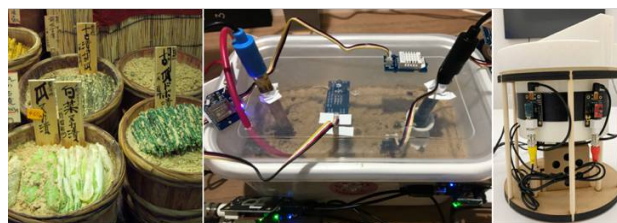


Figure 1: Traditional *nukadoko* (left), *nukadoko* with sensors (center), NukaBot system (right)

Fermentative Bacteria and Human

Symbiosis with a Dynamic Microbiome

Nukadoko is a popular traditional form of fermented food production in Japan. It consists of a bed of rice bran, mixed with salt and water. Billions of lactic acid bacteria and other microbes are implanted inside a *nukadoko*, through the vegetables placed inside. The lactic acid bacteria metabolize the glucose of the vegetables to produce lactose, which adds a unique sour taste to the resulting pickles.

The microbiome of a *nukadoko* is a highly complex system where various kinds of bacteria interact dynamically. Although lactic acid bacteria such as *Lactobacilli* and *Lactococci* are considered principal contributors to the generation of tasty pickles, it is also known that a sufficiently aged *nukadoko* (from 2 weeks to 100 years) contains a considerable amount of gram-negative bacteria such as *Enterobacter*, and natural yeasts, which are said to create a deeper flavor to the pickles.

The biodiversity of *nukadokos* is an important characteristic when compared with Western pickling culture. Traditional European pickles such as the French *cornichon* are immersed in vinegar and are designed to keep the same taste homeostatically. In contrast, the taste of *nukadoko* pickles changes dynamically depending on its synthetic status, usually on a daily basis. The maintainer of a *nukadoko* needs to stir the rice bran periodically (at least once a day) in order to inhibit the excessive proliferation of aerobic bacteria. Should the aerobic bacteria become dominant, a *nukadoko* can easily rot, resulting in a clearly unpleasant odor.

Designing Human-Microbe Interaction

Design Motivations and Principles

We can define *nukadokos* as risk-sensitive systems, where human intervention not only plays a decisive role to their survival, but also enables them to generate rewards for humans in the form of delicious food. Since taste is a subjective value that varies from person to person, it is impossible to define an universal good taste for *nukadokos*.

Based on these premises, we started to conceptualize the idea of NukaBot, an information and communication system that assists humans in keeping their *nukadokos* in good condition. The system should not be a dictating or an automated machine that would break the relationship between human and *nukadoko*: rather, it should serve as a mediator that contributes to the well-being of both sides, and especially empowers human cognition so that the maintainer gains more awareness of her *nukadoko*'s dynamics.

Hypothetical Aging Model

Before assembling the system, we have reviewed past biological scientific research on fermentation in *nukadoko*, in order to devise a hypothetical model for us to evaluate the aging process of a *nukadoko* by our own.

Based on Imai et al. 1983's analysis on the aging of *nukadoko* and Nakagawa et al. 2001's study on the shift of lactic acid bacteria, we have classified 3 different stages of a *nukadoko*'s growth: at first, the *nukadoko* resembles a salted preservation container, where the fermentative activities are low; secondly comes the pickling stage, when the lactic acid bacteria become dominant and the generated *nukazuke* tastes similar to European pickles; thirdly, the *nukadoko* enters a

stage where both aerobic and anaerobic bacteria become active, and the flavor gains much depth and richness.

Continuous Experiments

To test the hypothesis, we have assembled 3 nukadokos equipped with affordable sensors and electronic circuits to record various data. For the period of October 2018 to February 2019, we have logged the pH (potential of hydrogen), the ORP (oxidation-reduction potential), internal and external temperature, humidity and moisture, electric conductance, and 8 different types of gases (NH₃, CH₄, C₄H₁₀, NO₂, C₂H₅OH, CO, C₃H₈, H₂), for every minute. All sensed data are stored in a cloud relational database, and we crafted a visualization dashboard online (Figure 2).

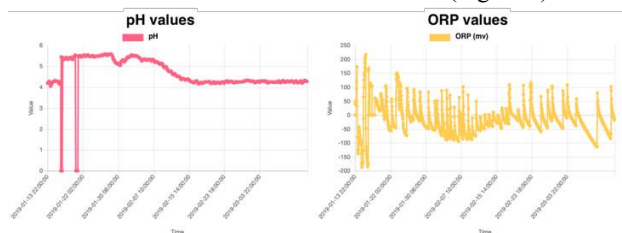


Figure 2: Example of sensor data graph (pH and ORP)

During the course of the experiment, we used the collected data as proxies to microbial activities. pH values indicate increase and decrease of lactic acid bacteria; ORP values reflect activities of aerobic (positive ORP) and anaerobic (negative ORP) bacteria (Higashi et al., 1985). Electric conductance is used to approximate change in salinity. Methanol and ethanol show existence of alcoholic fermentation by yeasts. Propane, butane, ammonia are signs of gram-negative aerobic bacteria. Data were collected both in fermenting and in rotting states.

Data Analysis and System Implementation

We conducted a Principal Component Analysis using 173 data points of 15 sensor values collected during the period of November 2018 to January 2019, with 3 data points per day on average. Figure 3 shows the resulting data obtained from 1st and 2nd PCs. The Proportion of Variance for the 1st and 2nd PCs were 0.366 and 0.189. We found that ORP has a strong influence in the 1st PCs. We conjectured that there are clusters of sensors with similar data patterns throughout the evolution of a nukadoko's status.

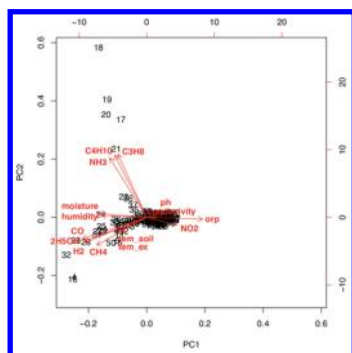


Figure 3: Principle Component Analysis of sensor data

Based on these sensor values, we have implemented a preliminary algorithm to evaluate the health status of a nukadoko at any moment. We connected a voice recognition system with a speaker to the NukaBot, so that it can alert human when it needs assistance and answer simple questions from human: it can verbalize prediction of taste (sourness/bitterness), the current stage of growth (salting, pickling, nukadoko), and whether there is a need to stir the rice bran (Figure 4). Additional explanatory materials (PCA results and videos) can be accessed at the following URL: <http://infinityloops.xyz/nuka/alife/add.html>.

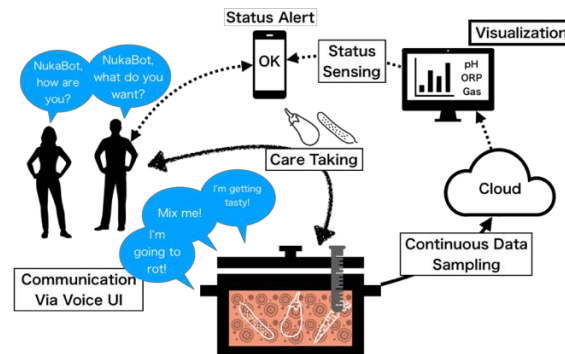


Figure 4: Schematics of the Human-Microbe Interaction

Future Outlook and Discussions

We have so far implemented one complete NukaBot for the occasion of an exhibition at the XXII Triennale di Milano. Currently, we are continuously collecting data from 1 NukaBot located in Milan, in addition to 3 nukadokos with sensors placed in each co-author's respective home in Japan.

We need to reflect sensory assessment labels into our analysis to show correlation between data and actual status of nukadokos. Furthermore, the human interaction with the NukaBot also needs to be evaluated. How does her perception of the innumerable yet invisible microbes change? Does it yield more affection to the nukadoko? How can we design a valid protocol between human and microbes, so to realize a sustainable symbiotic relationship between them? We believe that our development leads us to a broader discussion about design approaches in representing and understanding the coevolution of humans and microbes.

References

Imai, M., Hirano, S. and Aiba, M. (1983). Aging of "Nukadoko": Changes in Microflora and Constituents. *Nippon Nogeikagaku Kaishi*, 57(11):1105-1112.

Higashi, K., Yamamoto, Y. and Yoshii, H. (1985). The Effect of Oxidation-Reduction Growth of the Salt-Tolerant Microorganisms. A Study of Oxidation-Reduction Potential (Part I). *J. Brew. Soc. Japan*, 80(4):270-273.

Nakagawa, H., Mizuno, T., Shimizu, T., Kaneko, J., Kadono, M., Itoh, T., Sakai, S. and Terada A. (2001). Lactic Acid Bacteria Flora Isolated from Salted Vegetables. *Jpn. J. Food Microbiol.* 18(2):61-66.

Ono, H. and Nakayama, J. (2014). Microbiota analysis on fermented foods using next generation sequencer—Digging complex microbiota of fermented foods—. *Japanese Journal of Lactic Acid Bacteria*, 25(1):3-12.

Hybrid Synthetic Approach to Animal Interaction

Hiroyuki Iizuka, Yosuke Nakamoto and Masahito Yamamoto

Graduate School of Information Science and Technology, Hokkaido University, Japan
iizuka@complex.ist.hokudai.ac.jp

Introduction

Not only studying and analyzing real lives on earth but also creating life as it could be on computer simulations can be our strong method to understand what life is (Langton, 1989). It becomes the main philosophy in artificial life community and various types of simulations have been developed in different levels of life such as artificial chemistry, evolutionary robotics, multi-agent social system, etc. The synthetic approaches have been applied not only to computer simulations but also to chemistry and robots in our real world. Regardless of the virtual or real world as experimental grounds, when we can reconstruct life there, we could understand the essential mechanisms that make an entity capable of being life.

Those experiments have been performed in either real or virtual world separately. As rather new direction, in this paper, we investigate a hybrid synthetic approach that combines the real and virtual experiments. In biology, there is a close idea for our hybrid synthetic approach called playback experiments where real animals interact with the virtual individuals, which can be robots or computer graphics displayed on a monitor. For example, Nakayasu et al. examined how much real medaka (small fish, *Oryzias latipes*) can be attracted by the computer graphics animation or the biological motions, which has been made from the actual behaviors of medaka (Nakayasu and Watanabe, 2014). They showed that the color, shape, and movements of the virtual individuals affect the real medaka's movements and change attractiveness.

However, the playback experiments cannot be our hybrid approach in a sense that real individuals just react to virtual one and there is no mutual interaction. In the real interactions, the behaviors that individuals have performed in the past affect the future behaviors of the others mutually. It is known that the mutuality is important for the interaction in the human and animal interactions. In this paper, we establish a hybrid experiment where the real and virtual individuals interact with each other and investigate if there is a difference between mutual and playback interactions. We used medaka for test animals for the sake of the simplicity

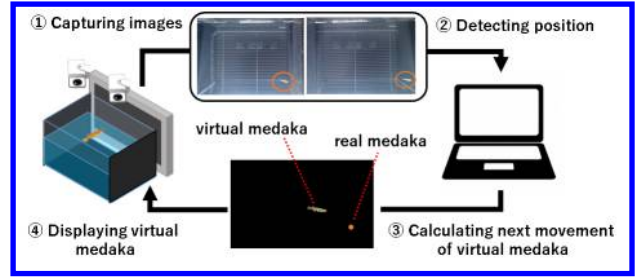


Figure 1: Experimental environment.

of establishing an experiment environment.

Experiment

Our experimental environment consisted of an aquarium tank, OLED display attached on the side of the tank, and two web cameras to detect the real medaka positions (Fig. 1). The glass walls at the bottom and both sides of the tank were covered with black plastic sheets to shut out the other visual stimulus. The three dimensional positions of the real medaka were detected by two cameras.

The movement of the virtual medaka was controlled by the Boids rules (Reynolds, 1987). Because the movements of the virtual medaka are restricted on the 2-D plane of the monitor, the positions of the real medaka is projected onto the 2-D plane of the monitor and the virtual medaka moves on the plane while following the Boids rules. The position and velocity vectors of the virtual medaka v_v , p_v were updated according to the Boids rules as follows,

$$\mathbf{a}_v^t = \alpha(\bar{\mathbf{p}} - \mathbf{p}_v) + \beta \sum_i \frac{(\mathbf{p}_i - \mathbf{p}_v)}{\|\mathbf{p}_i - \mathbf{p}_v\|} + \gamma(\bar{\mathbf{v}} - \mathbf{v}_v) \quad (1)$$

where α , β , and γ are coefficients that describing the influence of cohesion, separation, and alignment, respectively. The bars show the averages of other individuals including real and virtual ones. In our current experiment, there are only one each for the real and virtual medaka, which means that the average position and velocity are equal to the position and velocity vectors of the real medaka. The virtual

medaka has the view range R , and moves as described above when the distance to the real medaka is smaller than R . Otherwise, it moves simply go straight and turn when reach the edge of the 2-D space. The motions of the virtual medaka was created by exchanging the medaka images, which were prepared using female 3D medaka model shared in figshare (Watanabe, 2017)

The experiment consisted of three different conditions, i.e., reactive (mutual), non-reactive (playback) and blank. No stimulus was presented to the medaka in the blank condition. Under the reactive condition, the virtual medaka moved in accordance with the Boids rules, which reflected the current real medaka motions. During this condition, the whole movements of the virtual medaka were recorded. The recorded video was replayed under the non-reactive condition. It means that the visual stimuli to the real medaka of the reactive and non-reactive period were identical. The difference is that the movements of the virtual medaka in the reactive period were generated in an online manner as a result of ongoing interaction while it was just playback in the non-reactive period. The experiment was performed in the order of blank (blank1), reactive, blank (blank2), non-reactive conditions for all medaka participants ($N = 10$). Each condition lasted three minutes.

Results

We recorded the three dimensional positions of the real medaka during the experiments. Figure 2 shows the examples of the spatial distributions of the real medaka during reactive and non-reactive conditions. The colors of the plots indicate the passage of the time. The medaka was attracted to the display for longer time when the virtual medaka moved in a reactive way. On the other hand, when the non-reactive stimulus was presented, the medaka was hardly attracted to the display.

In order to confirm the effect of mutuality, we measured the total time when the medaka staying near the display (within 3cm from the display) during each stimulus period. Figure 3 shows the average staying time of all 10 medakas during the period of each condition. The medaka was most attracted to the display when the reactive stimulus was presented (red line, about 78 sec over 3 min in total). Soon after the reactive period, the real medaka tended to stay close to the monitor as the hysteresis effect but gradually it went away from the monitor. The staying time in the playback period (blue line, about 46 seconds in total) was longer than the blank1 period (black line, about 21 sec in total). These results show that the mutuality contributed to attract the medaka.

Discussion

In our current study, the virtual medaka was controlled by the Boids rules which can generate realistic swarm behaviors. The hybrid approach shows that there is a difference

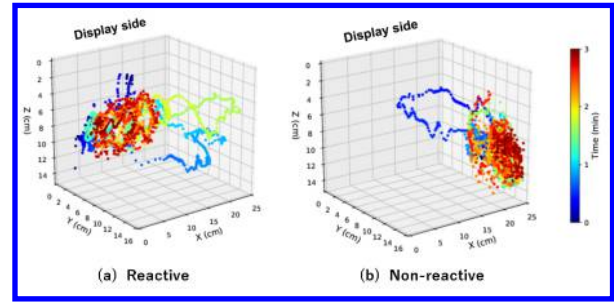


Figure 2: Examples of spatial distribution of real medaka during (a) reactive and (b) non-reactive periods.

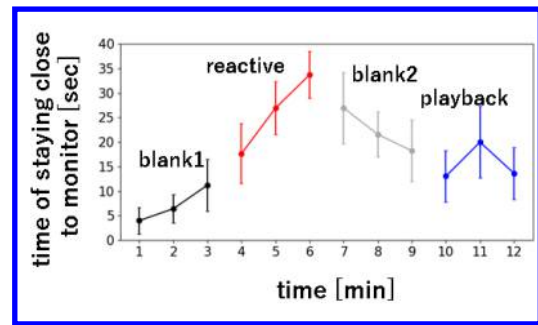


Figure 3: Averaging time when the distance between the real medaka and the monitor is less than 3 cm. It is obtained every minute. The bars show the standard errors.

in the medaka reactions between reactive and non-reactive experiments. It shows that the Boids interaction rules have an ability to establish an interaction with the real medaka. It seems that these abilities of generating realistic swarm behaviors among multiple agents and establishing an interaction with real individuals are different but our results suggest that they might be related. On the hand, it is not clear if the Boids algorithm is the best for our experiment. The reason why we chose it is because there is no choice in the other simple algorithms. To find the better algorithms becomes an interesting challenge but it would not be simple. This could be achieved by having ALIFE challenges as a competition.

References

- Langton, C. (1989). Artificial life. *Artificial life*, pages 1–47.
- Nakayasu, T. and Watanabe, E. (2014). Biological motion stimuli are attractive to medaka fish. *Animal cognition*, 17(3):559–575.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34.
- Watanabe, E. (2017). Medaka fish tracker and fish 3d polygon models. *figshare. Fileset*.

An artificial life approach to studying niche differentiation in soundscape ecology

David Kadish and Sebastian Risi and Laura Beloff
IT University of Copenhagen, Copenhagen, Denmark
davk@itu.dk

Abstract

Artificial life simulations are an important tool in the study of ecological phenomena that can be difficult to examine directly in natural environments. Recent work has established the soundscape as an ecologically important resource and it has been proposed that the differentiation of animal vocalizations within a soundscape is driven by the imperative of intraspecies communication. The experiments in this paper test that hypothesis in a simulated soundscape in order to verify the feasibility of intraspecies communication as a driver of acoustic niche differentiation. The impact of intraspecies communication is found to be a significant factor in the division of a soundscape's frequency spectrum when compared to simulations where the need to identify signals from conspecifics does not drive the evolution of signalling. The method of simulating the effects of interspecies interactions on the soundscape is positioned as a tool for developing artificial life agents that can inhabit and interact with physical ecosystems and soundscapes.

Introduction

Artificial life experiments have become important tools for exploring biological phenomena. In particular, they have allowed researchers to study the relationships between evolutionary processes and ecological theories (Aguilar et al., 2014), like the emergence of interspecies relationships like mutualism and parasitism (Watson et al., 2000).

One area of ecology that has received little attention thus far from artificial life (ALife) studies is soundscape ecology. The field of soundscape ecology has been formalized by researchers over the past decade (Pijanowski et al., 2011a), building on earlier conceptions of the soundscape (Schafer, 1977). One of its foundational theories is the acoustic niche hypothesis (ANH) (Krause, 1987), which applies the concept of ecological niches — the distribution of resources that are used by a species in an ecosystem (Pocheville, 2015) — to the soundscape.

This experiment tests the proposed mechanisms for the formation of these niches in a virtual soundscape in order to understand how species change vocalizations in response to one another. It models the behaviour of two species in a virtual ecosystem and tracks how their calls shift through

the audio spectrum in response to different evolutionary pressures. Through the experiment, evolutionary pressure to communicate within a species is found to play a significant role in the formation of acoustic niches.

In examining the emergence of communication between artificially evolved species, this study draws from a body (Arita and Koyama, 1998; Wagner, 2000; Sasahara and Ikegami, 2007) of ALife-based studies of communication including the work of Floreano et al. (2007) in emergent communication between robotic agents. However, it is distinct from these previous studies in its focus on the effect of the emergent communication on the ecological phenomena of niche differentiation.

The main contributions of this study are the development of a simplified model of a soundscape for the purpose of rapid experimentation and in-depth analysis of population-soundscape dynamics, and the demonstration of the ANH on this model.

Background

In the physical world, the concept of soundscape — the collection of the acoustic features of a landscape — has roots and influences in a diverse array of academic fields (Lyonblum, 2017). It grew initially out of the arts and cultural studies work of Westerkamp (1974), Schafer (1977), and Truax (1978), but has since expanded into the sciences. In the field of ecology, the soundscape is considered an important ecological resource and its composition is thought to indicate the diversity and stability of the ecosystem (Pijanowski et al., 2011b). Though the field of soundscape ecology was only proposed relatively recently (Pijanowski et al., 2011b), the application of ecological principles to the study of soundscape has a longer history. Notably, the concept of ecological niches was first introduced in the context of sonic resources by Krause (1987) as the acoustic niche hypothesis.

Acoustic niche hypothesis (ANH)

The acoustic niche hypothesis expands the concept of ecological niches to the spectro-temporal plane of the

soundscape. Krause proposed that, in the same way that niche differentiation leads to species making use of the range of physical resources available in an ecosystem, species tend to differentiate their use of an ecosystem's sonic resources. This differentiation, according to Krause, occurs spectrally in the sonic frequencies that animals use for vocalization and temporally in the time-based patterns of their sounds. The theory holds that older, more mature ecosystems should show a greater degree of differentiation between the auditory niches that long-established species occupy.

The ANH describes the result of acoustic differentiation, but Endler (1992) proposed the primary mechanism for this evolutionary driver: sexual selection based on a mate's ability to hear a call and the ability to maintain territory. In this formulation, vocalizations and auditory receptors have co-evolved to maximize the reception of signals from members of one's own species (conspecifics), while minimizing interference from members of other species (heterospecifics).

This type of spectral differentiation has been observed numerous times in the wild: in the calls of certain species of frogs (Feng and Schul, 2007); in the buzzing of cicadas (Sueur, 2002); and in the overall division of a soundscape in Borneo between a series of birds, gibbons, and accompanying insects (Krause, 2008). However, it has proved difficult to experimentally probe the formation and division of spectral niches, due to the lengthy timescales that would be required to allow evolutionary processes to progress (Miller, 1995) and the complexity of the systems and soundscapes that are encountered "in the wild" (Wheeler et al., 2002).

ALife approaches

Where ecological phenomena have been difficult to experimentally investigate, researchers have proposed that ALife approaches can be a mode of inquiry that allows for the manipulation of particular conditions and the rapid collection of large quantities of data about a simulated ecological system (Miller, 1995). In 2018, Eldridge and Kiefer proposed synthetic acoustic ecology (SAC) as a toolset for exploring questions in the field of soundscape ecology using ALife methods in virtual ecosystems. Their study examined one of the assertions of ANH (Krause, 1987) — that one can identify the maturity of an ecosystem by examining its acoustic signature. Using a multi-agent system model, they demonstrated that patterns emerge in two common acoustic indices that indicate the stability of a model ecosystem.

Niche differentiation mechanisms

The study in this paper uses a virtual soundscape to test hypotheses in soundscape ecology, building on the work of Eldridge and Kiefer (2018). While Eldridge and Kiefer's study focused on the verification of acoustic biodiversity

metrics, this study examines the mechanisms that breed interspecific diversity and intraspecific convergence in the vocalizations of communities in a soundscape. In particular, it is designed to test Endler's hypothesis (1992) that the ability to identify vocalizations from members of the same species drives acoustic niche differentiation.

The acoustic niche hypothesis posits that soundscapes niches are differentiated on both spectral and temporal levels, so that species ensure that their calls are isolated in both frequency and time. In order to simplify the modelling and analysis and to allow for a deeper examination of the effects of differentiation, this study focuses only on the spectral component of this differentiation.

Approach

The experimental setup for testing the drivers of acoustic niche differentiation consists of a set of evolving populations and a soundscape that they communicate within. The experiment tests two hypotheses: the alternative hypothesis (H_1), that acoustic niche spectral differentiation is driven a need to identify signals from potential mates or territorial rivals of the same species; and the null hypothesis (H_0) that spectral differentiation in acoustic niches is not driven by the need to distinguish the species of the signaller.

In order to facilitate rapid experimentation and ease the analysis of the emergent signalling systems, the experiments use a simplified, discretized model of a soundscape instead of a full-spectrum, temporally-varying acoustic space. Sounds are modelled as 9-bit vectors that represent the use of 9 available frequency bands in an instantaneous signal. These simplifications allow the repetition of the experiments many times with a large number of generations and individuals, such that results reflect general trends in the dynamics of these systems rather than the peculiarities of any single simulation. The entire system is illustrated in Figure 1 and described in detail in the sections below. Lettering in brackets refers diagram labels in Figure 1.

Populations

In soundscape ecology in the physical world, the communicative process is often assessed in two parts: sender and receiver. Every individual, of course, is both sender and receiver, but the processes experience different evolutionary pressures; "[n]atural selection favors signals that elicit a response in the receiver that increases or maintains the fitness of the sender" (Endler, 1992). The same is true in reverse, such that the sender and receiver of a particular species evolve alongside one another, but with slightly different driving forces.

The populations in this experiment are modelled as artificial neural networks, which are optimized with the neuroevolution of augmenting topologies (NEAT) algorithm (Stanley and Miikkulainen, 2002). NEAT models individual phenotypes as neural networks with a fixed

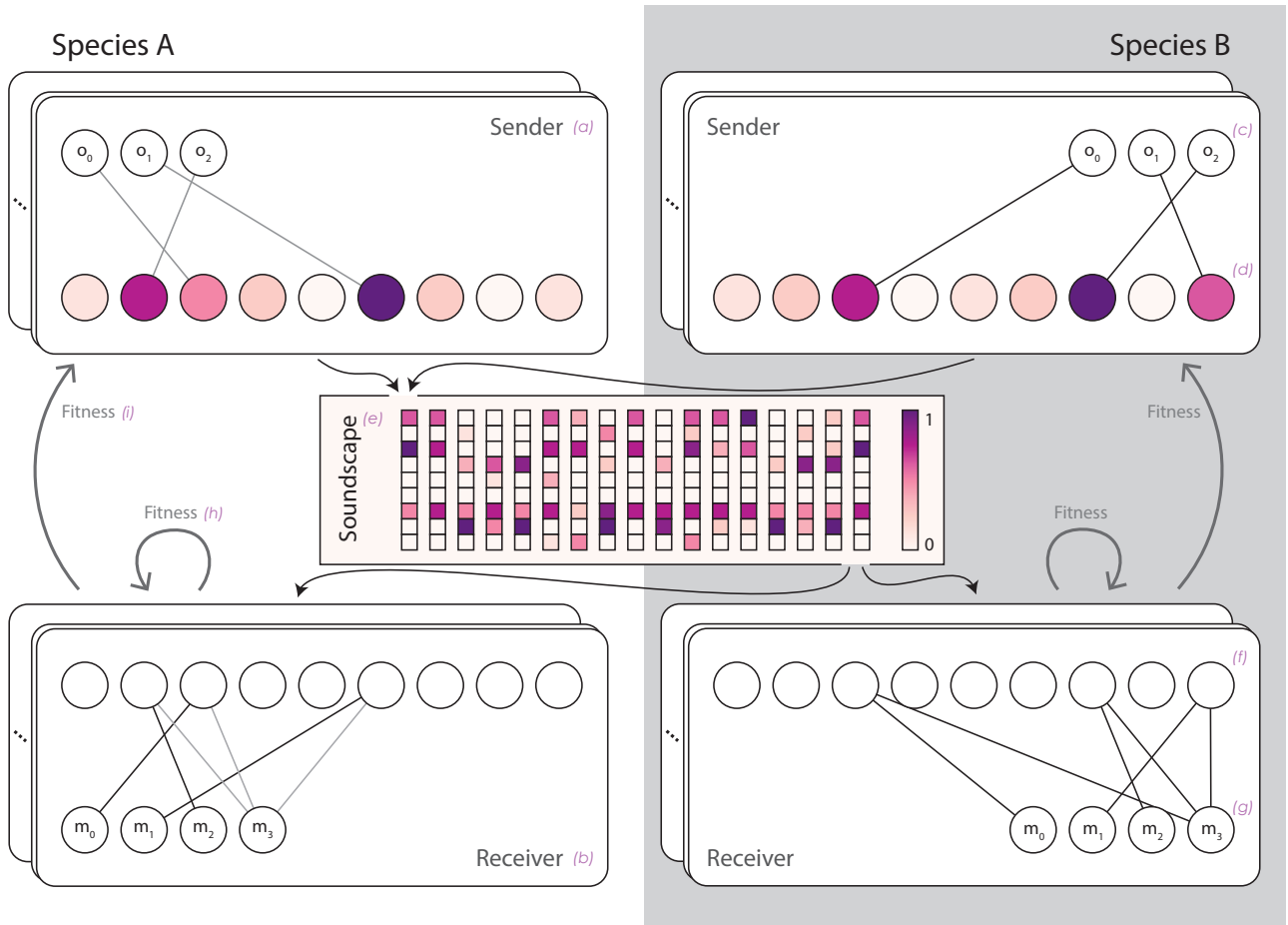


Figure 1: The experimental setup. Senders (a) encode a 3-bit message ($o_{0..2}$) into the 9-band soundscape (e) using a neural network with 3 inputs (c) and 9 outputs (d). Receivers (b) “hear” encoded messages from all species’ senders and predict the original message ($m_{0..2}$) and whether the message originates from a conspecific (m_3) using the 9 inputs (f) and 4 outputs (g) of their neural networks. Sender fitness (i) depends on how well conspecific receivers identify their species and decode their messages. Receiver fitness (h) depends on how well they identify the species of all senders and how well they decode messages from conspecifics.

number of inputs and outputs and an evolvable internal structure and connectivity. This allows the population to begin with simple neural structures and to evolve complexity as necessary to achieve the task.

For the experiment presented here, each species actually consists of two NEAT populations: a population of senders (a in Figure 1) and a population of receivers (b in Figure 1). The populations communicate over a simulated soundscape (e) that consists of 9-bit vectors, interpreted as acoustic frequency bands which can be used by senders to transmit messages. The senders encode 3-bit messages ($[o_0, o_1, o_2]$) into a representation in the 9 frequency bands using their 3-input (c), 9-output (d) neural network structures. The structure of 2 species encoding 3-bit messages into a 9-band soundscape allows for the development of relatively complex messaging while allowing the soundscape to remain undersaturated as each species could theoretically communicate in only 3 of the 9 bands. The frequency bands form the inputs to the 9-input (f), 4-output (g) receiver neural networks. The first 3 outputs ($[m_0, m_1, m_2]$) of the receiver network are its estimation of the original message and the final output represents the receiver's prediction of whether the message comes from a conspecific ($m_3 \geq 0.5$) or from a heterospecific individual ($m_3 < 0.5$).

The soundscape (e) is shared among species but messages are received serially in order to decouple timing effects; therefore, each receiver "hears" messages from the senders of all of the present species, but receives them one at a time. Additionally, any spatial arrangement of the individuals is not considered as part of this experiment, so each receiver "hears" the signals from every sender at the same "volume" with no attenuation due to a distance or set of obstacles between them.

Fitness

In a communicative process, the evolutionary pressure on senders and receivers is related but differs in some crucial aspects. The fitness functions used in this experiment reflect these differences. Since communication for mating and territorial maintenance is hypothesized to drive acoustic differentiation (Endler, 1992), the sender is indifferent to how its messages are interpreted by receivers from other species. The receiver, however, processes all messages regardless of their origin; it has to learn how to differentiate messages from conspecifics from those of heterospecifics.

Following this reasoning, the fitness of the sender (i) is formulated to reflect how well its message is understood — or correctly decoded — by the receivers of its own species; it does not depend on how the receivers of the another species process its messages. The fitness of the receiver (h) reflects both how well it is able to distinguish the species of the sender *as well* as whether it is able to correctly decode the message.

The ability of a receiver to perform these two tasks

— identifying messages from conspecifics and decoding messages — is formulated into components of the the fitness function as f_s (species identification fitness, Equation 1) and f_d (message decoding fitness, Equation 2). m is the decoded message where the first three components ($m_{0..2}$) are message as decoded by a receiver. The fourth value output by the receiver (m_3) determines whether the receiver has identified this message as coming from a conspecific ($m_3 > 0.5$) or from a member of another species. The original message is a three-bit string represented by o_i .

$$f_s(m) = \begin{cases} f_{adj}(1 - |1 - m_3|) & \text{if same species} \\ f_{adj}(1 - |0 - m_3|) & \text{if different species} \end{cases} \quad (1)$$

$$f_d(m) = 3 * \prod_{i=0}^2 f_{adj}(1 - |o_i - m_i|) \quad (2)$$

To achieve the desired fitness formulations, these equations are applied in different ways for senders and receivers by adjusting the enabling/disabling coefficients e_s and e_d in Equation 4. For each message produced, a sender's fitness is based on the interpretation of the message by all receivers *from its own species*. Equation 4 is applied for each receiver from the sender's species with $e_s = 1$. The value of e_d depends on whether the species is identified incorrectly ($e_d = 0$) or correctly ($e_d = 1$). If the species is incorrectly identified, then the interpretation of the message is of no consequence, which is why the fitness of the message decoding is ignored.

Receivers "hear" messages from the senders from both species and their ability to identify and ignore messages that are not from their species is an important component of their fitness. For each message that a receiver "hears", f_s is calculated as part of its fitness ($e_s = 1$). If the receiver correctly identifies that a message originated from a member of its own species, it receives an additional score for decoding the bits of the original message (f_d) and a bonus multiplier (f_b) for correctly identifying multiple bits ($e_d = 1$), as described in Equation 4.

$$f_b(e_s, N) = \begin{cases} 1.0 & \text{if } e_s = 0 \\ \prod_{i=0}^N \left(\frac{i}{10} + 1\right) & \text{if } e_s = 1 \end{cases} \quad (3)$$

$$f_t = (e_s f_s(m) + e_d f_d(m)) * f_b(e_s, N) \quad (4)$$

One detail that requires some explanation is the adjustment function (f_{adj}) applied to the fitness equations for species identification (f_s) and message decoding (f_d). The results that these equations evaluate are treated as binary in the operation of the system but the receivers produce output as decimal numbers between 0 and 1. If the receiver outputs $m_3 = 0.6$ for a message from a member of its own species, the consequence is no different from $m_3 = 1.0$ —

the receiver has correctly decided that the message should not be ignored. However, an application of Equation 1 without f_{adj} would result in quite different fitnesses for the two outputs. Equation 5 creates a sharp rise in the fitness, centred around a value of 0.5 without producing a discontinuity, which was found to create an effective fitness landscape for the evolutionary process.

$$f_{adj}(x) = \frac{1}{2} (\tanh(8.0 * (x - 0.5)) + 1) \quad (5)$$

Null model and hypothesis

The model used to test the null hypothesis (H_0) uses a modified formulation of the fitness functions. The null hypothesis is that the need to identify messages from members of the same species *does not* play a role in niche differentiation. Therefore receivers are assumed to be able to know *a priori* which messages come from senders of their own species and no fitness is assigned for the task of species identification in this null model.

In the null version of the model, this results in the receivers only processing messages from members of their own species and ignoring messages from the other species. Senders and receivers are evaluated with the fitness function in Equation 4 with $e_s = 0$.

Results

We ran simulations of our ecosystem with senders and receivers for two species. Each population consisted of 50 individuals and the simulation was run for 300 generations. The results discussed here are averages and standard deviations from 20 independent simulations. Additionally, the results from a representative example simulation are highlighted in figures and throughout this section in order to discuss specific features of an individual simulation.

For each simulation, we generated spectrograms that mirror the type of chart that is often presented in studies of soundscapes (Krause, 1987; Pijanowski et al., 2011a), except that the x -axis of these plots represents generations instead of real-time auditory signals. These diagrams, such as the one seen in Figure 2, show how the two species's use of the frequency bands shifts from generation to generation. The initial populations's encoded messages are randomly distributed across the 9 frequency bands, but the signals converge over the course of the first 50 to 100 generations into a subset of bands used primarily by one species. In this example, after an initial series of about 100 generations, both species show consistent use of 3 bands — 0, 2, and 4 for *Species A* and 1, 5, and 7 for *Species B* — for the remainder of the simulation. *Species A* develops and then eventually scales down the use of band 3 and band 8, but *Species B*'s use of 1, 5, and 7 remains remarkably stable through most of the latter 200 generations.

The spectral plots provide a useful visual representation of the divergent signals, but the actual level of separation

can be quantified further and visualized in another manner. Figure 3a shows a mapping of the high-dimensional messages to two-dimensions using t-distributed Stochastic Neighbour Embedding (t-SNE), plotted for particular generations of interest. The encoded messages generated by senders from the two species can be seen to rapidly separate into clusters from an initial state of near-total overlap. This can be further examined in the plot below the cluster maps (Figure 3b) which shows the silhouette score for the clusters over the course of generations. The silhouette score is used in the evaluation of clustering algorithms and is a measure of the density of clusters (Rousseeuw, 1987), where a score of 0 indicates overlapping data and a score of 1 indicates dense and well-separated clusters. The rapid rise of the silhouette score here indicates the splitting of the spectrum audio spectrum between the senders in relatively few generations.

The plot shows the average and standard deviations of the silhouette scores from the series of 20 trials of H_1 (dark grey) alongside the silhouette score from the specific run from which the clusters in the plot above were derived (pink). In addition, it shows the average and standard deviation of silhouette scores from 20 trials of the null model H_0 (light grey). A test of the hypotheses using Welch's t-test — because the variance of the samples cannot be assumed to be equal — reveals that the difference between the two models is significant after generation 4 ($P < 0.01$), with an average P-value of 15×10^{-5} for latter 295 generations.

While the null model does produce a level of clustering of the species' messages, this is to be expected as a result of the selection of frequency bands on which to communicate. However, in the null model, this selection is not competitively driven by the presence of the other species. In H_1 , the receivers of the two species drive their senders towards diverging frequency bands as their fitness increases with their ability to identify messages from their own species and reject those from the other.

We also examined the actual performance of the species with regard to their ability to recognize and decode messages from their conspecifics. Figure 4 shows the scores of the senders and receivers from a species over the course of 300 generations. On average, the proportion of messages that are correctly identified as being from members of the same or other species (red) rises sharply in the first generations before steadying near 80%. The proportions of bits that are correctly decoded and messages that are fully decoded correctly are slower to rise, but continue to do so throughout most of the evolutionary process.

Discussion

The results presented in the previous section demonstrate that it is possible to drive spectral differentiation in the acoustic signature of an agent through an impetus to communicate with other members of the same species. An analysis of the distance between intraspecies messages

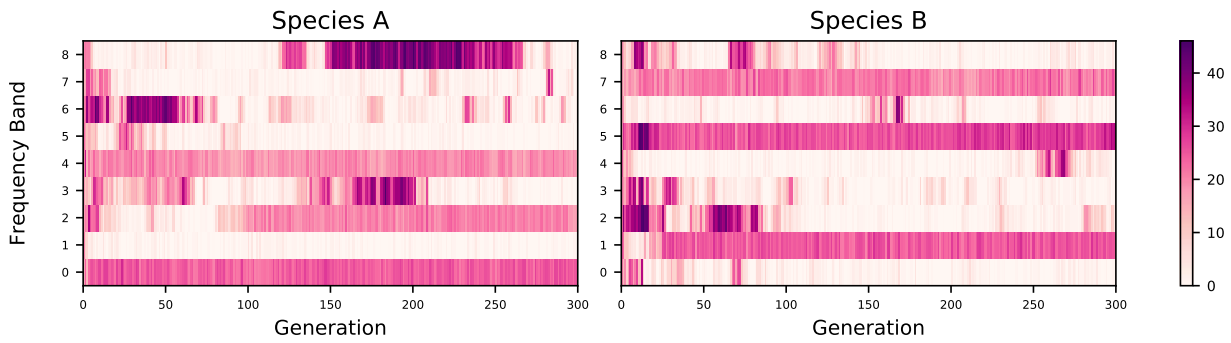


Figure 2: The use of the 9 frequency bands by the population of senders in the example simulation of H_1 . The graph shows the changing use of the frequency bands over 300 generations. In the first generations, both species’s signals are spread across the 9 bands such that the signals from the two species overlap. These signals converge rapidly to a smaller subset of the available bands. By about the 50th generation, there is little overlap between the two species — *A* uses mainly bands 0, 4, and 6 while *B*’s signals are concentrated on 1, 5, and 7 — though there is some use of bands 2 and 3 by both species. In this example, *Species B* uses band 2 heavily but intermittently until just before the 100th generation, when it ceases almost all activity on the channel and *Species A* begins to make consistent use of it for the remainder of the simulation. By the generation 300, both species have converged to the near-exclusive use of 3 channels: 0, 2, and 4 for *Species A* and 1, 5, and 7 for *Species B*.

and interspecies messages shows a significant difference between the test of the alternative hypothesis (H_1) and the null hypothesis (H_0), as seen in Figure 3b. Moreover, Figure 2 provides a visual reference for the division of the spectrum in a selected simulation of the alternative hypothesis (H_1). The spectrum has been split between the two species after the first 100 generations, such that *Species A* primarily makes use of bands 0, 2, and 4 while *Species B* relies on bands 1, 5, and 7. It is interesting to note that, in the first 100 generations, band 2 is used mainly by *Species B*, however this changes around generation 90 as *Species A* begins to use the band regularly. Once *Species A* establishes regular use of the band, *Species B* never returns to it with any stability for the remainder of the simulation.

In models of the null hypothesis (H_0), the two species occasionally achieve a level of differentiation of their messages, however this occurs only by chance. In both models, species tend to converge to the primary use of roughly 3 of the 9 available channels for communication. Three channels is the fewest that can be used to encode the three-bit message and it is often the easiest solution for the evolving neural networks to find. However, in the null case, the channel selection is not driven by competition between the species, only by cooperation within a species. This lack of competition often leads to overlapping channel selections, which in turn, is responsible for the lower silhouette scores for the null models (Figure 3b).

While these results cannot be taken as confirmation of the proposed mechanism of the ANH, they demonstrate that the mechanism is plausible. The drive to produce signals that are identifiable and understandable to members of one’s own species within the finite resource that is a soundscape results

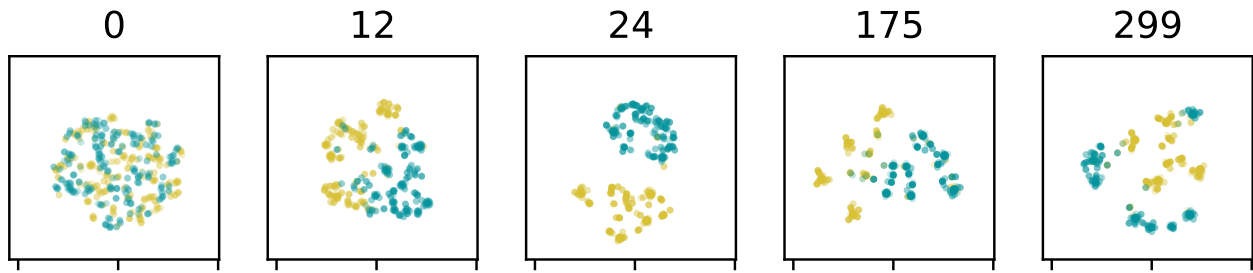
in the formation of acoustic niches for vocalizing species.

This study also demonstrates the efficacy of a highly simplified model in demonstrating the plausibility of a particular mechanism for the formation of patterns within a soundscape. It compliments the work of (Eldridge and Kiefer, 2018), which explores the way that common acoustic indices respond to changing populations and signals, and presents another application for a synthetic acoustic ecology. Together with other types of computational studies of soundscapes (Eldridge and Kiefer, 2018), this paper lays the foundation for a method of rapidly interrogating evolutionary acoustic processes. In addition to providing insight into ecological studies, research in this area can also be used to inform the development and analysis of evolutionary acoustic agents live “in the wild” and interact with biological ecosystems.

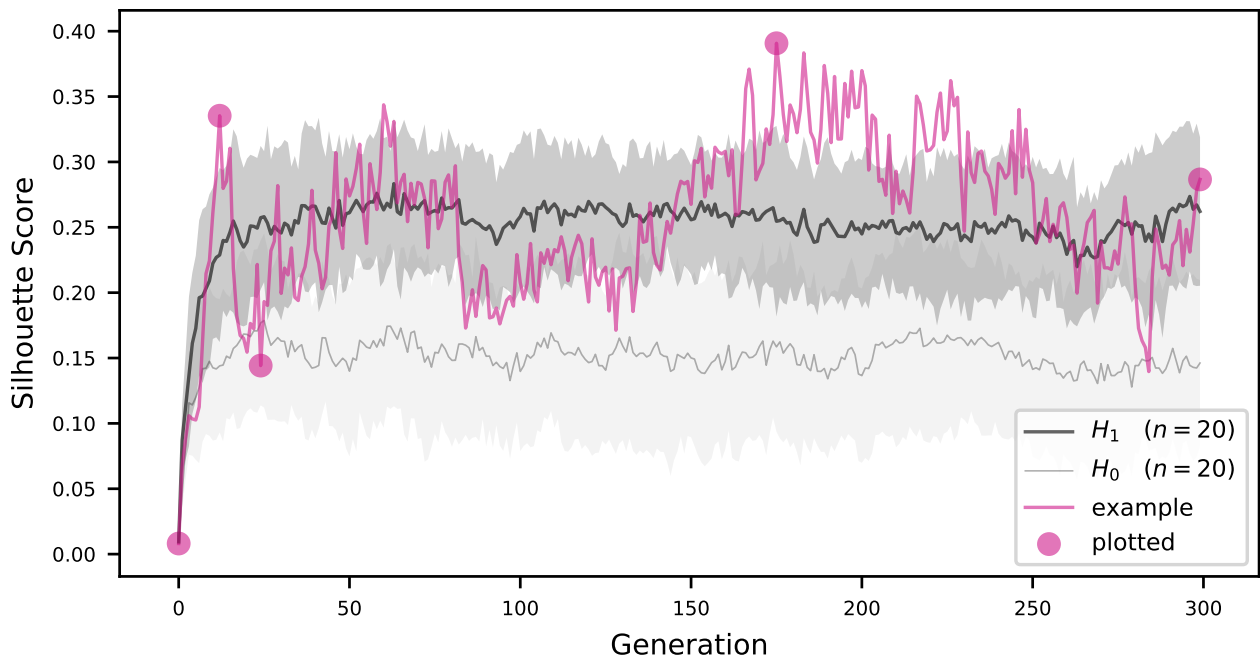
Conclusion

Though the experiment presented here is based on a highly simplified model of a physical ecosystem, it demonstrates that it is possible to rapidly and repeatedly test some of the basic principles of soundscape ecology. As predicted, the experiment was able to demonstrate the important role of intraspecies communication in the partitioning of the acoustic resources of an ecosystem.

This has important implications for the development of hardware-based ALife agents for the production of sound in a physical, hybrid ecosystem. It suggests that, if one of the goals of that agent is to identify a niche for itself in the soundscape, it is important to co-evolve the auditory production with auditory perception to drive the vocalizations into an empty portion of the spectrum.



(a) Cluster diagrams of messages in selected generations, mapped to 2 dimensions using t-SNE. Each point is a message generated by a sender in a single generation (labeled above the plot) of the simulation. The selected generations are marked on the silhouette score plot below with pink circles. The different colours represent messages originating from members of the two different species. The messages in generation 0 are scattered randomly from both species as the initial neural network connections for the senders are randomly generated. The messages rapidly converge to two clusters by generation 12. However, these clusters are still evenly spaced internally, as the initial selection pressure is mainly to differentiate messages between the two species. In later plots, for example in those from generations 175 and 299, smaller clusters form within the messages from a single species as the senders from each species converge on representations for particular bits and messages. This clustering drives the increasing bit and total scores in Figure 4. Note that the t-SNE and silhouette scores operate differently on high-dimensional data, which is the likely cause of an apparent mismatch between some of the diagrams in Figure 3a and Figure 3b. The t-SNE is a 2D visualization of high-dimensional data, while the silhouette scores reflects cluster validity in n -dimensions which is why, for example, the clusters in generation 24 appear visually more distinct than those in generation 12, even though the silhouette score in generation 24 is lower.



(b) Silhouette score of the encoded messages, grouped by species, over the course of 300 generations. Scores reflect the validity of the message clusters when grouped by species, averaged over 20 runs of the simulation, and plotted with the standard deviation in the background. An example of an individual run is also plotted (pink) and the generations of that run that are plotted in the cluster diagram above are noted. The difference between the alternative hypothesis (H_1) and the null hypothesis (H_0) is significant ($P < 0.01$) after generation 4. The average P-value after generation 4 is 15×10^{-5} .

Figure 3: Cluster validity scores over 20 runs of the simulation. Message clusters are shown above for selected generations of an example run.

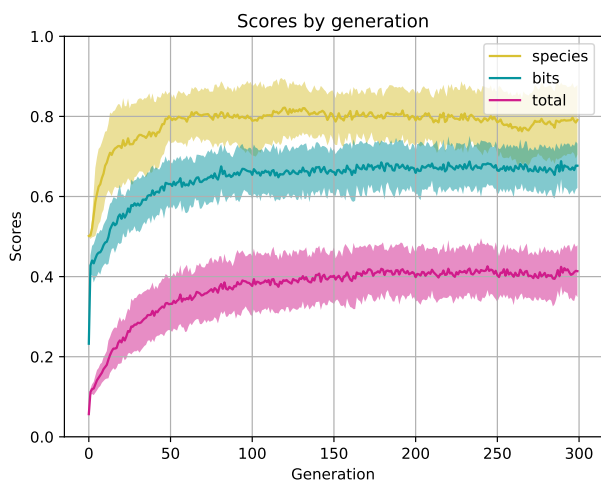


Figure 4: Performance of a species, showing the receivers' ability identify the species (yellow), and the rate at which sender-receiver pairs were able to correctly identify individual message bits (blue) and the entire message (pink).

In a broader sense, this experiment sets out the foundation for a method of testing ideas for hardware-based agents in software simulations to understand the possible dynamics once they are released in the field. It grounds the inquiry into a complex phenomenon with a concrete example that solidly demonstrates the theoretical basis for a physical experiment through repetition and statistical analysis on a scale that is difficult to achieve in the field. And it demonstrates the feasibility of a key theory in soundscape ecology.

Acknowledgements

Thanks to the REAL Lab, especially Kasper Støy, Rosemary Lee, Djordje Grbic, Miguel Gonzalez Duque, Mads Johansen Lassen, and Niels Justeen for their suggestions.

References

Aguilar, W., Santamaría-Bonfil, G., Froese, T., and Gershenson, C. (2014). The Past, Present, and Future of Artificial Life. *Frontiers in Robotics and AI*, 1:8.

Arita, T. and Koyama, Y. (1998). Evolution of Linguistic Diversity in a Simple Communication System. *Artificial Life*, 4(1):109–124.

Eldridge, A. and Kiefer, C. (2018). Toward a Synthetic Acoustic Ecology: Sonically Situated, Evolutionary Agent Based Models of the Acoustic Niche Hypothesis. In *The 2018 Conference on Artificial Life*, pages 296–303, Cambridge, MA. MIT Press.

Endler, J. A. (1992). Signals, Signal Conditions, and the Direction of Evolution. *The American Naturalist*, 139:S125–S153.

Feng, A. S. and Schul, J. (2007). Sound processing in real-world environments. *Springer Handbook of Auditory Research*, 28(Bregman 1990):323–350.

Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). Evolutionary Conditions for the Emergence of Communication in Robots. *Current Biology*, 17(6):514–519.

Krause, B. L. (1987). Bioacoustics, Habitat Ambience in Ecological Balance. *Whole Earth Review*, (57):14 – 18.

Krause, B. L. (2008). Anatomy of the soundscape: Evolving perspectives. *AES: Journal of the Audio Engineering Society*, 56(1-2):73–80.

Lyonblum, E. Z. S. (2017). *An Omnivorous Ear: The Creative Practice of Field Recording*. Doctor of philosophy (phd), University of Cambridge.

Miller, G. (1995). Artificial life as theoretical biology. How to do real science with computer simulation. *Cognitive Science Research Paper*, 378(5):1–33.

Pijanowski, B. C., Farina, A., Gage, S. H., Dumyahn, S. L., and Krause, B. L. (2011a). What is soundscape ecology? An introduction and overview of an emerging new science. *Landscape Ecology*, 26(9):1213–1232.

Pijanowski, B. C., Villanueva-Rivera, L. J., Dumyahn, S. L., Farina, A., Krause, B. L., Napoletano, B. M., Gage, S. H., and Pieretti, N. (2011b). Soundscape Ecology: The Science of Sound in the Landscape. *BioScience*, 61(3):203–216.

Pocheville, A. (2015). The Ecological Niche: History and Recent Controversies. In *Handbook of Evolutionary Thinking in the Sciences*, pages 547–586. Springer Netherlands, Dordrecht.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sasahara, K. and Ikegami, T. (2007). Evolution of Birdsong Syntax by Interjection Communication. *Artificial Life*, 13(3):259–277.

Schafer, R. M. (1977). *The tuning of the world*. A.A. Knopf.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, 10(2):99–127.

Sueur, J. (2002). Cicada acoustic communication: potential sound partitioning in a multispecies community from Mexico (Hemiptera: Cicadomorpha: Cicadidae). *Biological Journal of the Linnean Society*, 75(3):379–394.

Truax, B. (1978). *The World Soundscape Project's Handbook for acoustic ecology*. Music of the environment series. A.R.C. Publications, 1 edition.

Wagner, K. (2000). Cooperative Strategies and the Evolution of Communication. *Artificial Life*, 6(2):149–179.

Watson, R. A., Reil, T., and Pollack, J. B. (2000). Mutualism, Parasitism, and Evolutionary Adaptation. In Bedau, M., editor, *Artificial life VII : proceedings of the seventh International Conference on Artificial Life*, page 564. MIT Press.

Westerkamp, H. (1974). Soundwalking. *Sound Heritage*, 3(4).

Wheeler, M., Bullock, S., Paolo, E. D., Noble, J., Bedau, M., Husbands, P., Kirby, S., and Seth, A. (2002). The View From Elsewhere: Perspectives on ALife Modeling. *Artificial Life*, 8(1):87–100.

Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network

James C. Knight¹, Daniil Sakhapov², Norbert Domcsek¹, Alex D.M. Dewar¹,
Paul Graham¹, Thomas Nowotny¹ and Andrew Philippides¹

¹Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, UK

²Department of Computer Science, Tomsk State University, Tomsk, Russia

J.C.Knight@sussex.ac.uk

Abstract

Insect-Inspired models of visual navigation, that operate by scanning for familiar views of the world, have been shown to be capable of robust route navigation in simulation. These familiarity-based navigation algorithms operate by training an artificial neural network (ANN) with views from a training route, so that it can then output a familiarity score for any new view. In this paper we show that such an algorithm – with all computation performed on a small low-power robot – is capable of delivering reliable direction information along real-world outdoor routes, even when scenes contain few local landmarks and have high-levels of noise (from variable lighting and terrain). Indeed, routes can be precisely recapitulated and we show that the required computation and storage does not increase with the number of training views. Thus the ANN provides a compact representation of the knowledge needed to traverse a route. In fact, rather than losing information, there are instances where the use of an ANN ameliorates the problems of sub optimal paths caused by tortuous training routes. Our results suggest the feasibility of familiarity-based navigation for long-range autonomous visual homing.

1 Introduction

Visual homing – the ability to navigate back to a place of interest using visual information alone – is a problem of great interest for both engineers seeking to build autonomous robots and neuroethologists seeking to understand its neural basis in animals (Graham and Philippides, 2014). Solitary foraging ants are amongst the champion visual navigators in the animal kingdom (Wehner, 2009). Despite having small brains and low-resolution vision, these ants can learn visually guided routes many metres long through complex terrain (Knaden and Graham, 2016). However, in direct contrast with most modern robotic methods, insects use route knowledge not mental maps to navigate between two locations (Wehner et al., 2006). That is, insects learn procedural instructions for navigation: “What should I do here?” rather than “Where am I?”. This allows for simpler representations of the visual world with corresponding potential for computational efficiencies. We have shown that route knowledge can be learnt and represented holistically using an artificial neural network (ANN) (Philippides et al., 2015), without

specifying when or what to learn (Baddeley et al., 2011a) and from a single exposure to the route data (Baddeley et al., 2012). One of the reasons to use an ANN is that route knowledge can be encoded and used with memory and computational constraints that do not scale with route length, making ANNs well-suited to a small, power-efficient, robot. While our ANN-based algorithms have been tested in simulations of ant habitats, they had not previously been tested in the real world. In this paper, we show that a single layer ANN can autonomously guide a robot through outdoor routes of up to 10 m with all computation performed on-board.

Our route navigation algorithms start with two observations. First, if an agent stores a view when facing a given direction, the difference between this view and views from nearby locations will be minimised when the agent is facing the same direction as when the original view was stored (Zeil et al., 2003). Second, for ants and many wheeled robots, there is a fixed relationship between viewing direction and direction of travel, meaning that a view implicitly defines a movement direction. Therefore, when an agent is facing in a familiar direction, it is likely travelling in the correct direction. This allows the problem of navigation to be reframed in terms of a search for familiar views, that is, views that are associated with a previously learned route. Based on this, we have developed a parsimonious insect-inspired navigation algorithm in which a route, or routes, are learnt holistically and route recapitulation is driven by a search for familiar views (Baddeley et al., 2012).

The algorithm proceeds as follows: an agent equipped with a low-resolution 360° panoramic visual sensor first travels a route. The views it experiences along this route – crucially determined by both the agent’s positions and headings (poses) – are used to sequentially train an artificial neural network (ANN) which learns a holistic representation of the views encountered. Subsequently, the network is used to estimate the likelihood of whether a given view – and thus a pose – has been experienced before. When trying to repeat the route, the agent derives a direction of movement at a position by visually scanning the environment (either by physically rotating – a behaviour seen in ants (Wystrach



Figure 1: **A** Robot platform with onboard computation. **B** Environment used for robot testing.

et al., 2014) – or rotating the view ‘in silico’). Each rotated version of the current view is applied as an input to the network which outputs an estimate of its familiarity. The agent then moves in the direction corresponding to the view most similar to those encountered during learning.

Algorithms of this sort have been used to successfully learn routes through simulations of the habitats of solitary foraging ants, using a variety of neural networks for route encoding, ranging from restricted Boltzmann machines (Baddeley et al., 2011b) to a spiking model of a part of an ant’s brain known as the mushroom body (Ardin et al., 2016). The paths of these simulated agents were found to have many of the characteristics of the paths of ants (Wystrach et al., 2013). Here, we follow Baddeley et al. (2012) who showed that a single layer network trained with an Infomax learning rule (Bell and Sejnowski, 1995) can not only navigate robustly, but can learn multiple paths to a single goal after a single training run per path and with performance robust to sensor and motor noise. We choose to use this approach mainly because it only requires a single pass through the data, meaning that each view is experienced just once and then discarded. While this algorithm is well-suited for use on autonomous robots, it had not previously been tested in the real world. Here, we demonstrate that the robot shown in Figure 1 can navigate fully autonomously outdoors using only a single layer neural network, with all processing performed using a Jetson TX1 (NVIDIA Corporation, 2016) on-board the robot.

2 Methods

As described above, at the heart of our algorithm an agent navigates by sampling views from the current position at a number of headings and finding the direction that is deemed most familiar when compared to the views perceived along a training route. We find this most familiar direction in two ways. Firstly, we train a neural network to output a ‘decision function’ in response to rotated versions of the current view based on their familiarity to training views. This results in a rotational familiarity function (RFF – Figure 2) which, for each heading faced, gives a decision function value. The

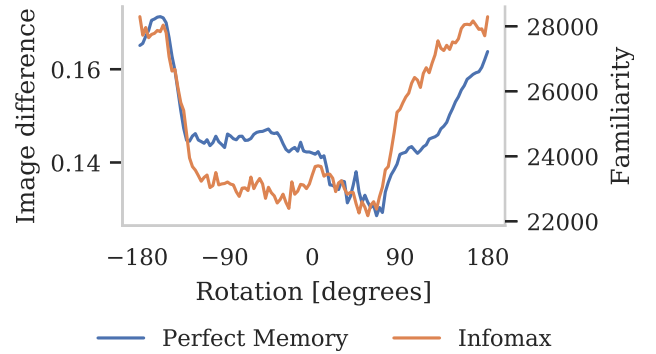


Figure 2: Example Rotational Image Difference Function (RIDF) from the Perfect Memory algorithm and Rotational Familiarity Function (RFF) from Infomax.

heading with the lowest decision function value is the most familiar and sets the movement direction taken. Secondly, as a baseline and because it allows easier interpretation of the data (for instance, to identify points in routes that are easy/hard or contain misleading information), we also implement what we previously termed the Perfect Memory model. In this variant, rotated versions of the current view are compared directly in a pixel-wise manner to every training view in turn (which would not be plausible for an ant). In the following sections we first describe the Perfect Memory algorithm before describing the ANN and Infomax learning rule.

2.1 Route navigation with a Perfect Memory

In our Perfect Memory version of the algorithm, each rotated view is sequentially compared to all of the training views. The best matching heading is then defined as the one with the lowest image difference, across all training views and rotations. Image difference can be calculated by various functions but here we use the average absolute difference between each of the image pixels to calculate the Image Difference Function (IDF) (Zeil et al., 2003):

$$IDF(C(\vec{a}, \theta), S(\vec{b}, \phi)) = \frac{1}{p \times q} \sum_{i=1}^p \sum_{j=1}^q |C_{i,j} - S_{i,j}| \quad (1)$$

where $C(\vec{a}, \theta)$ is a $p \times q$ pixel view captured at location \vec{a} with heading θ , $S(\vec{b}, \phi)$ is a $p \times q$ pixel snapshot stored in memory and $C_{i,j}$ and $S_{i,j}$ refers to the intensity of pixels in row i and column j of the captured view and stored snapshot respectively. A Rotational Image Difference Function (RIDF) is then generated by calculating the IDF across a range of θ . Where there is a good match, there will be a minimum in the RIDF which defines the best matching direction. An example RIDF is shown in Figure 2 which has a clear minimum at the best matching heading of around 60° . As the RIDF will also have local minima at other headings,

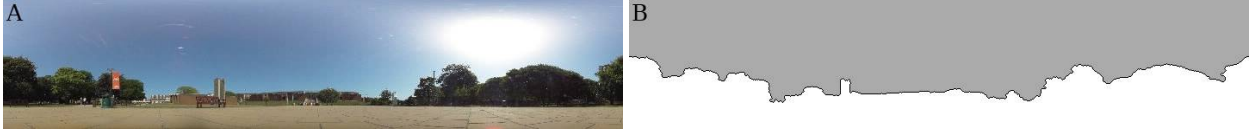


Figure 3: Example panoramic images from our database. **A** Raw unprocessed image. **B** Sky-segmented binary image.

we can use the RIDF to, for example, analyse whether errors in navigation were the result of visual aliasing.

2.2 Familiarity-based route navigation

To estimate view familiarity we follow Baddeley et al. (2012) and use a neural network model that was specifically designed to perform this task (Lulham et al., 2011). The network consists of an input layer and a novelty layer with $\tanh()$ activation functions. The number of input units is equal to the dimensionality of the input which in our case is $[120 \times 25] = 3000$, the number of pixels in a down-sampled view of the world. The number of novelty units is arbitrary and here we use the same number of novelty units as inputs, although using fewer novelty units has worked in simulation and will be tested in future work. The network is fully connected by feedforward connections w_{ij} . Weights are initialised randomly from a uniform distribution in the range $[-0.5, 0.5]$ and then normalised so that the mean of the weights feeding into each novelty unit is 0 and the standard deviation is 1. The network is then trained using the Infomax learning rule (Bell and Sejnowski, 1995), adjusting the weights so as to maximise the information that the novelty units provide about the input, by following the gradient of the mutual information using Equation 4 which performs gradient ascent using the natural gradient (Amari, 1998) of the mutual information over the weights (Lee and Sejnowski, 1997). During learning the activation of each of the M novelty units h_i is computed as:

$$h_i = \sum_{j=1}^N w_{ij} x_j \quad (2)$$

where x_j is a row vector assembled by concatenating the rows of $C(\vec{a}, \theta)$ and $N = p \times q$ (the number of input units). The output y_i of the novelty units is then:

$$y_i = \tanh(h_i) \quad (3)$$

and the weights are adjusted using:

$$\Delta w_{ij} = \frac{\eta}{N} \left(w_{ij} - (y_i + h_i) \sum_{k=1}^N h_k w_{kj} \right) \quad (4)$$

where η is the learning rate which is set as 0.01 for this paper following Baddeley et al. (2012). Finally, the response of the network to the presentation of an unseen N-dimensional

input \vec{x} is computed as

$$d(C(\vec{a}, \theta)) = d(\vec{x}) = \sum_{i=1}^N |h_i|, \quad (5)$$

where $|\cdot|$ denotes the absolute value. By applying $C(\vec{a}, \theta)$ to the ANN for a range of θ , an RFF can be calculated from $d(\vec{x})$ and hence the most familiar direction can be found.

2.3 Robot platform

In this work we use the robot platform developed by Domcsek et al. (2018) shown in Figure 1a. This robot is based on a Parallax ‘Shield-Bot’ chassis (Parallax Inc., 2012), with a Jetson TX1 embedded computer (NVIDIA Corporation, 2016) mounted on top for additional onboard computation and additional batteries mounted underneath. The Jetson TX1 is connected via USB to a Kodak PixPro SP360 4K camera (JK Imaging Ltd., 2016) mounted on top of the robot which provides panoramic visual input.

2.4 Image processing and data collection

Using a Kodak PixPro SP360 4K panoramic camera (JK Imaging Ltd., 2016), we recorded 195 images of the Library Square at the University of Sussex (Figure 3). These were taken on a 1.2m grid, aligned with the slabs the square is paved with. As well as this reference grid of images, we also recorded videos from the same camera mounted on the mobile robot as we manually drove it along six routes of varying lengths and tortuosity across the square (Figure 4a). We tracked the robot by using the Discriminative Correlation Filter Tracker with Channel and Spatial Reliability (Lukežič et al., 2018) implementation provided by OpenCV (Bradski, 2000) to extract the position of the robot over time from video captured by a tripod-mounted camera. Finally, we used OpenCV to apply a perspective transform to the positions extracted by the tracker and married these final positions with the video frames captured by the robot.

As it has previously been shown that the sky can give erroneous information for visual homing, and that visual homing can in fact be achieved using a binary image consisting of sky/not-sky (Philippides et al., 2011; Stone et al., 2014), we wanted to compare the use of raw and binary images. To achieve this in real-time on the robot, we used the watershed segmentation algorithm (Beucher, 1979) with markers placed at the top and bottom of each image. Figure 3 shows some example images from this database.

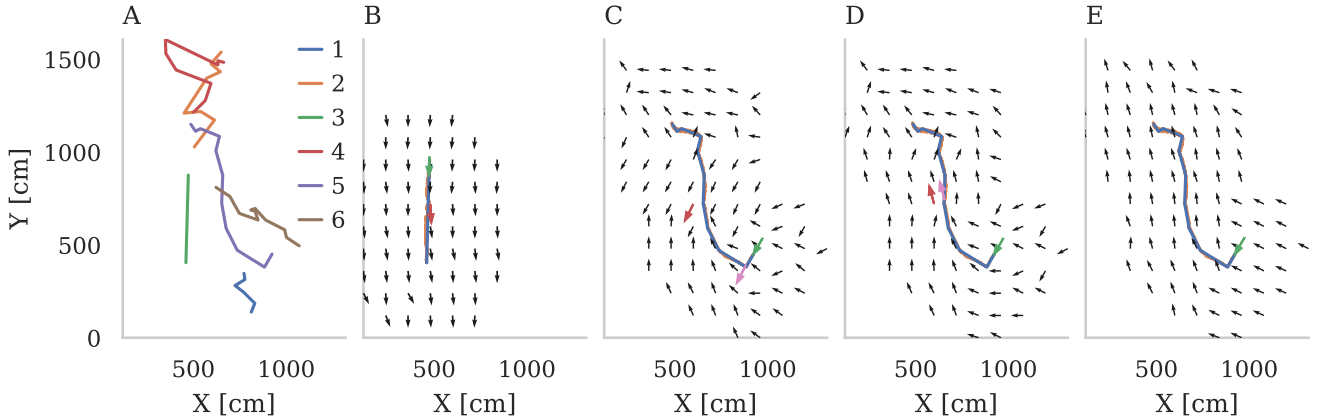


Figure 4: **A** The 6 numbered training routes, simplified using the Ramer-Douglas-Peucker algorithm (Ramer, 1972). **B-E** Vector fields showing the directions an agent, trained on a route, would move at each point within 4 m of the route using a Perfect Memory algorithm with the skyline extracted using the watershed algorithm. Orange lines shows data from our camera-based tracking of the robot and blue lines show the version simplified using the Ramer-Douglas-Peucker algorithm (Ramer, 1972). Green arrows indicate the position and direction the robot starts at. Red and pink arrows indicate positions in the vector field referred to in later analysis. **B** Simple route (route 3). **C** Longer route (route 5) where visual aliasing occurs in the middle section. **D** By only considering rotated views within 90° of the route, visual aliasing problems can be avoided in route 5. **E** Using Infomax, tortuous elements of route 5 are smoothed out in the vector field.

3 Results

3.1 Offline simulations

We trained both the Infomax and Perfect Memory algorithms on the images taken along the six routes shown in Figure 4a and used them to find the direction a robot would move when placed at each grid point within 4 m of the trained route. Figure 4 shows some example vector fields obtained by plotting this direction at each of the chosen grid locations when using the Perfect Memory algorithm described in Section 2.1 with the watershed segmentation-based pre-processing discussed in Section 2.4. While the vector field suggests that the route shown in Figure 4b would be recapitulated successfully, in the middle section of the route shown in Figure 4c, errors occur. Figure 5a shows an RIDF taken at one of the problematic locations on this route (indicated with a red arrow in Figure 4c) and it is clear that, as well as the local minimum representing the correct heading at 15° , there is an additional, slightly lower global minimum at 153° which is overriding the correct choice. Figure 5 shows the per-pixel differences between the best-matching image and the closest image (in terms of distance to the current point) taken from the training route; and the rotated versions of the current image which best match these training images. Although the *shape* of the skyline in the closest image (Figure 5c) is clearly better matched than it is with the best-matching image (Figure 5b), there is a vertical offset – probably caused by variations in pitch due to uneven terrain. This introduces sufficient difference between the grid

image and the correct route image (Figure 5c) that the Perfect Memory model matches the more distant image better. Therefore, at this grid position, the algorithm selects a direction corresponding to the more distant point (indicated with a pink arrow in Figure 4c), resulting in an error.

If we were recapitulating the route using a real robot, these false-positive matches could be eliminated and computation could be saved by simply *not* scanning the full $\pm 180^\circ$ but instead only scanning $\pm 90^\circ$ around the robot’s current heading. However, unlike in the live robot tests, in the database there is no ‘current’ heading. We therefore simulated the effect of this modified algorithm using our database of images by simplifying each route using the Ramer-Douglas-Peucker algorithm (Ramer, 1972) and calculating the direction of each of the resultant segments. Each point in the database is then assigned a ‘current’ heading which is equal to the direction of the nearest segment. Using these headings, we can then ignore matches that would involve heading more than $\pm 90^\circ$ away from the direction of the nearest section of the route and Figure 4d shows that this step solves the aliasing problems in this particular case.

In order to quantify the performance of our algorithms, we used the absolute difference between the most familiar heading angle obtained at each location on the grid and the direction of the nearest route segment as an error measure. The distributions of these errors across all of the routes, algorithms and image pre-processing steps is plotted in Figure 7 and shows that, in fact, only scanning $\pm 90^\circ$ im-

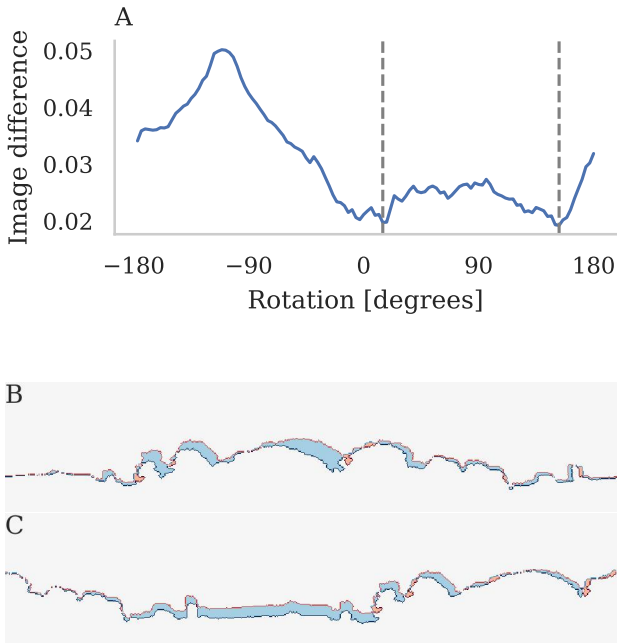


Figure 5: Analysis of aliasing shown in Figure 4c using grid image taken at location marked with red arrow in Figures 4c and 4d (600 cm, 720 cm). All images were pre-processed using watershed segmentation algorithm. **A** Rotational Image Difference function. **B** Difference image with visual aliased route image from location marked with pink arrow in Figure 4c. **C** Difference image with correct route image found at location marked with pink arrow in Figure 4d

proves performance in the majority of cases. Furthermore, Figure 7 also shows that, when using the Perfect Memory algorithm, sky-segmentation improves performance for almost all routes. This is particularly noticeable on route 3 (a straight line) where, when using sky-segmented input images, all of the algorithms reconstruct the direction of the route with a very small median error ($\approx 4^\circ$) which increases to around 40° when using unprocessed input images. Figure 6a shows that, unlike the situation explored earlier in this section, this is not an aliasing problem as there is only a single minimum present in each RIDF. However, the magnitude of the average image differences between the raw images is much larger than that between the sky-segmented images and the minimum is located at the incorrect location. The difference images shown in Figures 6b and 6c suggest that the incorrect position of the minima when comparing the raw images is likely to be due to the large differences in the sky portion of the image due to clouds and the images having been recorded at a different time of day.

After applying sky-segmentation to the input images and only scanning $\pm 90^\circ$ away from the direction of the nearest section of the route, the Infomax ANN achieves a lower median error than the Perfect Memory algorithm in 3 of the

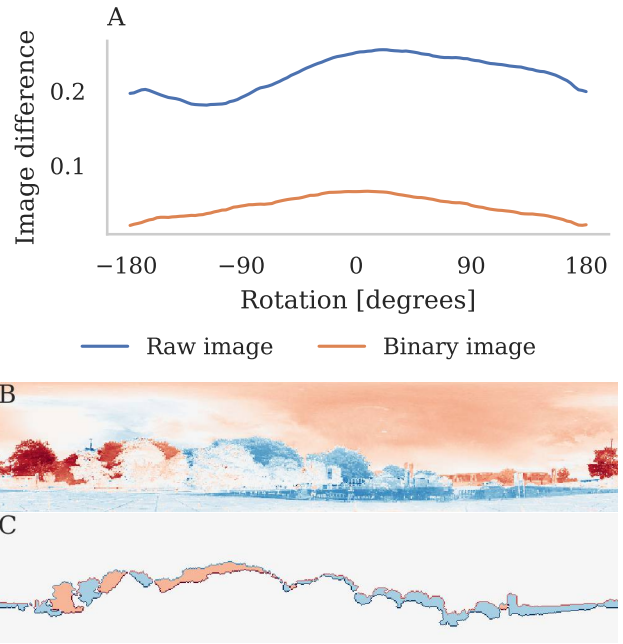


Figure 6: Analysis of poor performance on route 3 when using raw input images. Using grid image taken at location marked with red arrow in figure 4b (480 cm, 720 cm). **A** Rotational Image Difference function. **B** Difference image based on raw input images. **C** Difference image based on sky-segmented binary input images.

6 routes although this difference is not statistically significant ($p = 0.84$, paired, two-sided Wilcoxon signed rank test, $n = 6$). Furthermore, comparing Figures 4d and 4e, the vector field derived from the Infomax ANN appears smoother, suggesting that rather than losing useful information, the Infomax encoding may actually be beneficial for smoothing out tortuous training routes.

3.2 Computational cost of algorithm

In the previous section, we showed that the algorithms described in Sections 2.1 and 2.2 are capable of robustly extracting heading direction in outdoor scenes. However, in order to deploy these algorithms on a real robot with constrained on board processing, their computational cost is important. In order to measure computational cost in a controlled scenario, we wrote a benchmarking application in C++ which runs on the Jetson TX1 and trains either the Perfect Memory or the Infomax algorithm with varying numbers of images and then measures how long it takes to extract a heading from a testing image (averaged over 100 iterations) using `std::chrono::high_resolution_clock`. Figure 8 shows the performance of our implementation of the Infomax algorithm compared to the Perfect Memory control using 120×25 pixel input images. Clearly, for small num-

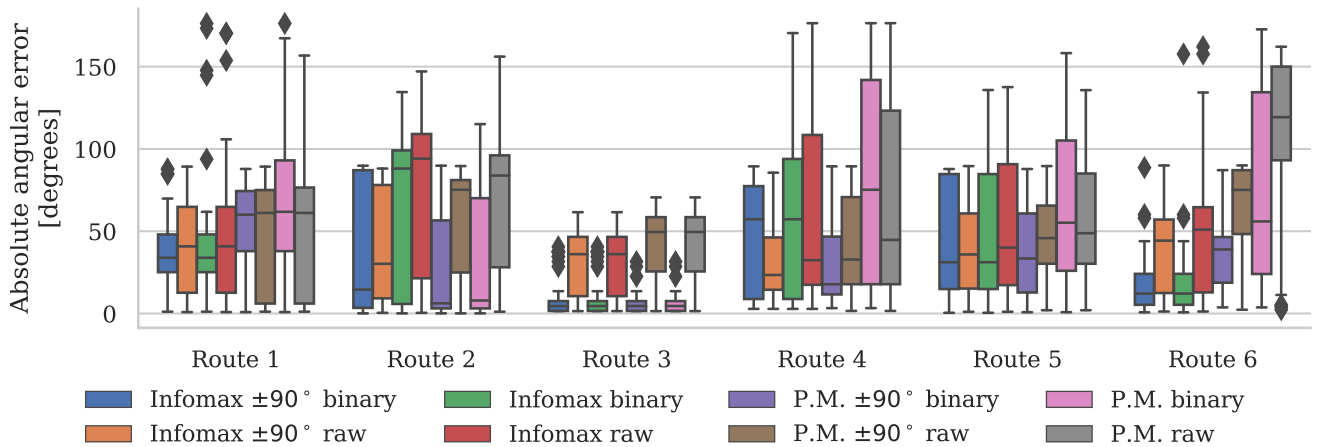


Figure 7: Performance of different algorithms on each of the 6 routes. ‘Boxes’ indicates the three quartile values of each error distribution, ‘whiskers’ indicate error values which lie within 1.5 IQRs of the upper and lower quartiles and values outside of this range are marked as outliers.

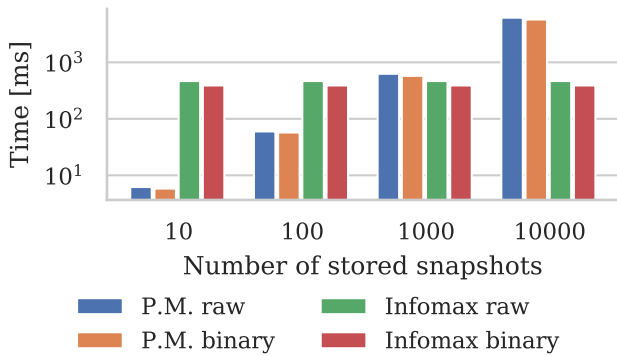


Figure 8: Performance of visual navigation algorithms running on Jetson TX1. Reported times are measured using `std::chrono::high_resolution_clock` and average is taken over 100 images. Variances between times are too small to measure so we do not include error bars.

bers of stored images, extracting heading information using the Perfect Memory is faster. However, assuming that training images are sampled every 100 ms, Infomax begins to be more efficient for routes with only a little over 1 min of training data making it much more feasible approach for long-range visual homing.

3.3 Autonomous robot

Using the implementations of Infomax and the Perfect Memory algorithms from our Brains-on-Board Robotics library (Dewar et al., 2017), we built a simple application which can recapitulate learned routes on the robot described in Section 2.3 by running the following simple algorithm every 500 ms (based on the performance for 1000 images

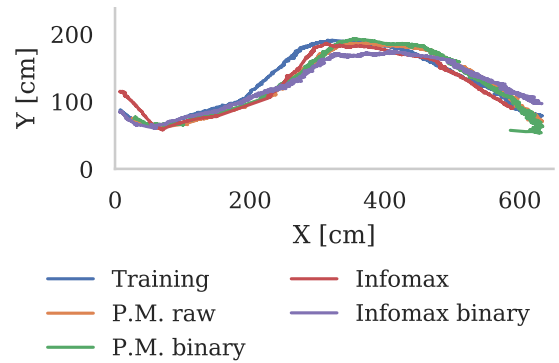


Figure 9: Reconstructed paths of autonomous robot during training and testing using each algorithm.

established in the previous section):

1. Capture and unwrap a panoramic image.
2. Perform one of the image processing steps described in Section 2.4.
3. Using either the Infomax or Perfect Memory algorithm, calculate the familiarity with the processed image in-silico when rotated through $\pm 90^\circ$.
4. Find the orientation with the highest familiarity and, if it is within 4° , start driving forwards. Otherwise, start turning in the correct direction to align with the image.

In order to compare the performance of the navigation algorithms and image processing steps described in Sections 2.1, 2.2 and 2.4 running on the robot, we first manually drove the robot along a sinuous route through the wooded

area shown in Figure 1b, recording training images every 100 ms. We then trained each of the navigation algorithms on the resultant dataset of 455 images and allowed the robot to recapitulate the path using the procedure described above. Throughout this training and testing process, we used the method described in Section 2.4 to track the robot, resulting in the data shown in Figure 9. The robot was able to successfully recapitulate the training path using each of the navigation and image processing algorithms with little difference in performance immediately apparent in Figure 9. We confirmed this by calculating the shortest distance to the training path at each location along each recapitulated path and found that, in this environment, the different algorithms performed very similarly (differences were within margin of error for extracting robot location from video) with mean distance between the training and recapitulated paths of 9 cm and standard deviation of 8 cm.

4 Summary and Discussion

In this paper we present the first familiarity-based navigation algorithm that can function effectively in the real world on board a small mobile robot using an ANN-based route encoding. Indeed, we show that in some cases, using an ANN can improve performance as it imposes a level of generalisation meaning that particularly tortuous elements of the training route (which we presume to be from a noisy training process) are smoothed out in the recapitulated trajectories. While our formulation of the Infomax learning rule is non-local and therefore not biologically plausible, we have shown the potential for biomimetic algorithms to efficiently control small, fully autonomous robots. Additionally, Infomax implements a form of decorrelation, a process which has been attributed to insect mushroom bodies (Hige et al., 2015; Nowotny et al., 2005). Furthermore, local variants of Infomax have been demonstrated within recurrent spiking neural networks (Hayakawa et al., 2014) and future work will explore the use of such, more biologically plausible learning rules for navigation.

In addition, we reinforce results showing that removing the sky and using only a binary image make these algorithms more robust to variability in lighting and weather conditions (Philippides et al., 2011; Stone et al., 2014). Furthermore, these improvements are seen even if the segmentation is performed automatically on board the robot using an RGB image processed using a simple watershed algorithm rather than requiring specialist sensors such as a UV camera. Interestingly, these improvements are not noticeable in the results collected using the autonomous robot. We believe that this is due to the fact that we performed these experiments in a more enclosed environment, where the sky covered a smaller portion of each image and because the testing and training both occurred at the same time of day. These two factors both act to reduce the differences between the raw images meaning that raw image comparisons are more successful.

Finally, we show that the computation and storage needed to recapitulate a route using the Infomax ANN does not increase with the number of training views. To put this into the context of robotic control algorithms, this is not typically a property of SLAM based localization systems where more keyframes are accumulated over time although there has been recent work to cap (Maddern et al., 2012) or at least cull (Mur-Artal et al., 2015) the number of keyframes accumulated. While visual SLAM implementations based around SURF or SIFT can take several hundred milliseconds to extract features from each frame (Bay et al., 2006), recent SLAM implementations such as FLaME (Greene and Roy, 2017) have been demonstrated running on autonomous quadrotors at much higher framerates than our current Infomax implementation can achieve on the Jetson TX1. However, not only was FLaME implemented on an Intel CPU which Biddulph et al. (2018) found to be $5\times$ faster than a Jetson TX1, but the performance of our Infomax algorithm could be significantly improved. View-based algorithms in general seem to work well with low-resolution wide-field views (Wystrach et al., 2016) and this matches the observation that high-resolution vision in ants is associated with hunting not with long distance navigation. Therefore, while 120×25 pixel images were used throughout the work presented in this paper, Baddeley et al. (2012) first demonstrated Infomax for visual navigation using input images with around half this number of pixels (90×17). Because Equation 5 can be implemented as a matrix-vector product, the computational complexity of which scales quadratically with the number of pixels, using input images with half the number of pixels would reduce the time taken to evaluate Equation 5 by 75%. Furthermore, while our Infomax implementation uses OpenMP (Dagum and Menon, 1998) to take advantage of the Jetson TX1's four CPU cores, it does not utilise its 256 core GPU. Initial experiments using the cuBLAS (NVIDIA Corporation, 2007) GPU-accelerated linear algebra library suggest that Equation 5 could be evaluated in around 100 ms for 120×25 pixel images – a $5\times$ speedup over our current implementation. Thus, as with SLAM which has seen vast increases in performance efficiency, future work on the ANN encoding, visual processing and behavioural strategy, will extend the range of homing and reduce computation and training times further.

5 Acknowledgements

This work was funded by the EPSRC (Brains on Board project, grant number EP/P006094/1).

References

- Amari, S.-i. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276.
- Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., and Webb, B. (2016). Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments. *PLoS Computational Biology*, 12(2):1–22.

- Baddeley, B., Graham, P., Husbands, P., and Philippides, A. (2012). A model of ant route navigation driven by scene familiarity. *PLoS Computational Biology*, 8(1).
- Baddeley, B., Graham, P., Philippides, A., and Husbands, P. (2011a). Holistic visual encoding of ant-like routes: Navigation without waypoints. *Adaptive Behavior*, 19(1):3–15.
- Baddeley, B., Graham, P., Philippides, A., and Husbands, P. (2011b). Models of visually guided routes in ants: Embodiment simplifies route acquisition. In *International Conference on Intelligent Robotics and Applications*, pages 75–84. Springer.
- Bay, H., Tuytelaars, T., and Gool, L. V. (2006). SURF: Speeded Up Robust Features - Demonstration. *Computer Vision ECCV*, pages 404–417.
- Bell, A. J. and Sejnowski, T. J. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159.
- Beucher, S. (1979). Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing*.
- Biddulph, A., Houlston, T., Mendes, A., and Chalup, S. K. (2018). Comparing Computing Platforms for Deep Learning on a Humanoid Robot. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11307 LNCS:120–131.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobbs's Journal of Software Tools*.
- Dagum, L. and Menon, R. (1998). OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55.
- Dewar, A., Knight, J. C., Domcsek, N., and Cope, A. J. (2017). BoB robotics. https://github.com/BrainsOnBoard/bob_robotics. Last accessed on 07/03/2019.
- Domcsek, N., Knight, J., and Nowotny, T. (2018). Autonomous robot navigation using GPU enhanced neural networks. In *Journal of Robotics and Autonomous Systems*, volume 1, pages 77–79, Bristol.
- Graham, P. and Philippides, A. (2014). Insect-inspired visual systems and visually guided behavior. *Encyclopedia of Nanotechnology*, pages 1–9.
- Greene, W. N. and Roy, N. (2017). FLAME: Fast Lightweight Mesh Estimation Using Variational Smoothing on Delaunay Graphs. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:4696–4704.
- Hayakawa, T., Kaneko, T., and Aoyagi, T. (2014). A biologically plausible learning rule for the Infomax on recurrent neural networks. *Frontiers in Computational Neuroscience*, 8(November):1–19.
- Hige, T., Aso, Y., Rubin, G. M., and Turner, G. C. (2015). Plasticity-driven individualization of olfactory coding in mushroom body output neurons. *Nature*, 526(7572):258–262.
- JK Imaging Ltd. (2016). SP360 4K 360 Degree VR Camera. <https://kodakpixpro.com/cameras/360-vr/sp360-4k>. Last accessed on 05/03/2019.
- Knaden, M. and Graham, P. (2016). The sensory ecology of ant navigation: from natural environments to neural mechanisms. *Annual review of entomology*, 61:63–76.
- Lee, T.-w. and Sejnowski, T. J. (1997). Independent Component Analysis for Mixed Sub-Gaussian and Super-Gaussian Sources. *Joint Symposium on Neural Computation*, 441:6–13.
- Lukežič, A., Vojí, T., ČehovinZajc, L., Matas, J., and Kristan, M. (2018). Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *International Journal of Computer Vision*, 126(7):671–688.
- Lulham, A., Bogacz, R., Vogt, S., and Brown, M. W. (2011). An infomax algorithm can perform both familiarity discrimination and feature extraction in a single network. *Neural Computation*, 23(4):909–926.
- Maddern, W., Milford, M., and Wyeth, G. (2012). Capping computation time and storage requirements for appearance-based localization with CAT-SLAM. In *2012 IEEE International Conference on Robotics and Automation*, pages 822–827. IEEE.
- Mur-Artal, R., Montiel, J. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.
- Nowotny, T., Huerta, R., Abarbanel, H. D. I., and Rabinovich, M. I. (2005). Self-organization in the olfactory system: One shot odor recognition in insects. *Biological Cybernetics*, 93(6):436–446.
- NVIDIA Corporation (2007). cuBLAS. <https://developer.nvidia.com/cublas>. Last accessed on 05/03/2019.
- NVIDIA Corporation (2016). Jetson TX1. <https://developer.nvidia.com/embedded/buy/jetson-tx1>. Last accessed on 05/03/2019.
- Parallax Inc. (2012). Robot Shield with Arduino. <https://www.parallax.com/product/32335>. Last accessed on 05/03/2019.
- Philippides, A., Baddeley, B., Cheng, K., and Graham, P. (2011). How might ants use panoramic views for route navigation? *Journal of Experimental Biology*, 214(3):445–451.
- Philippides, A., Graham, P., Baddeley, B., and Husbands, P. (2015). Using Neural Networks to Understand the Information That Guides Behavior: A Case Study in Visual Navigation. In *Artificial Neural Networks*, pages 227–244.
- Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256.
- Stone, T., Mangan, M., Ardin, P., and Webb, B. (2014). Sky segmentation with ultraviolet images can be used for navigation. *Robotics: Science and Systems X*.
- Wehner, R. (2009). The architecture of the desert ants navigational toolkit (hymenoptera: Formicidae). *Myrmecol News*, 12(September):85–96.
- Wehner, R., Boyer, M., Loertscher, F., Sommer, S., and Menzi, U. (2006). Ant navigation: one-way routes rather than maps. *Current Biology*, 16(1):75–79.
- Wystrach, A., Dewar, A., Philippides, A., and Graham, P. (2016). How do field of view and resolution affect the information content of panoramic scenes for visual navigation? A computational investigation. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 202(2):87–95.
- Wystrach, A., Mangan, M., Philippides, A., and Graham, P. (2013). Snapshots in ants? new interpretations of paradigmatic experiments. *Journal of Experimental Biology*, 216(10):1766–1770.
- Wystrach, A., Philippides, A., Aurejac, A., Cheng, K., and Graham, P. (2014). Visual scanning behaviours and their role in the navigation of the Australian desert ant *Melophorus bagoti*. *Journal of Comparative Physiology A*, 200(7):615–626.
- Zeil, J., Hofmann, M. I., and Chahl, J. S. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *JOSA A*, 20(3):450–469.

Hybrid Variational Predictive Coding as a Bridge between Human and Artificial Cognition

André Ofner¹ and Sebastian Stober²

¹ University of Potsdam, Potsdam, Germany

² Otto von Guericke University, Magdeburg, Germany
stober@ovgu.de

Abstract

Predictive coding and its generalization to active inference offer a unified theory of brain function. The underlying predictive processing paradigm has gained significant attention in artificial intelligence research for its representation learning and predictive capacity. Here, we suggest that it is possible to integrate human and artificial generative models with a predictive coding network that processes sensations simultaneously with the signature of predictive coding found in human neuroimaging data. We propose a recurrent hierarchical predictive coding model that predicts low-dimensional representations of stimuli, electroencephalogram and physiological signals with variational inference. We suggest that in a shared environment, such hybrid predictive coding networks learn to incorporate the human predictive model in order to reduce prediction error. We evaluate the model on a publicly available EEG dataset of subjects watching one-minute long video excerpts. Our initial results indicate that the model can be trained to predict visual properties such as the amount, distance and motion of human subjects in videos.

Introduction

Predictive processing has been used to explain a large variety of phenomena in human cognition within neuroscience and psychology. The notion of predictive coding refers to the idea that perception involves hierarchical predictive models with expectation error propagation (Friston and Kiebel, 2009). The more general framework of active inference suggests that perception and action exist in a closed loop, maintaining an agent’s generative model of the world (Adams et al., 2013). These ideas have recently found traction in machine learning (ML). ML has been used to classify, predict and learn shared embeddings of stimuli and brain activation (Du et al., 2017). We propose to interface human and artificial inference on the basis of predictive coding as a shared principle. A predictive coding based artificial neural network processes human neurophysiological data simultaneously with visual stimuli that are perceived by both human and machine. The processed neurophysiological signals reflect predictive coding based inference in the human brain. This means that the artificial model fuses predictions about changes in the shared environment and the corresponding physiological response from human inference. We suggest that this allows the network to subsume its own and human predictions in a joint generative model, in a process referred to as hybrid predictive coding (HPC). Here we focus on augmenting artificial predictive coding using electroencephalography (EEG). The suggested generative model learns to

predict compressed representations of multi-modal sensory states in the future by means of prediction error minimization. Deep convolutional neural networks and variational inference are used to parameterize the low-dimensional latent space at each time step.

Hybrid variational predictive coding

We introduce an architecture that processes stimuli, EEG and physiological signals by generating multiple views from a shared latent embedding z : $p(\text{stimulus}, \text{eeg}, z) = p(z)p(\text{stimulus}|z)p(\text{eeg}|z)$. EEG and physiological signal are treated as a single view, denoted with $p(\text{eeg})$. The distributions $p(z)$, $p(\text{stimulus} | \text{eeg})$, and $p(\text{eeg} | z)$ are set to be Gaussian. The expectations $E[z | \text{stimulus}]$ and $E[z | \text{eeg}]$ of the maximum likelihood solution exist within a shared space that maximizes their correlations. We use deep convolutional neural networks (CNNs) to parameterize the means of $p_{\Theta}(\text{eeg} | z)$ and $p_{\Theta}(\text{stimulus} | z)$ and the approximate posteriors $q_{\phi}(z | \text{eeg}, \text{stimulus})$.

Training with this shared embedding using variational inference can be done by sampling from $q_{\phi}(z | \text{eeg})$. We optimize the lower bound of the log likelihood $L(\text{eeg}, \text{stimulus}; \theta, \phi)$ with stochastic backpropagation by optimizing the sum of reconstruction losses and the Kullback-Leibler (KL) divergence between the learned $q_{\phi}(z | \text{eeg}, \text{stimulus})$ and $p(z)$. In order to process a total of n consecutive time-steps, we iteratively feed inputs into the encoders and compute a total reconstruction loss. For each time-step, an arbitrary selection of encoders can be active. Decoding from the latent space however is always executed for all modalities. The inputs of the first step are directly used to compute the latent embedding. For time-steps 2 to n , a hierarchy of predictive coding layers process the latent embeddings of previous time-steps and predict the current embedding. This module extends the hierarchical convolutional predictive coding network introduced by Lotter et al. (PredNet) to multimodal processing and variational inference (Lotter et al., 2016).

Each layer l of the predictive coding module features recurrent convolutional network units R^l that are used to compute predictions \hat{A}^l for each layer. These predictions are compared with a target for the corresponding layer A^l . For the lowest layer, the targets are approximate posteriors $q_{\phi}(z | \text{eeg}, \text{stimulus})$. For higher layers, the targets are the error E^l between A^l and \hat{A}^l . The recurrent representation units R^l receive information about the error E^l of their layer as well as top-down feedback from the representation

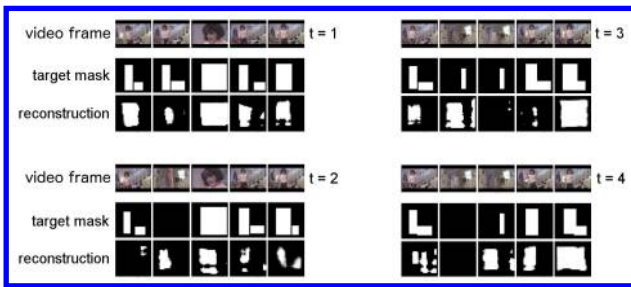


Figure 1: Predicted segmentation masks indicating the position of human subjects within presented videos. Each of the 5 presented examples corresponds to 4 sequential inference steps. Masks for step 1 are reconstructed with target masks available. Step 2-4 are predicted using only EEG and physiological signal.

units in the next higher level of the network R^{l+1} . The error units and the layer-wise predictions are computed with CNNs and the recurrent representations are convolutional LSTMs. We iteratively feed the latent embeddings of time-steps 1 to $n-1$ as inputs and use the resulting predictions \hat{A}^l of the lowest predictive coding layer for variational inference for time-steps 2– n . Their embeddings are replaced with the predicted counterparts. This way, the model encodes the inputs into representations that minimize the surprise for future steps. We suggest that this forces the network to learn temporal representations of the human physiology and brain signals that are congruent with the model’s own perception.

We refer to this approach of learning a shared generative model that aims at integrating the model’s own predictions and the human generative model as hybrid predictive coding. We suggest that predictive coding of (neuro-)physiological signal resembles interoceptive predictive coding, i.e. inference on internal states of the body, which seems to play a crucial role for human cognitive capacity (Seth et al., 2012).

Predicting with hybrid representations

We used the publicly available DEAP dataset to evaluate the model for the ability to predict future states (Koelstra et al., 2012). 32 channel EEG was recorded of 22 subjects while watching 40 one-minute long excerpts of music videos as well as the presented visual stimuli were provided as inputs to the model. Electrooculography (EOG) and electromyography (EMG) signals were recorded. The electrodes were mounted around the eyes, mouth and the shoulder blades. EEG and physiological signals were split into segments of 1 sec duration. The first frame of each second of video was extracted. We used a pre-trained image segmentation network to replace each video frame with a segmentation mask marking human subjects if present. This reduces the complexity of visual input, but the EEG signal still refers to the complex stimuli. The data for each subject was split by video identity. The test set contained only previously unseen stimuli.

We iteratively fed 4 consecutive seconds of EEG and physiological data to the HPC encoders. The preprocessed visual stimulus was only presented for the first step, i.e. steps 2-4 used only EEG and physiological inputs. The loss was computed as the sum of the reconstruction losses and the KL divergence for each step.

The network tended to predict the existence of human subjects more frequently than annotated using the segmentation network. Interestingly, many of these predictions were wrongly annotated by the segmentation network but still correctly interpolated by the HPC network. In longer scenes without visible human subjects, the HPC network tended to predict many false positives with large fluctuation between frames. If one or multiple humans were visible, the HPC predictions tended to be more sparse in comparison. Upon visual inspection, HPC seemed to improve the quality of its predictions within the 4 time-steps and often chose to not rely on visually guided interpolation. Examples for reconstructions within a single subject are shown in Figure 1). As there is no way for the model to infer whether a subject will move or appear/disappear into the frame, these results indicate that the network learns to guide visual predictions with information from the brain and body. Information about the initial distance and size of an object could be inferred from the given video frame or from its physiological representation. For future frames however, no visual input is provided. Change in amount, distance or motion in the environment has to be inferred from the physiological representation.

Conclusion

We proposed a hybrid variational predictive coding architecture that interfaces artificial and human predictive coding. HPC performs predictive coding based inference about a shared visual environment, human physiology and brain signal. The same stimuli are perceived by human and machine, allowing the model’s predictions to be modulated by human inference. Our initial results using an EEG dataset suggest that such a model can be used to predict aspects of the visual content of future frames of videos, such as the movement of human subjects.

Acknowledgments

This research is funded by the Federal Ministry of Education and Research of Germany (BMBF).

References

- Adams, R. A., Shipp, S., and Friston, K. J. (2013). Predictions not commands: active inference in the motor system. *Brain Structure and Function*, 218(3):611–643.
- Du, C., Du, C., and He, H. (2017). Sharing deep generative representation for perceived image reconstruction from human brain activity. *arXiv preprint arXiv:1704.07575*.
- Friston, K. and Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1521):1211–1221.
- Koelstra, S., Muhl, C., Soleymani, M., Lee, J.-S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., and Patras, I. (2012). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31.
- Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.
- Seth, A. K., Suzuki, K., and Critchley, H. D. (2012). An interoceptive predictive coding model of conscious presence. *Frontiers in psychology*, 2:395.

Sensorimotor contingency modulates visual awareness of virtual 3D objects

Keisuke Suzuki^{1,2}, David J. Schwartzman^{1,2}, Rafael Augusto³ and Anil K. Seth^{1,2}

¹Sackler Centre for Consciousness Science, University of Sussex, Brighton BN1 9QJ, United Kingdom

²Department of Informatics, University of Sussex, Brighton BN1 9QJ, United Kingdom

³Universidade de Lisboa, Alameda da Universidade, Lisboa 1649-004, Portugal

K.Suzuki@sussex.ac.uk

Abstract

To investigate how embodied sensorimotor interactions shape subjective visual experience, we developed a novel naturalistic Virtual Reality setting combined with motion tracking that allow object interactions with a high degree of freedom, which we implemented within an adapted breaking continuous flash suppression (bCFS) paradigm. This setup allowed us to manipulate the sensorimotor contingencies governing interactions with virtual objects, while characterising the effects on subjective visual experience by measuring breakthrough time to awareness of the virtual objects. We found that breakthrough times were faster for live compared to replayed sensorimotor interactions, demonstrating that visual awareness for unfamiliar 3D virtual objects is influenced by the contingency of the dynamic causal coupling between a person's actions and their visual consequences, in line with theories of perception that emphasise the influence of sensorimotor contingencies on visual experience.

Introduction

The brain engages with the environment through the body, instantiating a closed loop between perception and action. The relevance of these interactions for perceptual phenomenology is emphasized by the Sensorimotor Theory of Consciousness (STC) (Noë, 2004; O'Regan & Noë, 2001). In STC, perceptual phenomenology is shaped by "mastery" of the sensorimotor contingencies governing how sensory signals respond to actions. STC has been applied, conceptually, to many aspects of perceptual phenomenology. One prominent application has been to the phenomenology of "objecthood" in vision. In STC, objecthood depends on the brain's encoding knowledge about how afferent visual signals change given motor actions, such as eye movements. For example, when I experience the coffee cup in front of me as a three-dimensional object with a back-and-sides, it is because my brain "knows about" the sensory consequences of moving my eyes, or rotating the mug. In this sense, I perceive that the mug has a back even though I cannot directly see it (Noë, 2004; O'Regan & Noë, 2001; Seth, 2014). However, empirical tests of the influence of sensorimotor contingencies on perception using realistic or real objects have yet to be achieved, mainly due to the technical challenges of real-time manipulation of sensorimotor contingencies in such contexts.

Here we address this challenge by leveraging recent developments in virtual reality that allow flexible manipulations of the sensorimotor coupling of morphologically complex virtual 3D objects. To investigate

how sensorimotor interactions shape subjective visual experience, we combined Virtual Reality technologies with a variant of the binocular rivalry paradigm, known as breaking continuous flash suppression (bCFS) (Stein, Hebart, & Sterzer, 2011; Tsuchiya & Koch, 2005). During bCFS, perception of a target stimulus presented to one eye is suppressed by a series of rapidly changing, high contrast, Mondrian patterns presented to the other eye, and the time it takes the target to 'break through' into awareness is measured.

Experiment

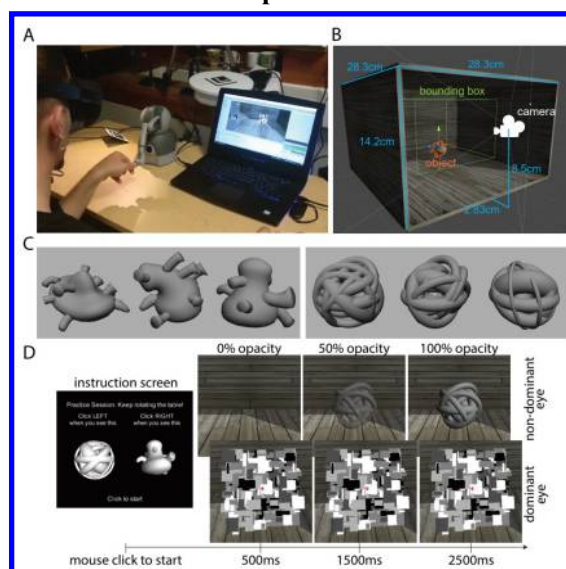


Figure 1: **A.** Experimental setup. **B.** Overview of the virtual environment. **C.** The six virtual objects used in this experiment, divided into two categories. **D.** The objects opacity gradually increased over time and was presented to the non-dominant eye (Top) and the Mondrian mask was presented to the dominant eye (Bottom).

Participants were asked to identify if an object belonged to one of two categories (Figure 1C) presented using the bCFS paradigm, while performing natural unrestricted rotational movements of the object using a stylus attached to a motion-

tracking device (Figure 1A). The stylus motion was transferred to the virtual object (Figure 1B). Virtual objects were presented through a head-mounted display to the non-dominant eye, while a dynamic Mondrian mask was presented to the dominant eye (Figure 1D). Participants were required to identify the object, using a mouse-click as quickly as possible.

We compared three aspects of sensorimotor coupling on breakthrough times. In the 'Live' condition, the object responded directly to the participant's rotational movements. In the 'Replay' condition, the object rotated according to the rotational movements of a randomly selected pre-recorded practice trial. To provide a baseline for breakthrough times we added a 'Static' condition, in which the 3D virtual object always maintained the same orientation (apparent visual angle) with respect to the participant. See our published work for more details about the experimental setup and procedure (Suzuki, Schwartzman, Augusto, & Seth, 2019).

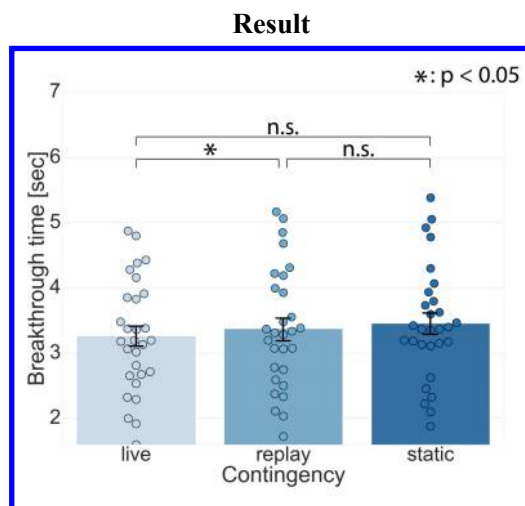


Figure 2: Breakthrough time by sensorimotor coupling type (Live, Replay, Static). Significantly shorter breakthrough times were found for the Live compared to the Replay condition. Dots show individual participant results.

31 participants completed the experiment. 2 participants' data were excluded due to extreme differences in reported and physical rotation speeds. The remaining data were submitted to a 1-way repeated measures ANOVA consisting of the main factors Live/Replay/Static. The results revealed a significant main effect ($F(56,2) = 4.125$, $p=0.021$, $\eta^2 = 0.128$) for breakthrough time, with Live ($M = 3.38$ s, $SE = 0.18$ s) showing shorter breakthrough times compared to Replay ($M = 3.58$ s, $SE = 0.21$ s) or Static coupling ($M = 3.67$ s, $SE = 0.21$ s) (Figure 2). Post-hoc t -tests, with Holm–Bonferroni correction, revealed a significant difference in breakthrough time only between Live and Replay conditions ($t(28) = -2.905$, $p_{\text{holm}} = 0.021$) (Live/Static: $t(28) = -2.354$, $p_{\text{holm}} = 0.052$; Replay/Static: $t(28) = 0.732$, $p_{\text{holm}} = 0.470$). These analyses indicate that breakthrough into awareness of virtual objects identity was faster for interactions that were directly coupled

to (i.e., contingent upon) participants' movements, compared to Replay or Static trials.

Discussion

We developed a novel combination of Virtual Reality and motion tracking technologies within an adapted breaking continuous flash suppression (bcFS) to explore how manipulations of sensorimotor contingencies affect visual awareness of realistic 3D virtual objects. Our result shows that the sensorimotor contingency of a person's actions and their visual consequences influences access to visual awareness, in line with theories of conscious perception that emphasise embodied sensorimotor interactions.

The effects of sensorimotor contingency on perception has also been investigated using autonomous robotics (Braitenberg, 1986; Buhmann, Di Paolo, & Barandiaran, 2013). However, empirical studies that attempt to study sensorimotor contingencies in humans and artificial systems remains sparse. Our study provides a novel setup that could be employed to investigate sensorimotor interactions in both human and artificial systems, with the results of both open to interpretation using the theoretical framework of Sensorimotor Theory.

Acknowledgements. This work has been supported by the Dr. Mortimer and Theresa Sackler Foundation, which supports the Sackler Centre for Consciousness Science.

References

- Braitenberg, V. (1986). *Vehicles : experiments in synthetic psychology*. Cambridge, MA: MIT Press.
- Buhmann, T., Di Paolo, E. A., & Barandiaran, X. (2013). A dynamical systems account of sensorimotor contingencies. *Frontiers in Psychology*, 4(May), 285.
- Noë, A. (2004). *Action in Perception*. MIT Press.
- O'Regan, J. K., & Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *The Behavioral and Brain Sciences*, 24(5), 939-73; discussion 973-1031.
- Seth, A. K. (2014). A predictive processing theory of sensorimotor contingencies: Explaining the puzzle of perceptual presence and its absence in synesthesia. *Cognitive Neuroscience*, 1–49.
- Stein, T., Hebart, M. N., & Sterzer, P. (2011). Breaking Continuous Flash Suppression: A New Measure of Unconscious Processing during Interocular Suppression? *Frontiers in Human Neuroscience*, 5, 167.
- Suzuki, K., Schwartzman, D. J., Augusto, R., & Seth, A. K. (2019). Sensorimotor contingency modulates breakthrough of virtual 3D objects during a breaking continuous flash suppression paradigm. *Cognition*, 187, 95–107.
- Tsuchiya, N., & Koch, C. (2005). Continuous flash suppression reduces negative afterimages. *Nature Neuroscience*, 8(8), 1096–1101.

Heuristics as Decision-making Habits of Autonomous Sensorimotor Agents

Erasmus Batta^{1,2} and Christopher Stephens^{2,3}

¹Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico City, MX 04510

²Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Mexico City, MX 04510

³Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, Mexico City, MX 04510

jebatta@c3.unam.mx

Abstract

Human judgement is better described as a heuristic process rather than maximisation of a utility function. Heuristics are high cognitive processes of decision-making whose rapid and effortless implementation is useful to confront risk scenarios that compromise the viability of an individual. They can be defined in an enactive frame as self-sustained and self-generated habits of abductive behaviour selection in sensorimotor agents influenced by the individual history of sensorimotor contingencies and the environment. In this work, we analyse the emergence of patterns of behaviour and its necessary ecological conditions when performed decisions are related to energy intake and energy expenditure. Agent's sensors and intentions are coupled to a variation of the iterant deformable sensorimotor medium (IDSM) (Egbert and Barandiaran, 2014). This model explains transparently the generation of sensorimotor habits in simulated robots through the influence of registers of previously executed behaviours reinforced by repetition. We create a decision-making frame based on intentions as probabilities of specific actions which constitute the motor component of sensorimotor states on IDSM. In this model is seen that specific behaviour correlations with the lifespan of agents depend on the availability of energetic sources on the environment. Inheritance of the medium is introduced in agents with a small improvement on the lifespan of agents.

Introduction

Human decision making is a complex process that goes beyond the maximisation of probabilistic utility functions. The traditional "rational" judgement method is intractable in practice due to the amount of information and cognitive capacities that would be required in natural scenarios. Many species (humans beings among them) constantly face decisions that compromise their survival and somehow they remain. The apparently non-optimal decision making performed by organisms works fine at least in ecological terms. Alternative models of decision making were developed considering efficiency and immediacy as key features. Tversky and Kahneman (1974) first describe *cognitive bias* in human judgement, a decision-making process which simplify original problems substituting complex attributes by other computationally less demanding. The idea of *bias* evolved from

the original conception of a complementary intuitive process triggered by individual limitations to the concept of *heuristics*: self-consistent cognitive processes that can generate the best outputs when environmental conditions are coupled (Cosmides and Tooby, 1994; Reyna, 2004; Gigerenzer and Gaissmaier, 2011). Human heuristics arise with a minimum effort and are applied immediately due to their accessibility. These features make them valuable in decisions under risk or uncertainty. Despite the explanatory power of heuristics, their formation has not been dealt in depth.

In an eco-cognitive perspective heuristics can be framed as manipulative abduction processes (Magnani, 2015). Manipulative abduction happens "... when we are thinking through doing and no only, in a pragmatic sense, about doing ... Manipulative abduction refers to an extra-theoretical behaviour that aims at creating communicable accounts of new experiences to integrate them into previously existing systems of experimental and linguistics (theoretical) practices" (Magnani, 2009). Action in heuristics can provide information unavailable by orthodox mechanisms based on the logical inference from axioms. Heuristics are selective or creative models *i.e.* select already conceived hypothesis or generates new ones. The epistemological value of heuristics has been questioned for its apparent ignorance-preserving nature despite its practical power to enhance knowledge empirically exploiting ecological regularities (Magnani, 2015). McCarthy (1981) recognises the utility of separation between epistemological and heuristic problems in artificial intelligence due to the lack of a formalisation of common sense in the classical paradigm of mind. The computational theory of mind explains cognition as the processing of representational states of the world as in a computational system (Cohen and Feigenbaum, 2014). Under this approach, any decision-making process besides utility maximisation is not supposed to produce optimal nor sustainable outputs, but that is actually what occurs. The strict classification on "rational decisions" and "remaining processes" makes the second category ambiguous and prohibits a reliable characterisation of truly intentional, ecologically-successful decision-making. What is missing to understand the nature of any

emergent cognitive process is an approach which takes in account embodiment and experience. The enactive view to cognition fulfils these conditions offering a paradigm with no *a priori* assumptions on the hierarchy of decision-making processes.

Enactivism is built based on the concepts of embodiment, autonomy, subjective experience and sense-making (Di Paolo et al., 2010). The enactive identity on a system comes from the operational closure of the network of self-produced and self-maintained processes. Those are coupled with the environment in a sensorimotor loop regulated by the system itself. Cognition happens when non-passive interaction loop is truly intentional, and this can be possible if the system lives in precarious conditions (Di Paolo et al., 2010; Stewart, 2010). The cognitive process has the world as its own model instead of symbolic representation. Then the system creates meaning when perceiving. This agrees with the immediate accessibility on intuitive reasoning making the enactive approach appealing to study the formation of heuristics.

Froese and Ziemke (2009) state two principles in the design of enactive artificial intelligence: (EAI-1) Agents must be capable of generating its own systemic identity at some level, and (EAI-2) must have the capacity to actively regulate its ongoing sensorimotor interaction in relation to a viability constraint. De Loor et al. (2009) suggest that the regulation of the sensorimotor interaction must modify the environment too, sometimes irreversibly. Is it possible to address heuristic behaviours within these principles describing them as self-maintaining precarious systems of judgement actively regulated by a sensorimotor loop which resemble habits of decision-making. The concept of habit has not a universally accepted definition but is possible to recognise two branches on the historical development of it: An explanation of habit as an automatic association and an organicist and ecological definition which links self-organising structures with the agent and the environment in a co-evolving relationship (Barandiaran and Di Paolo, 2014). Following the former trend, Egbert and Barandiaran (2014) developed an enactive model that represents habits of sensorimotor contingencies: the Interant Deformable Sensorimotor Medium (IDSM). This system has two main features: (1) Sensorimotor states are points of a continuous space representing a history of sensory inputs and motor activity, at a mesoscopic level (2) a weighted record of sensorimotor states associated with sensorimotor changes which influence agents future actions according to her present state. Habits appear as self-reinforcing repetitive patterns of behaviour in the sensorimotor space. Reinforcement depends on how frequently and/or recently a pattern of behaviour has been performed. IDSM has been implemented to represent living organisms with sensory inputs of metabolic variables with no specification of the life-sustaining role of this particular sensorimotor dimension (Egbert and Canamero, 2014). Such that sys-

tems are more likely to keep agents and sensorimotor habits into their respective viability region. This result suggests that cognitive and biological structures could be deeply integrated sharing essential variables.

We model agents as living organisms which take decisions with bounded cognitive capacities, limited information and an environment demanding rapid responses. The aim of this paper is to analyse the effect of history-based control on the willingness to perform an action. To determine whether is possible to generate such that influence on intentions we employ an variation of IDSM where the strict relation between sensory activation and motor action is substituted by a relation between the sensory activation and the probability to perform the motor action. To portray the effects of this change we use the formalised sensorimotor coupling between environment (e), agent bodily configuration (p), internal state (a) and sensory (s) and motor states (m) proposed by (Buhrmann et al., 2013) $\dot{e} = E(e, p)$, $\dot{s} = S(e, a)$, $\dot{a} = A(a, s)$, $\dot{m} = M(a)$, $\dot{p} = B(m, e)$ which permits the operational description of four types of sensorimotor contingencies (SMCs). Changing $\dot{m} = M(a)$ for an stochastic function weakens the lawful closure of the sensorimotor loop but the sensorimotor contingencies still can be formulated without a mayor problem as the relationship $S(e, a)$ remains unchanged. The main advantage of the redefinition of the motor dynamic is that it gives the system a physical indeterminacy which opens an space for the subjectivity of the simulated enactive agent (Froese and Taguchi, 2019). The decision model as described also can show an irreversible co-evolution of agent, heuristics (as habits) and environment (De Loor et al., 2009).

A decision-making habit tied to a decision problem involving the viability of the agent is likely to be integrated into the identity cycle of both, the agent and the habit (Egbert and Canamero, 2014). We model agents whose actions are related with essential variables and whose sensory inputs integrate the recognition of the proximity with the viability limits of essential variables. An organism with this characteristics can develop an adaptive stochastic behaviour-changing mechanism to keep essential variables on the viability region. This process of adaptation in the system agrees with a broader definition of an ultrastable system proposed by (Izquierdo et al., 2013) in which the separation between parameter-changing and behaviour-producing mechanisms is no longer required.

The decision-making problems modelled in this work are the energy intake and the execution of movement with an energetic expenditure, both related to long-term energy imbalance. The energy of the agent is evidently an essential variable: A living organism that approaches zero energy starts to experience a malfunction and eventually death. Some living systems accumulate energy surplus in their bodies as structural components or as reserves for periods of scarcity. Humans store energy excess as fat reserves which, if the surplus

is maintained, can cause an increase in general and specific-disease related mortality (Flegal et al., 2005, 2007). The asymmetrical conflict between drives that increase and reduce energy reservoirs suggests a natural tendency to store as much energy as possible. The dynamics of available energetic sources also play a role in shaping long term behaviour. For example, a constant strategy consisting in always attempt to eat can keep an agent into the viability region and can avoid the accumulation of excessive energy reserves in an environment with a physically or temporally sparse distribution of food. In this case, the success of such that decision making strategy relies on the high uncertainty of the achievement of the objectives. Agents face this decision process with limited information and a perceived signal of the internal energy in the form of unpleasant states that trigger a feeding action (stop or start eating). The limited available information comes from their embodiment and situatedness but is plausible to assume that at least part of this information is transmitted from ancestral organisms as a phenotype of behaviour or a cultural context. We study if the congenital trends of behaviour could be translated into an ecological advantage in particular environments. The main environmental feature studied in this work is the availability of energetic sources.

Agent model

Agents are embedded in a one-dimensional physical environment divided into cells with periodic boundary conditions. Every cell have one energetic source. If the source is eaten, the internal energy of the agent who ate it is increased by E_s and a new source can grow in the empty cell with a probability p_g .

Energy expenditure of agent α at time t is composed of a basal metabolic spending $M_b E_{max}(\alpha)$ plus a cost for movement $C_m E_\alpha(t)$ if movement is performed. The basal energy price, required for the maintenance of the vital functions of the agent is equal to the product of a constant M_b times the maximum level of energy ever reached by the agent $E_{max}(\alpha)$ at the present time. The change on energy at a unit of time is given by the following expression,

$$\Delta E_\alpha = E_s X_{eat} - M_b E_{max}(\alpha) - C_m E_\alpha(t) X_{move}, \quad (1)$$

where X_{eat} and X_{move} are random variables equal one when the agent eats and moves, respectively, and zero otherwise. When agent reaches zero energy she dies. Then, as the internal energy is the only essential variable, the viability region is defined by $E_\alpha > 0$.

Agents can eat the energetic source found in the cell, can move to one of the two neighbour cells (the agent randomly decides whether to move to the right or to the left), can perform both actions or none at every unit of time. The agent's IDSM controls the volition of every action. It is compound by *nodes* (not related with the concept of node

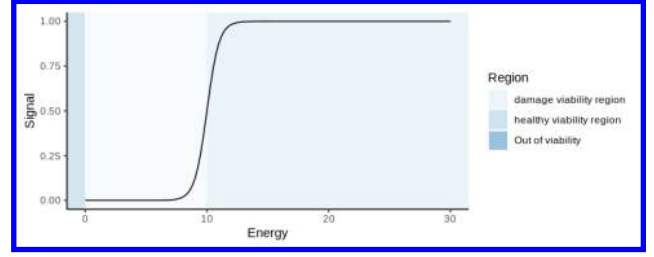


Figure 1: Signal of internal energy described by Eq.2 according the parameters on Table 1 and the three distinguishable regions of the essential variable E_α .

or vertex in graph theory) which are records of the combination of sensorimotor states and consequent changes on them. The medium maps a sensorimotor state to a motor output $f_t(S_t, M_t) \rightarrow M_{t+1}$ and the mapping itself change as a function $\frac{\Delta f}{\Delta t} = g(f, S, M, \frac{\Delta S}{\Delta t}, \frac{\Delta M}{\Delta t})$ so the trajectories bias the reinforcement of particular paths of sensorimotor activity (Egbert, 2018). In the system, the sensory state only depends on the internal energy E_α . $S(E_\alpha(t))$ is a sigmoidal function centred on a threshold value E_t , which represents a soft limit on energy that triggers the sensation of hunger.

$$S(E_\alpha) = \frac{1}{1 + \exp(A - BE_\alpha)}, \quad (2)$$

where A and B are such that $S(E_t) = 0.5$. The sensory signal lead every possible value of $E_\alpha(t)$ into the range $(0, 1)$ (Fig. 1). The concept of motor dimension was adapted to include the intention to perform an action. The *motor* state is represented by two independent probabilities to consume energy p_c and to move p_m instead of the actual actions of consuming and moving. I keep the term "motor" to refer the probabilities of action and the term "sensorimotor" to refer the complete sensory-intentional structure of agents' dynamics. Then the resulting sensorimotor space is three-dimensional with one sensory and two motor dimensions.

Nodes are tuples of two vectors and one scalar: N_p and N_ω ; the sensorimotor state and the sensorimotor velocity that an agent experience when the node is created, respectively, and N_ω ; the weighting of the node (which is zero when the node is created). By the construction of the sensorimotor space, every component of the state lies in $[0, 1]$.

$$\phi(x) = \sum_N \omega(N_\omega) \cdot d(N_p, x), \quad (3)$$

$$\omega(N_\omega) = \frac{2}{1 + \exp(-k_\omega N_\omega)}, \quad (4)$$

$$d(N_p, x) = \frac{2}{1 + \exp k_d \| (N_p - x) \|^2}. \quad (5)$$

New nodes are added to the medium if the node density $\phi(x)$ (eq. 3) is less than a threshold $k_t = 1$. This function

combines a factor of the weight of node (eq.4) and a factor related with the distance of the current state of the agent x and the state of a given node N_p (eq. 5). In these simulations the values of k_ω and k_d are 1000 and 0.0025 as in the original IDSM model (Egbert and Barandiaran, 2014). The weight of each node is updated according the following expression,

$$\frac{dN_\omega}{dt} = -k_{deg} + k_{rej} \cdot d(N_p, x), \quad (6)$$

whose parameters $k_{deg} = 1$, $k_{rej} = 10$ are related to the degradation of the weight of the node if the agent does not visit it frequently and with the rejuvenation of the weight when the agent is close. After a node is added to the medium it remains inactive during ten temporal units to avoid its effects to fully overshadows the rest of the nodes. The influence of IDSM in the motor state of the agent is:

$$\frac{d\mu}{dt} = \frac{1}{\phi^*(x)} \sum_{N^*} (\omega(N_w) \cdot d(N_p, x) \cdot (N_v + \Gamma(N_p - x, N_v))^\mu), \quad (7)$$

$$\Gamma(a, N_v) = a - \left(a \cdot \frac{N_v}{\|N_v\|} \right) \frac{N_v}{\|N_v\|}, \quad (8)$$

where $\phi^*(x)$ is the node density of the N^* active nodes and superindex μ means that only the motor components of the vector are taken. Medium net influence is built by the weighted effect of every active node in the agent's sensorimotor state. Node influence has two factors: One which throws agent away from node meanwhile a second one attracts it to the node. The motor states, the nodes and the medium (annexation of nodes) are updated continuously. As actions are performed (according the intention of the agent) every unit of time, the sensory signal changes in a discrete time. Then, during a unit of time, the sensorimotor states of the agent follow a path on a plane of motor dimensions, and every time step, the plane changes.

Simulations

To explore the creation of self-organised habits we create IDSM with nodes generated by random walks in the sensorimotor medium (Egbert and Barandiaran, 2014; Egbert and Canamero, 2014; Egbert, 2018). The medium is initialised with n_n random nodes generated by performing n_{RW} random walks in the sensorimotor space, starting from a random location and with subsequent loci calculated according to the following equation, $l_{i+1} = l_i + r$, where the components of r came from a uniform distribution $\mathcal{U}[-0.05, 0.05]$ (if l_{i+1} got out of range, it was inverted). At a locus l_i a node ($N_p = l_i, N_v = l_{i+1} - l_i, N_\omega = 0$) was added. This way of exploring the sensorimotor states does not necessarily generate a homogeneous distribution of nodes and may favour some possible sensorimotor habit. However, the assembly of agent systems allows recognising which of these

habits, if formed, are maintained for a longer time given certain environmental conditions. We add an amplifier factor C_{IDSM} which multiplies the influence of IDSM on intentions and thereto on actions to counteract the weakening of the sensory-action bond due to the stochasticity of the model. We compared two types of agents: The first type of agent uses a randomly initialised IDSM for behaviour control. The second type of agent inherits her IDSM to her descendants as if the behavioural history could be transmitted to offspring. The employed parameters are shown in Table (1).

Parameter	Symbol	Value(s)
Number of cells	N_c	10
Initial internal energy	$E_\alpha(t = 0)$	20
Energy threshold	E_t	10
Energy signal constants	A, B	20, 2
Metabolism rate	M_b	0.05
Cost of movement	C_m	0.01
Energy on cell sources	E_s	3
Regeneration probability	p_g	{0.7, 0.8, 0.9, 1.0}
Maximum time	t_{max}	500
Initial motor states	(p_c, p_m)	(0.5, 0.5)
Number of nodes	n_n	5000
Number of random walks	n_{RW}	100
Amplification of IDSM	C_{IDSM}	3
Repetitions		25
Type of agent		random IDSM, inherited IDSM

Table 1: Simulations parameters.

Sensorimotor behaviour is identified when changes on sensory inputs follows changes on motor states which follows changes on sensory states. If a set of behaviours is sustained and appear recurrently during the life-span of the agent it is considered a habit. We use the time an agent α survives (t_{ext}) and the time her internal energy E_α stays below the threshold E_t (t_{dam}) to classify healthy agents. Every action is preceded by an intention represented in the motor variables p_c and p_m . The effective attainment of the intention of eating depends on the availability of sources. Any other decision in this model (not to eat, move and not move) are always executed by the agents as they do not depend on any external factor. Eating behaviour can be categorised as homeostatic if is a coordinating response to an stimulus which disturbs or jeopardises the agent existence. If the internal energy decreases to the point where $E_\alpha(t) < E_t$ the sensory signal changes drastically. If this trend continues the agent can rapidly leave the viability region ($E_\alpha > 0$) then eating is an homeostatic reaction. As the system is built, the agent can increase her willingness to eat but eating still

depends on the availability of sources in the environment.

Results and discussion

On the 100 experiments in which agents always start with an untrained medium, the average time the agent survives is 188 and the mean proportion of time an agent stay below the threshold E_t (close to the viability limit) is greater than a sixth. Only in one case the agent is always above the energy threshold. In 6 scenarios the agents survived 500 temporal units. In this experiments less than a half of the times the internal energy of an agent is below E_t , she eats. This rate is higher than that of the complete set of agents. We can label this eating behaviour as homeostatic because it pushes the agent state into the viability region, away from the boundary. Homeostatic eating on agents depends on the perception of an state of hunger (or any sensory input of a low level of energy), the decision of eating and the availability of sources on cell. The rate of homeostatic eating (number of homeostatic eating events over number of eating events) by probability of regeneration of sources, agent by agent or aggregated, does not increase linearly with p_g . The rate of homeostatic eating and the rate of eating (number of eating events over the lifespan of the agent) are moderately correlated with a longer lifespan (0.51 and 0.63 respectively). Survivors move some more frequently than all the agents but this increase does not seem to be directly related with a change on the lifespan of agents (the linear correlation is 0.1). The effect of the availability of energetic sources, captured in the parameter p_g , on the strength of the linear correlation between rates of specific behaviour and the length of agent's life is summarised on Table 2. All this behaviours

p_g	ER vs LS	HER vs LS	MR vs LS
0.7	0.71	0.61	0.24
0.8	0.58	0.42	0.21
0.9	0.69	0.40	-0.08
1.0	0.65	0.57	0.09
All	0.63	0.51	0.10

Table 2: Linear correlation coefficients of eating rate (ER), homeostatic eating rate (HER) and moving rate (MR) against lifespan (LS) in agents grouped by p_g of the system.

appear to be more linked with an increase of lifespan for the lowest availability of energetic sources. Movement's weak direct effect on survival of agents disappears for high values of p_g . This is expected as the agent's chances to find food in her current cell increases enough to make the search for food unnecessary. Moving does not correspond to a rational mechanism for seeking food or expenditure of energy but the increase of dimensions on the registered memories extends the possible ways to create habits. The direct relationship between the eating rate and lifespan is moderate in

every environmental scenario. On the other hand, the homeostatic eating behaviour is moderately influential in systems with a complete availability of sources and systems with a relatively low availability.

Fig 2 shows the sensorimotor dynamics of surviving agents. The variations of motor variables p_c and p_m follow far larger changes on the signal, and in most of the cases they have the opposite direction of $\Delta S(E_\alpha)$. Some experiments keep the values of the motor dimensions (intention of eating and moving) close and apparently tied. The behavioural dynamic exhibited by surviving agents has not a cycle of states which could explain the fitness in the six scenarios but it shows that the transition states visited when the agent's energy is around the threshold E_t generate a reaction in the motor dimensions (intentions). The distribution of the possible combined actions (eat and move, eat but not move, move but not eat, not move nor eat) is almost uniform for the complete set of agents. For survivor agents, the simultaneous execution of eating and moving increases to the detriment of static strategies. We are not imposing any *a priori* utility directed behaviour in agents so the pattern of intentions and actions of surviving agents come directly from the interaction with the environment, the original cognitive setting and the reinforcement of sensorimotor memories. The aggregated absolute influences of the medium through time in p_c and p_m are not correlated for the complete set of experiments but if we only consider surviving agents a weak correlation appears (0.308) which points to a fragile coordinated influence.

When agents transmit to their descendants their sensorimotor history they are not spreading a utility function but a resume of contingencies to be interpreted by the newborn agents in their own terms. Agents with an inherited sensorimotor history have a longer average lifespan than the first set of agents (200.37 temporal units) but less of them survive 500 generations. This result suggests that these agents have a less diversified repertoire of habits to face environmental challenges, but their means of behaviour control, moulded by generations, increases their life expectancy. There is a significant change in the distribution of the time that agents survive ($p = 0.03757$ on a Mann-Whitney test) but no big difference in the distribution of the time spent below the energy threshold (Fig. 3). Agents with the inheritance of IDSM spent on average a little less time out of the viability region than their pairs without it. The distribution of combined actions does not change in comparison with agents with random IDSM. However, the improvement on lifespan appears at early generations and vanished until be imperceptible with the most distant descendants.

Conclusions

Heuristics as habits of decision-making have ideal characteristics for their study as enactive self-sustained and self-generated systems. We employed a model of enactive con-

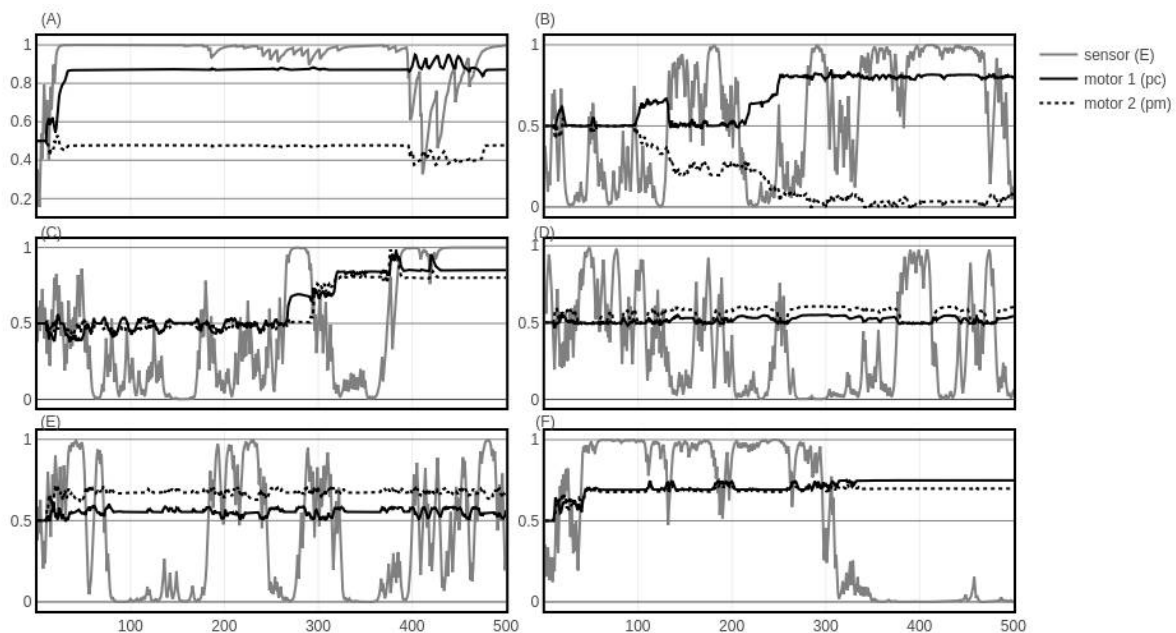


Figure 2: Sensorimotor dynamics of agents which survive 500 generations ordered inversely according to the time t_{dam} they spend below the energy threshold E_t . The variables shown are the energy signal $S(E_\alpha)$, the probability of consumption p_c and the probability of moving p_m . From left to right, top to bottom the value of the tuples (p_g, t_{out}) are: A.(1.0, 0), B.(0.7, 5), C.(0.8, 25), D.(1.0, 33), E.(1.0, 66) and F.(1.0, 119).

trol of sensorimotor events, the iterant deformable sensorimotor medium (IDSM), based on the repetition and reinforcement of previously visited states. We introduced the intention of performing actions as the motor dimensions on an iterant deformable sensorimotor medium and created an agent-based model of decisions related to energy intake and energy expenditure where agents decide stochastically if they eat sources on a cell of a periodic ring and/or if they move to one of their neighbouring cells. Those decisions are linked to an essential variable: the internal energy of the agent. The sensor dimension was built based on a sigmoidal function of the energy where the proximity with the viability limit is immediately recognised. Agents with the longest lifespan does not share a universal behavioural habit but is observed that homeostatic-eating and moving behaviour effects on fitness depends on the availability of energy on the environment. Inheritance of the iterant deformable sensorimotor medium seems to improve the outputs of the offspring but reduces the possibility to have very successful agents. This work is a first step towards an understanding of the creation of high cognitive decision-making mechanisms observed in organisms with subjective meaning. The enactive dynamical approach provides the minimal conditions for a transparent definition of heuristics as something more than an alternative mechanism. Further work needs to be carried out to endogenize some features of IDSM, as for example,

the sigmoidal sensory signal of the internal energy which could be replaced by a linear signal subject to evolve over time, and to formalise the stochastic functions on the sensorimotor coupling.

Acknowledgements

Erasmus Batta's contributions to this research were funded by the National Council for Science and Technology (CONACYT Mexico) [grant No.CVU 310372]. The authors would also like to thank Tom Froese for discussions of the research presented above.

References

- Barandiaran, X. E. and Di Paolo, E. A. (2014). A genealogical map of the concept of habit. *Frontiers in human neuroscience*, 8:522.
- Buhrmann, T., Di Paolo, E. A., and Barandiaran, X. (2013). A dynamical systems account of sensorimotor contingencies. *Frontiers in psychology*, 4:285.
- Cohen, P. R. and Feigenbaum, E. A. (2014). *The handbook of artificial intelligence*, volume 3. Butterworth-Heinemann.
- Cosmides, L. and Tooby, J. (1994). Better than rational: Evolutionary psychology and the invisible hand. *The American Economic Review*, 84(2):327–332.

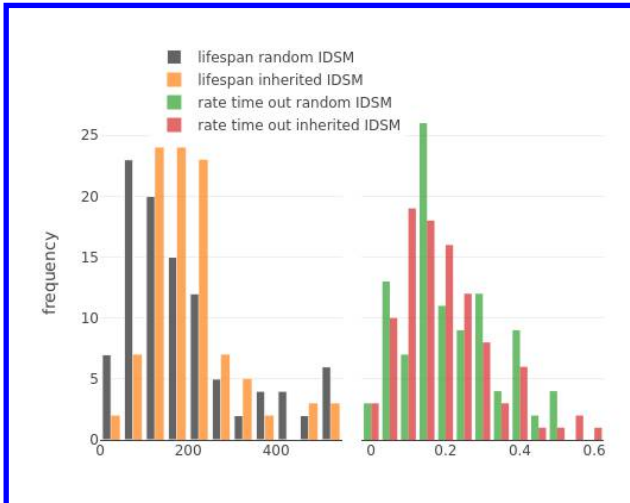


Figure 3: Comparison between histograms of the lifespan of agents and the rate of time out of the viability region for agents which start with a random IDSMS and agents which inherited IDSMS from ancestors.

De Loor, P., Manach, K., and Tisseau, J. (2009). Enaction-based artificial intelligence: Toward co-evolution with humans in the loop. *Minds and Machines*, 19(3):319–343.

Di Paolo, E., Rohde, M., and De Jaegher, H. (2010). Horizons for the enactive mind: Values, social interaction, and play. In *Enaction: Towards a new paradigm for cognitive science*.

Egbert, M. (2018). Investigations of an adaptive and autonomous sensorimotor individual. In *Artificial Life Conference Proceedings*, pages 343–350. MIT Press.

Egbert, M. and Canamero, L. (2014). Habit-based regulation of essential variables. In *Artificial Life Conference Proceedings 14*, pages 168–175. MIT Press.

Egbert, M. D. and Barandiaran, X. E. (2014). Modeling habits as self-sustaining patterns of sensorimotor behavior. *Frontiers in human neuroscience*, 8:590.

Flegal, K. M., Graubard, B. I., Williamson, D. F., and Gail, M. H. (2005). Excess deaths associated with underweight, overweight, and obesity. *Jama*, 293(15):1861–1867.

Flegal, K. M., Graubard, B. I., Williamson, D. F., and Gail, M. H. (2007). Cause-specific excess deaths associated with underweight, overweight, and obesity. *Jama*, 298(17):2028–2037.

Froese, T. and Taguchi, S. (2019). The problem of meaning in ai and robotics: Still with us after all these years. *Philosophies*, 4(2):14.

Froese, T. and Ziemke, T. (2009). Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence*, 173(3-4):466–500.

Gigerenzer, G. and Gaissmaier, W. (2011). Heuristic decision making. *Annual review of psychology*, 62:451–482.

Izquierdo, E., Aguilera, M., and Beer, R. (2013). Analysis of ultrastability in small dynamical recurrent neural networks. In *Artificial Life Conference Proceedings 13*, pages 51–58. MIT Press.

Magnani, L. (2009). *Abductive cognition: The epistemological and eco-cognitive dimensions of hypothetical reasoning*, volume 3. Springer Science & Business Media.

Magnani, L. (2015). Are heuristics knowledge-enhancing? abduction, models, and fictions in science. In *Heuristic reasoning*, pages 29–56. Springer.

McCarthy, J. (1981). Epistemological problems of artificial intelligence. In *Readings in artificial intelligence*, pages 459–465. Elsevier.

Reyna, V. F. (2004). How people make decisions that involve risk: A dual-processes approach. *Current directions in psychological science*, 13(2):60–66.

Stewart, J. (2010). Foundational issues in enaction as a paradigm for cognitive science: From the origin of life to consciousness and writing. *Enaction: Toward a new paradigm for cognitive science*, pages 1–31.

Tversky, A. and Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131.

From embodied interaction to compositional referential communication: A minimal agent-based model without dedicated communication channels

Jorge I. Campos^{1,2} and Tom Froese^{2,3}

¹Faculty of Higher Education Aragon, National Autonomous University of Mexico, Mexico City, Mexico

²Institute of Applied Mathematics and Systems Research, National Autonomous University of Mexico, Mexico City, Mexico

³Center for the Complex Sciences, National Autonomous University of Mexico, Mexico City, Mexico

jorgeivan.campos.bravo@gmail.com

Abstract

Referential communication is a "representation-hungry" behavior, and the bee waggle dance is a classical example of referential communication in nature. We used an evolutionary robotics approach to create a simulation model of a minimalist example of this situation. Two structurally identical agents engage in embodied interaction such that one of them can find a distant target in 2D space that only the other could perceive. This is a challenging task: during their interaction the agents must disambiguate translational and communicative movements, allocate distinct behavioral roles (sender versus receiver), and switch behaviors from communicative to target seeking behavior. We found an evolutionary convention with compositionality akin to the waggle dance, correlating duration and angle of interaction with distance and angle to target, respectively. We propose that this behavior is more appropriately described as interactive mindshaping, rather than as the transfer of informational content.

Introduction

Communication in nature shows that there are certain crucial processes that require referential communication in order to keep the species alive. The best example in nature is the waggle dance of the bee (Figure 1), where an explorer bee goes out of the honey comb in order to search for a source of food. Once the explorer bee has found a good field the explorer bee comes back to the honey comb and the dance starts (Crist, 2004; Dornhaus and Chittka, 2004; Dyer, 2002; Seyfarth and Cheney, 2003).

Through the waggle dance, the bee communicates the location of the source to the foraging bees. In the dance, the behavior of the bee corresponds with elements present in the environment that helps the foraging bees to go to the source of food. In animals, simple associations between world entities and signals are mostly innate, or they can be explained by mere mechanisms of rote learning and conditional learning (Cangelosi, 2001). In the case of the waggle dance, the angle with respect to the sun and the center of the dance is assumed to be a genetically determined association of a bee's movements with a certain state of affairs of the world.

In our previous model (Campos and Froese, 2017) we found an analogous communicative behavior performed by

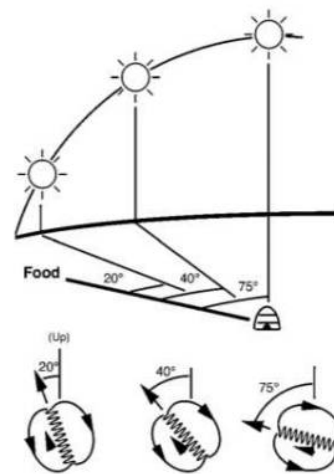


Figure 1: Waggle dance of the bee. The communication of the bee waggle dance depends on the angle at which the buzzing in the center of the figure of "eight" with the sun is performed. In addition the duration of those buzzings correlate with the distance to where the food is located. Figure taken from (Dyer, 2002)

artificial agents that were evolved to solve a minimal referential communication task in a 1D environment. When we analyzed the performance of the agents we found that the best explanation involves an appeal to both agents as an extended system formed via their embodied interaction process: through their interaction the agents shaped the dynamical basis of each other's behavior into giving rise to an adequate solution to the task. This solution is better described as a non-representational form of "mind shaping" (Zawidzki, 2013), rather than a traditional Theory of Mind mechanism. However, it could be argued that the task was not sufficiently "representation-hungry," because its solution did not depend on the principle of compositionality. Here we therefore extended the spatial dimension of the model to a 2D environment and analyzed the resulting behaviors and internal dynamics. We expect that some of the embodied

strategies for achieving referential communication will take advantage of the possibility of separately responding to these two dimensions, and to do so in a combinatorial manner.

Compositionality in communication

In communication compositionality can be defined as the expression of a complex content in terms of a function of the expressions of its parts and their mode of composition.

In other words, the meaning of the whole is determined by the meanings of the parts and the mode of composition. Since the receiver knows the meanings of the simple parts, knows the semantic significance of a finite number of syntactic modes of composition and can parse the expression (i. e. recognize how it is built up out of simple parts) the interpreter can work out the meaning of the whole (Pagin, 2003). In this case we can attempt to explain compositional communication using both agents: how the receiver finds the right interpretation (meaning) and the sender finds an appropriate linguistic item (signal) where the receiver solves a task of interpretation and the sender solves a task of expression.

In our previous model (Campos and Froese, 2017), there is no compositionality present in the communicative behavior: the distance toward the target is simply proportional to the contact time between the agents, and this is sufficient to make possible their communication.

In order to get an intuitive idea about what compositional referential communication consists in, we can consider an alternative modeling approach. The Iterated Learning Approach (Kirby, 2000; Smith et al., 2003) is a framework to study the emergence of the language with compositionality in a sequential model that requires a couple of agents: a mature user of the language (speaker/mentor) and a new user of the language (hearer/learner), which will become the mature agent after learning to teach a new novice agent. During a series of mentor/learner iterations applied to a set of initially arbitrary meaning-signal pairs, a compositional language emerges mainly due to the pressure of keeping the language easy to learn. In other words, structured parts of the signal pick out structured parts of the meaning space.

In contrast, in our minimal referential communication model, there is no predefined meaning-signal pair set and no predefined communication channels. Instead, the "meaning" is the target's location in the environment and the "signal" is the movement of the agents in space. There is no sequential learning process and the pressure of finding a good referential communication system derives directly from the artificial evolution process. Nevertheless, by increasing the environment to two dimensions, we expect the agents to evolve a compositional behavior that involves each dimension of the space (i.e. distance and height) in order to solve the referential communication task. We can then analyze the dynamical basis of this behavior to see whether it involves internal compositional representations, or whether an alternative subpersonal explanation exists.

Interactive approach to referential communication

From the perspective of the dynamical approach to cognition, the components of an agent tend to be in continuous mutual interaction, and none of the components can be removed without thereby also modifying the behavior of the other components. In the case of a system consisting of two agents, this perspective implies that their social interaction process should be better conceived of as a collective property of a brain-body-environment-body-brain system as a whole (Froese et al., 2014). We applied this idea to study the phenomenon of referential communication. In addition, we required that the roles of sender and receiver are initially ambiguous and must be negotiated as part of a continuous flow of nonlinear interaction. Moreover, in contrast to a traditional broadcasting approach to communication, it cannot be assumed on an a priori basis that the agent that turned out to adopt the role of the "receiver" will not play a role in the successful realization of communication. We call this the interactive approach to referential communication.

The interactive approach has the virtue of being a broader perspective that includes the broadcasting approach as a special case, in which the dynamics of the sender are endogenously generated and sufficiently decoupled from its environment. We employ agent-based modeling to help us to develop this alternative theoretical framework.

Previous modeling work

In the field of artificial life there is a long tradition of modeling the evolution of communication and the compositionality of language (Cangelosi and Parisi, 1998; Cangelosi, 2001; MacLennan and Burghardt, 1993; Williams et al., 2008; Manicka, 2012; Nolfi, 2005, 2013). In broad terms it can be said that communication occurs when the behavior of one agent modifies the future behavior of another agent in a task-relevant manner. Several researchers have analyzed the phenomenon as a special example of coordinated behavior between individuals Di Paolo (1997); Di Paolo (2000). But there is still a need to explore how the compositionality of communication can be explained from the dynamical systems approach, and we propose to do so by developing an interactive approach of the referential communication.

Our model is inspired by the work of Williams et al. (2008), who applied the minimal cognition modeling paradigm to the study of referential communication. In their model, they placed two embodied agents, a sender and a receiver, on a 1-D circle. The agents can perceive their own location on the circle and the presence of each other, but only the sender can also perceive the location of the target that the receiver must reach. Following the tradition of research of communication as a form of behavioral coordination (Ackley and Littman, 1994; Di Paolo, 1997; Di Paolo, 2000; Maturana and Varela, 1987), their model does not include dedicated communication channels and so the agents have to learn to distinguish communicative movements from

translational movements (Quinn, 2001). The task is for the receiver to end up as close as possible to the target. They optimized the behavior of the two agents, each controlled by a structurally distinct continuous-time recurrent neural network (CTRNN) (Beer, 1995), using an evolutionary robotics approach (Beer et al., 1996; Harvey et al., 2005). They tested three types of conditions: one involving unconstrained interaction between the sender and the receiver, and two conditions involving constrained interaction. In brief, they found that the easiest solution to the task in the unconstrained condition was for both the sender and the receiver to move to the target together. In this way the sender could use the location of its own body to indicate to the receiver where the location of the target is. However, this solution does not seem to consist in referential communication. To force the evolutionary algorithm to find a solution that involves a form of referential communication they introduced a spatial constraint, such that the receivers position was limited to a subsection of the circle away from the target. They found that it was possible to evolve agents that coordinated their behavior in such a way that the receiver was able to locate a randomly chosen target of a small set of targets (discrete condition) as well as of a range of targets (continuous condition).

But why does artificial evolution tend to prefer solutions to multi-agent coordination problems that are based on joint embodied action rather than on referential communication? This open question makes it even more interesting to consider why bees evolved a complex waggle dance, given that it would be much simpler for the sender bee to directly guide the receiver bees to the target location by simply flying back to where it came from. In other words, it is likely that there is an important constraint that prevents this solution from being feasible. We will return to this open question in the discussion.

In our previous 1D version of this model (Campos and Froese, 2017), we showed that we can reduce the number of nodes in the agents' controllers (in that case to 3 nodes), and use structurally identical artificial neural networks for the two distinct roles (Izquierdo and Buhrmann, 2008), and still evolve the agents to make use of their movements in space as a means for referential communication about target locations of varying distance. Nevertheless, we still had to maintain the constraint of preventing the sender to move directly to the target location; otherwise referential communication would not emerge.

Also, we found an analogous strategy for the agents to solve the task, namely that the distance to the target is correlated with the contact time between the agents, which results from mutually coordinated movements. Therefore, this solution requires both agents to be active in the communicative process for the task to be achieved successfully. However, given that the solution consisted in the co-regulation of one continuous parameter, it did not exhibit compositionality.

The current contribution

The current contribution increases the complexity of our original model in a way that is intended to facilitate the emergence of compositional referential communication. We used the same modeling approach as before, but we expanded the environment from a 1D to a 2D space. We were interested in exploring the possible space of solutions to this 2D-version of the task. Accordingly, we did not pre-specify that the agents had to evolve to solve the task in a compositional manner, although we expected that compositional embodied behavior would be the most effective solution to this task. In other words, we evolved the embodied agents to solve a task that required referential communication between a sender and a receiver and ideally in a combinatorial manner, but without dedicated communication channels, without dedicated roles, and without dedicated signal components.

This kind of scenario is very far removed from traditional studies of communication based on information theory, which typically already assume a well-defined sender, receiver, channel, and symbol system. Our model can therefore serve as an inspiration for thinking about the origins of compositional referential communication in nature.

Methods

The model

Following our approach in Campos and Froese (2017), we created a model in which a pair of embodied agents needs to find a way for the agent named receiver to move through the environment to a target position, but only the agent termed sender knows the exact position of the target. The behavior of the agents was evolved using the methodology of evolutionary robotics (Harvey et al., 2005), using the structure of their artificial neural networks (a standard CTRNN) as their genome. Each artificial neural consisted of 6 nodes, and in line with related work on minimal cognition we used only one structural copy of the network for both agents (Izquierdo and Buhrmann, 2008). We now provide more details of the model.

The task

The task that both agents must fulfill is relatively straightforward: the so-called receiver must arrive at a target area, which will be one of four possibilities in the environment. The environment is a 2-unit-side square with center at (0,0), with a centered interaction zone of 0.6 units side where the sender is constrained to move. The only agent that can sense the position of a target is the so-called sender. The so-called receiver does not sense the target, and the agents only sense each other agent while both remain inside the interaction zone. Once the receiver leaves the interaction zone in order to find the target, there no longer is any possibility for interaction between the agents. Moreover, the receiver cannot

sense the location of the target, and so must find the location of the target by relying solely on the history of embodied interaction between the agents (see Figure 2).

Each pair of agents was tested in four separate trials using four distinct targets in the environment. These targets were selected by arbitrarily varying the distance and angle to the interaction zone within fixed limits.

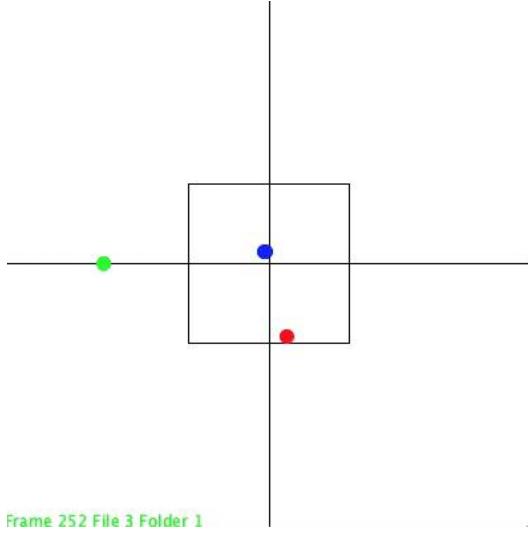


Figure 2: Simulation of the agents in the environment. The blue circle represents the sender while the red circle represents the receiver. The green circle is the target, which is in a random position along one of the two axes. The square in the middle position represents the interaction zone. After interacting in the interaction zone, the red agent has to leave the zone and try to locate the target based only on its history of interaction with the other agent.

The agents

The sensorimotor system we used has six sensors and two motors, which are the inputs and outputs of an agent's six-node CTRNN (see Figure 3).

Three of an agent's sensors serve to locate the other agent's position with respect to the self's location: a distance sensor providing a continuous input that varies linearly with the distance to the other agent, and two sensors providing the angle to the other agent, i.e. a sensor for $\text{Sin}(\theta)$ and another for $\text{Cos}(\theta)$. If the receiver moves outside the interaction zone then these three sensors are turned off in both agents. So the range of the sensor is $[0, 1]$, where 0 is the maximum distance in the interaction zone (and also any further distances, which imply that the receiver is outside the zone), and 1 indicates that both agents are exactly on top of each other.

The other three of an agent's sensors serve to locate the target's position with respect to the self's location: a dis-

tance sensor providing a continuous input that varies linearly with the distance to the target, and two sensors providing the angle to the target, i.e. a sensor for $\text{Sin}(\theta)$ and another for $\text{Cos}(\theta)$. These three sensors are always turned on in the sender and always turned off (i.e. set to -1) in the receiver. Given that the sender is constrained to the interaction zone, the distance sensor is restricted to range $[0.2, 0.8]$.

Each agent has a body in the shape of a circle of radius 0.05 that has two simulated wheels, one on each side, giving the agent a facing property. The velocity of a wheel is controlled by the output of a dedicated neuron, and these motor neurons were arbitrarily chosen to be the nodes also connected to the distance sensors. The agents control their overall velocity using differential steering control, which means that for an agent to move forward at a certain speed it is necessary that both wheels have the same velocity. Collisions are not modeled.

Artificial neural network

The behavior of each agent is modulated by a continuous-time neural network (CTRNN) (Beer, 1995) (Figure 3), whose equation is the following:

$$\tau_i \dot{s}_i = -s_i + \sum_{j=1}^N (\omega_{ij} \sigma(s_j + \theta_i)) + G_i I_i \quad i = 1, \dots, N \quad (1)$$

where s is the state of each neuron, τ is the time constant, ω_{ij} is the weight of the j -th neuron to the i -th neuron, θ is the bias, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the standard logistic activation function, G is the gain constant of the input of the neuron and I is the input of each neuron.

The CTRNNs of the two agents are structurally identical. The integration step size was set to 0.1.

Evolution of the agents

The parameters of the neural network were encoded using floating-point numbers and then were optimizing using an artificial evolution procedure. We perform 50 separate evolutionary runs and in each we used 100 individuals that were evolved for 1,000 generations. In each generation, the individuals were evaluated in the environment as follows:

At the start of a trial the agents are placed randomly inside the interaction zone within a circle of radius 0.3, and always facing rightward (0 degree). They are then allowed to behave for 300 units of time. At the end of the trial their fitness score for that trial is calculated with the following equation: $\text{fitness} = 1 - \text{finalDistanceToTarget}$. Every time we picked a target, we calculated the mean fitness over the population of pairs. Once the fitness increased by 20% from this initial value, we switch the target for a different randomly selected target location. This was done to allow local task optimization but without losing generality.

Each pair of agents is evaluated for 10 trials. Overall fitness of the pair is calculated as the inverse weighted average score of all 10 trials.

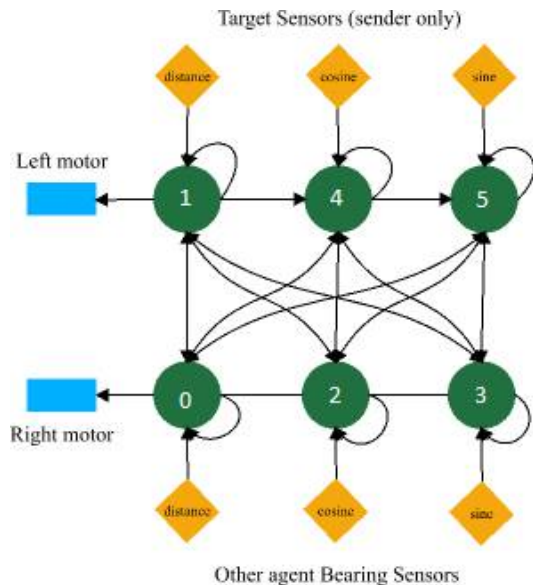


Figure 3: Illustration of artificial neural network structure. Green circles represent each node of the six-node network. Orange diamonds represent the six sensors, and the blue rectangles show the two motors.

The evolutionary algorithm employed the hill-climbing method: each solution generates a new solution by applying Gaussian mutation with variance of 0.2 to each parameter, and this descendant replaces the parent solution if its performance is better.

Results

Once the 50 independent evolutionary runs finished, we saved the best pair of agents of each run. With these resulting 50 pairs of agents we performed several tests to see the performance of the agents for different initial conditions.

Behavior analysis

By looking at the 50 best agents we noticed that this is a challenging task. Most of the runs failed to give rise to strategies involving referential communication. Instead the agents relied on suboptimal strategies, such as getting to a certain target spawning position. In some cases the agents apparently got too entrained in the interaction and the receiver never left to find the target in the first place. In other cases the interaction between the agents resulted in complex behavior of the receiver, but without getting close to the target.

Just one evolutionary run out of the 50 runs accomplished the task successfully: the two agents interacted at the beginning of the trial and then the receiver moved to where the target is. The rest of our analysis focuses on this successful pair of agents.

To confirm that their strategy relied on referential communication rather than on direct influence on each other's behavior via the agent sensors, we ran a test trial where we removed the sender as soon as the receiver first goes outside the interaction zone. Using this setup, most trials still achieved the task. A few failed, but this was because we broke the interaction too soon for the receiver to fulfill the task (i.e. the receiver had returned into the interaction zone but the sender was no longer there).

To improve the success of these trials we introduced this pair of agents in a new optimization process. This time we generated a population of 300 individuals that begin with the same CTRNN, and we change the task by adding the constraint that once the sender has left the interaction zone he can't enter again and interact with the sender. We let the population evolve through 300 generations.

At the end of this new evolutionary run, the best pair of agents found the target on all of the trials. We further tested these agents as described in the following subsections.

Overall performance

We performed 1,000 test trials with these best agents, placing random targets all over the environment. More specifically, at the start of each trial a target was randomly placed inside a space defined by x-range $[-1, 1]$ and y-range $[-1, 1]$, but not inside the interaction zone (ranges $[-0.6, 0.6]$). We got the following distribution of final distances between the receiver and the target (Figure 4).

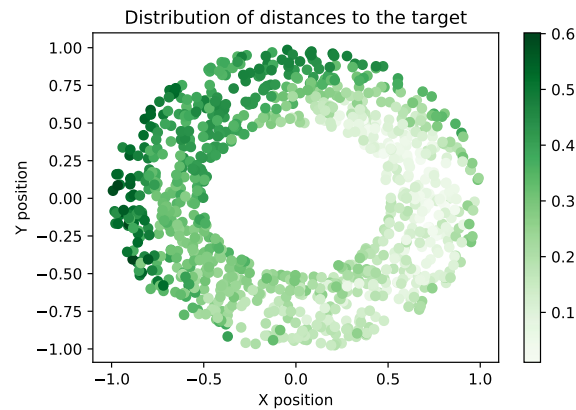


Figure 4: Distribution of 1,000 randomly placed test target locations. Shading represents distances between the receiver and the target at the end of a trial. Light green represents closer to the target and darker represents further away.

For the 1,000 trials we got an average distance to the target of 0.262094, where the most of the trials with the target at the right of the agent's interaction zone the distances range between 0.15 and 0.05. The worst case was located at $[-0.8858, -0.4051]$ with a distance of 0.6014 one of the

right most target appearances. We can say that the agents generalized well the task for most of these targets, because the agents were evolved to achieve the task only for 4 different angle locations.

In general, the agents perform the task with the following behavior. First, they spend some time interacting inside the interaction zone. This typically involves spiraling in ever larger circles. Then, the receiver eventually is on a trajectory that takes it out from the interaction zone, and then it starts its movement trajectory to the target to end up as close as possible based on the history of interaction with the sender.

Strategy per target position

We are interested in finding the strategy that the agents perform to achieve the task.

First, we compared the final interaction distances between the sender and receiver, i.e. their distance at the moment when the receiver leaves the interaction zone, with the distance from the center of the space to the target location. We wanted to see if there is a correlation between their distance and target distance. And indeed, we find such a correlation, at least for some areas of the space (see Figure 5).

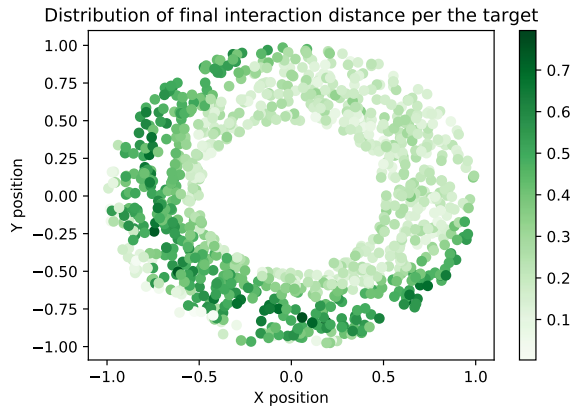


Figure 5: Distribution of 1,000 randomly placed test target locations. Shading represents the distance between the agents before their interaction breaks off, i.e. before the receiver leaves the zone of interaction. Light corresponds to a shorter and dark corresponds to a longer distance. There is a tendency for targets that are further away from the interaction zone to be associated with greater distances between the agents (darker dots).

Next, we analyzed the angles exhibited by the agents when their interaction ends (Figure 6). What we found is that there is also a relation between the angle between the agents at the end of their interaction and the angle from the center of the interaction zone to the target position. This relation complements the previous relation.

The agents found a strategy, in general, using two components: their distance to each other and their angle with respect to each other. This is the compositionality we tried to find, similar to the compositionality of the bee’s dance, albeit in our model there was not such a clear demarcation of the components of the referential communication given that everything is embedded in a parallel manner in a continuously spiraling mutual interaction.

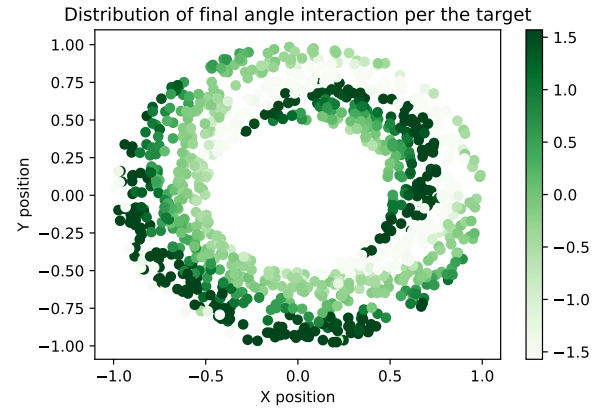


Figure 6: Distribution of 1,000 randomly placed test target locations. Shading represents the difference between the angle between the agents at their final point of contact and the angle from the center of the interaction zone toward the target’s location. The shades of green represent the angle difference going from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, which is the possible range of angle differences for the quadrant in which the target is located.

Attractor analysis of the artificial neural network

In order to get an intuitive grasp of the state space configuration of the CTRNN of the best pair of agents, we performed 10 tests with different initial conditions with the isolated neural network. We found that the trajectories in CTRNN state space converged toward a single attractor located at coordinates $[-7.10114, 4.76134, -6.69269, 10.3879, -8.68615, 0.199544]$. The fact that the agent’s dynamical system contains a single attractor means that the complexity of the behavioral solution of the task is strongly dependent on the interaction between the agents in the environment. It is a solution that is better conceived of as a property of the system as a whole than as a property of an agent’s neural network activity. This is consistent with what we found in our previous 1D model.

In other words, it appears that increasing the complexity of the referential communication task has only increased the complexity of the collective system but not the internal complexity of the agents. This increase in interaction complexity without concomitant increase in internal complexity is con-

sistent with findings reported from another multi-agent system modeling study that investigated this relationship more systematically (Candadai et al., 2019).

In Figure 7 we show the states of three neurons of the isolated CTRNN starting from arbitrary initial states, where in this 3D-space all the states converge on the same attractor.

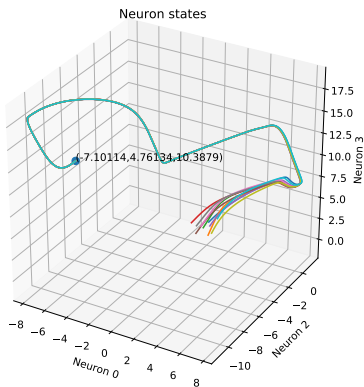


Figure 7: Neural states of 3 representative neurons of the agent’s decoupled CTRNN during 10 trials with different initial conditions. The colored lines represent each trial and the point represents the attractor of the system

Discussion

This model illustrates the interactive approach to referential communication, where the agents tend to be in continuous mutual interaction. We chose a task that cannot be accomplished by the receiver alone, because the information about the target is only available by the sender. However, the sender also cannot behave as expected without the presence of a receiver that interacts in the right manner, giving rise to a mutual interaction process.

The artificial neural network of both the sender and receiver is structurally the same. That is why the agents must negotiate their respective roles during their embodied interaction before one of them attempts to go to the target. Once the interaction begins they must coordinate their behavior to break their interaction at the right moment. This means that, in accordance with the interactive approach, successful referential communication is co-created by both agents: their individual behaviors give rise to the interaction process but the interactive process shapes their individual behaviors.

As we can see in figures 5 and 6 we can analyze their mutual interaction process into separate components that together help the receiver to reach the target position in space. This can be seen as the compositionality of referential com-

munication that was absent in our previous 1D version of the model.

Interestingly, the complex behavior of the agents does not require equally complex internal organization of their CTRNNs. As Figure 7 shows, the agents succeed with a CTRNN that in its decoupled mode only exhibits a single attractor. In other words, the complexity of the behavior is outsourced into the complexity of the interaction itself. This finding is in line with the enactive approach to social interaction, which holds that social cognition can be constituted by social interaction (De Jaegher and Froese, 2009), and that this can lead to performance that would otherwise be outside the agents’ reach in practice and even in principle (Froese et al., 2013).

Future work

In order to be able to evolve agents that were able to perform the 2D version of the referential communication task, we had to increase the number of CTRNN nodes to six nodes from the three nodes we used for the 1D version. It is possible that this increased internal dimensionality is needed for the receiver to sufficiently cope after being decoupled from the sender, as was also found in related modeling work (Fernandez-Leon and Froese, 2010). This requires further analysis, but it seems that more nodes may facilitate the production of more robust internal dynamics for the agents.

The best-evolved strategy also needs to be analyzed in more detail. While we found that two components of the relationship between the agents at the end of their mutual interaction, namely their distance and angle, tended to correspond to the distance and angle to the target, this correspondence was perhaps not as consistent or uniform as would normally be expected from a genetically evolved referential communication strategy such as the bee’s dance. In future work we would therefore like to apply other measures.

Another interesting direction for future research is to increase the number of agents in the model. For instance, if several senders had to compete with each other in the interaction zone (similar to the chaotic situation of multiple concurrent dances in a bee’s hive), it would not be convenient for the sender to simply return to the target’s location with other agents in tow. In other words, perhaps the externally imposed restriction of the sender to the interaction zone would no longer be necessary under such more realistic conditions.

Acknowledgements

This work was supported by CONACYT scholarship for J.I.C.

References

- Ackley, D. H. and Littman, M. L. (1994). Altruism in the evolution of communication. In Brooks, R. A. and Maes, P., editors, *Artificial life IV: Proceedings of the Fourth International*

- Workshop on the Synthesis and Simulation of Living Systems, pages 40–48, Cambridge, MA. MIT Press.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509.
- Beer, R. D. et al. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, *From animals to animats: Proceedings of the Fourth International Conference on Simulation of Adaptive behavior*, volume 4, pages 421–429, London, England. MIT Press.
- Cangelosi, A. and Parisi, D. (1998). The emergence of a 'language' in an evolving population of neural networks. *Connection Science*, 10(2):83–97.
- Campos, J. I. and Froese, T. (2017). Referential communication as a collective property of a brain-body-environment-body-brain system: A minimal cognitive model. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 863–870, Honolulu, Hawaii, USA. IEEE Press.
- Candadai, M., Setzler, M., Izquierdo, E. J., and Froese, T. (2019). Embodied dyadic interaction increases complexity of neural dynamics: A minimal agent-based simulation model. *Frontiers in Psychology*, 10:540.
- Cangelosi, A. (2001). Evolution of communication and language using signals, symbols, and words. *IEEE Transactions on Evolutionary Computation*, 5(2):93–101.
- Crist, E. (2004). Can an insect speak?: The case of the honeybee dance language. *Social Studies of Science*, 34(1):7–43.
- De Jaegher, H. and Froese, T. (2009). On the role of social interaction in individual agency. *Adaptive Behavior*, 17(5):444–460.
- Di Paolo, E. A. (1997). Social coordination and spatial organization: Steps towards the evolution of communication. In Husband, P. and Harvey, I., editors, *Proceedings of the Fourth European Conference on Artificial Life (ECAL'97)*, pages 464–473. MIT Press.
- Di Paolo, E. A. (2000). Behavioral coordination, structural congruence and entrainment in a simulation of acoustically coupled agents. *Adaptive Behavior*, 8(1):27–48.
- Dornhaus, A. and Chittka, L. (2004). Why do honey bees dance? *Behavioral Ecology and Sociobiology*, 55(4):395–401.
- Dyer, F. C. (2002). The biology of the dance language. *Annual Review of Entomology*, 47(1):917–949.
- Fernandez-Leon, J. A. and Froese, T. (2010). What is the relationship between behavioral robustness and distributed mechanisms of cognitive behavior? In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.
- Froese, T., Gershenson, C., and Rosenblueth, D. A. (2013). The dynamically extended mind. In *2013 IEEE Congress on Evolutionary Computation*, pages 1419–1426. IEEE.
- Froese, T., Iizuka, H., and Ikegami, T. (2014). Embodied social interaction constitutes social cognition in pairs of humans: a minimalist virtual reality experiment. *Scientific reports*, 4:3672.
- Harvey, I., Paolo, E. D., Wood, R., Quinn, M., and Tuci, E. (2005). Evolutionary robotics: A new scientific tool for studying cognition. *Artificial life*, 11(1-2):79–98.
- Izquierdo, E. and Buhrmann, T. (2008). Analysis of a dynamical recurrent neural network evolved for two qualitatively different tasks: Walking and chemotaxis. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial life XI: Proceedings of the eleventh international conference on the simulation and synthesis of living systems*, pages 257–264. MIT Press.
- Kirby, S. (2000). Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. In Knight, C., Studdert-Kennedy, M., and Hurford, J., editors, *The Evolutionary Emergence of Language: Social Function and the Origins of Linguistic Form*, pages 303–323, Cambridge, MA. Cambridge University Press.
- MacLennan, B. J. and Burghardt, G. M. (1993). Synthetic ethology and the evolution of cooperative communication. *Adaptive Behavior*, 2(2):161–188.
- Manicka, S. (2012). Analysis of evolved agents performing referential communication. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *ALIFE 2012: The Thirteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 393–400. MIT Press.
- Maturana, H. R. and Varela, F. J. (1987). *The tree of knowledge: The biological roots of human understanding*. New Science Library/Shambhala Publications, Boston, MA.
- Nolfi, S. (2005). Emergence of communication in embodied agents: co-adapting communicative and non-communicative behaviours. *Connection Science*, 17(3-4):231–248.
- Nolfi, S. (2013). Emergence of communication and language in evolving robots. In Lefebvre, C., Comrie, B., and Cohen, H., editors, *New perspectives on the origins of language*, pages 533–554. John Benjamins, Amsterdam, Philadelphia.
- Pagin, P. (2003). Communication and strong compositionality. *Journal of Philosophical Logic*, 32(3):287–322.
- Quinn, M. (2001). Evolving communication without dedicated communication channels. In Kelemen, J. and Sosk, P., editors, *Advances in Artificial Life: 6th European Conference on Artificial Life, ECAL 2001*, pages 357–366, Berlin, Heidelberg. Springer.
- Seyfarth, R. M. and Cheney, D. L. (2003). Signalers and receivers in animal communication. *Annual Review of Psychology*, 54(1):145–173.
- Smith, K., Kirby, S., and Brighton, H. (2003). Iterated learning: A framework for the emergence of language. *Artificial Life*, 9(4):371–386.
- Williams, P. L., Beer, R. D., and Gasser, M. (2008). Evolving referential communication in embodied dynamical agents. In Seth Bullock, Jason Noble, R. W. and Bedau, M. A., editors, *Artificial life XI: Proceedings of the eleventh international conference on the simulation and synthesis of living systems*, pages 702–709. MIT Press.
- Zawidzki, T. W. (2013). *Mindshaping: A New Framework for Understanding Human Social Cognition*. MIT Press, Cambridge, MA.

Void Reduction in Self-Healing Swarms

Neil Eliot¹, David Kendall¹, Alun Moon¹, Michael Brockway¹ and Martyn Amos¹

¹ Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, NE1 8ST, UK
neil.eliot@northumbria.ac.uk

Abstract

Swarms consist of many agents that interact according to a simple set of rules, giving rise to emergent global behaviours. In this paper, we consider swarms of mobile robots or drones. Swarms can be tolerant of faults that may occur for many reasons, such as resource exhaustion, component failure, or disruption from an external event. The loss of agents reduces the size of a swarm, and may create an irregular structure in the swarm topology. A swarm's structure can also be irregular due to initial conditions, or the existence of an obstacle. These changes in the structure or size of a swarm do not stop it from functioning, but may adversely affect its efficiency or effectiveness. In this paper, we describe a self-healing mechanism to counter the effect of agent loss or structural irregularity. This method is based on the reduction of concave regions at swarm perimeter regions. Importantly, this method requires no expensive communication infrastructure, relying only on agent proximity information. We illustrate the application of our method to the problem of surrounding an oil slick, and show that void reduction is necessary for full and close containment, before concluding with a brief discussion of its potential uses in other domains.

Introduction

The natural phenomenon of *swarming* in organisms such as insects, fish and birds has, for a long time, served as inspiration for algorithmic solutions to problems (Blum and Merkle, 2008). Swarm-based algorithms use a number of *agents* which behave according to local rules (locality often being defined in terms of spatial proximity), but which - collectively - are capable of synergistically cooperative behaviour. Problems to which such methods have been applied include path finding (Hou et al., 2009), distribution across a space (Ekanayake and Pathirana, 2010; Gazi and Passino, 2002, 2004a), or foraging as a colony (Gurfil and Kivlevitch, 2007; Hereford, 2011). In order to model inter-agent interactions, many algorithms use *field effects*, which capture *attractive* and *repulsive* forces between agents (Andreou et al., 2009; Barnes et al., 2006a,b; Bennet and McInnes, 2009; Gazi and Passino, 2002, 2004b, 2005, 2011; Mohan and Ponnambalam, 2009). Attraction is used as a *cohesive* force to bring agents close together, and repulsion is used to prevent collisions.

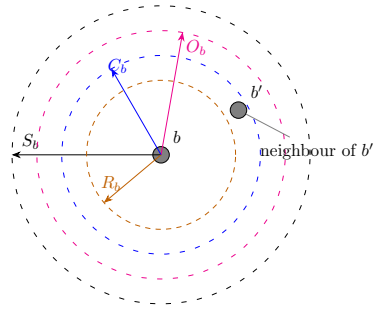


Figure 1: Agent field ranges. R_b implements repulsion, C_b implements cohesion, S_b is the agent's sensing range, and O_b is used to manage collisions with obstacles.

Forces are generally defined in terms of *ranges* around an agent, and the field effects are derived as vectors from these ranges (Figure 1). For any agent, b , all ranges must fall within the *sensing capability* of the agent, S_b , which might represent a visual or auditory range, some chemical sensing capability, or (in the context of mobile robotics) a communication range. It is usual for the cohesion field, C_b , to have a radius which is larger than the repulsion radius, R_b (so that agents are encouraged to group together, but not too closely). When another agent, b' , moves into the cohesion range of b then b' becomes a *neighbour* of b ; when b' moves into the *repulsion* field of b , then b is also subject to repulsion. When the repulsion magnitude exceeds the cohesion magnitude, then b has a tendency to move away from b' , i.e., it is repelled. When b moves too close to an obstacle, i.e., an obstacle is within the obstacle repulsion range, O_b , the repulsion vector is applied and the agent tends to move away from the obstacle.

When cohesion and repulsion are the only field effects used to create a swarming effect, the number of stable structures that can develop is limited (Eliot, 2017). These structures effectively take the form of either straight edges or partial lattices (Figure 2). The maintenance of a well-structured swarm is crucial to their effective deployment in a number of applications, including reconnaissance or artificial polli-

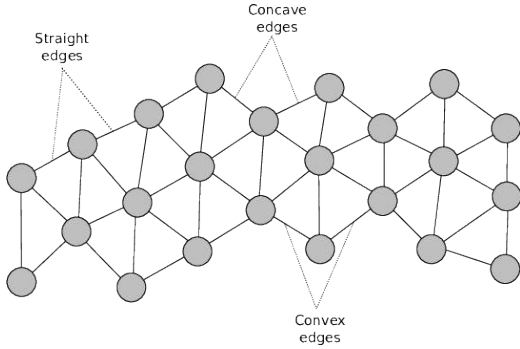


Figure 2: Stable swarm structure containing two types of anomaly.

nation, where coverage “blind spots” are eliminated (Elamvazhuthi and Berman, 2015), and containment, where the swarm is used to surround an object or region (Cao et al., 2012). Over time, the perimeters of partial lattices may contain so-called *anomalies*, such as concave “dents” or convex “peaks”, and these anomalies all contribute to the disruption of an otherwise well-structured swarm. The key, therefore, is to ensure that concave *voids* are dynamically removed from a swarm.

Here, we describe our *void reduction* technique for swarm management, which is a form of self-healing that encourages a swarm to coalesce into a more geometrically stable shape. This is achieved by removing voids and concave edges. Importantly, the techniques defined in this paper function without the need for inter-agent or global *messaging* (which can carry a significant overhead), and rely only on local *proximity detection*.

The rest of the paper is organized as follows: we first briefly review related work in the area of self-healing swarms, and then describe the baseline swarming model and our novel perimeter detection and void reduction mechanisms. We describe the results of computational studies in a specific application domain (surrounding an oil slick), before we conclude with a brief discussion of our results, and give pointers to possible future work.

Related Work

A prototype framework for self-healing swarms was developed by Dai, *et al.*, which considered the problem of agent failure in hostile environments (Dai et al., 2006). This was similar to work carried out by Vassev and Hinchey, who modelled swarm deployment using the ASSL (Autonomic System Specification Language) (Vassev and Hinchey, 2009). This technique was used by NASA (US National Aeronautics and Space Administration) when developing their ANTS (Autonomous Nano Technology Swarm) for use in asteroid belt exploration. However, this work was focused more towards the failure of an agent’s internal sys-

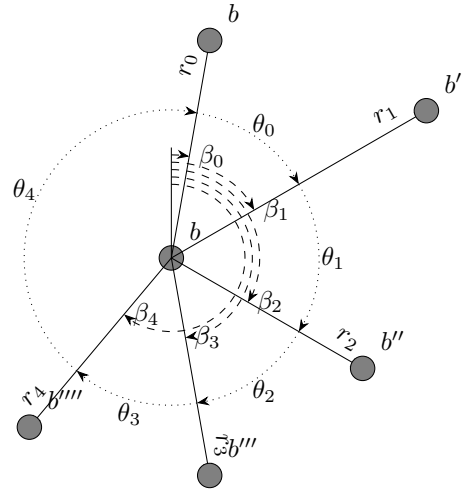


Figure 3: Swarm model: representation of interaction with neighbouring agents.

tems, rather than on the removal of anomalies in a swarm distribution.

In the context of swarm structure maintenance, Roach, *et al.* focussed on the effects of sensor failure, and the impact that this has on agent distribution (Roach et al., 2015). Lee and Chong identified the issue of concave edges within swarms in an attempt to create regular lattice formations (Lee and Chong, 2008), and the main focus of their work is the dynamic restructuring of inter-agent formations. Ismail and Timmis demonstrated the use of *bio-inspired* healing using *granuloma formation*, a biological method for encapsulating an antigen (Ismail and Timmis, 2010). They have also considered the effect that failed agents can have on a swarm when traversing a terrain (Timmis et al., 2016).

Our void reduction technique is an extension of the work presented in (Ismail and Timmis, 2010; Timmis et al., 2016), and also builds on the work of Lee and Chong on concave edge identification (Lee and Chong, 2008), and on the work of McLurkin and Demaine on the detection of perimeter types (McLurkin and Demaine, 2009). However, the technique employed in this paper does not explicitly require the identification of the perimeter type, as this would require a communication infrastructure.

Swarm Model

In this Section, we define the baseline swarm model. A swarm, \mathcal{S} , comprises a number of agents; in our application context, each agent is a mobile robot or drone, but this may remain unspecified. An agent $b \in \mathcal{S}$ has a sensor range, S_b , within which it may detect other agents in the swarm, and determine both their *range*, r , and *bearing*, β (Figure 3). At each time step, the agent generates a set of neighbours, \mathcal{N}_b , comprising other agents that are within a specific range

(usually defined as the range of the cohesion field, C_b), as given in Equation 1. These range and bearing pairs contain the relative position vector for each neighbour, b' , with respect to the sensor reference frame of agent b . This model was defined by Eliot, *et. al.* in a paper which introduced a new magnitude-based metric for the analysis of swarms (Eliot et al., 2018).

$$\mathcal{N}_b = \{(r, \beta) \dots\} \quad (1)$$

In order to calculate the new vector, v , for b , Equation 2 defines a weighted model that includes cohesion, repulsion, direction and obstacle avoidance ($v_c(b)$, $v_r(b)$, $v_d(b)$, and $v_o(b)$, respectively). The weightings k_c , k_r , k_d , k_o allow each component to be scaled in order to tailor the swarming effect.

$$v(b) = k_c v_c(b) + k_r v_r(b) + k_d v_d(b) + k_o v_o(b) \quad (2)$$

Repulsion, $v_r(b)$, defined in Equation 3, is the directional movement required to prevent agents colliding. \mathcal{R}_b is defined as the set of agents that are within the repulsion range of b .

$$v_r(b) = \frac{1}{|\mathcal{R}_b|} \left(\sum_{b' \in \mathcal{R}_b} \left(1 - \frac{|b'|}{R_b} \right) b' \right) \quad (3)$$

Cohesion, $v_c(b)$, defined in Equation 4, calculates the movement required to make an agent move towards other agents in order to form a cohesive structure. \mathcal{C}_b is defined as the set of agents that are within the cohesion range of b .

$$v_c(b) = \frac{-1}{|\mathcal{C}_b|} \left(\sum_{b' \in \mathcal{C}_b} b' \right) \quad (4)$$

Direction, $v_d(b)$, defined in Equation 5, generates a directional vector for an agent to move towards some destination, d .

$$v_d(b) = d \quad (5)$$

Obstacles, like agents, may be represented as a point. As an agent moves, it may enter an obstacle's *repulsion field*. If this occurs, then the agent should move away (as we assume that an obstacle is unable to take evasive action itself). Here, agents have a fixed *obstacle repulsion field*, O_b . If an obstacle enters the field, a vector of magnitude O_b is applied. If more than one obstacle is present within the field, the applied repulsion vector is the sum of the repulsion vectors (Figure 4). The resultant vector is normalised and scaled such that the magnitude is the same as the field distance, O_b , as given in Equation 6.

Equation 6 shows the repulsion vector, $v_o(b)$, for an agent. \mathcal{O}_b is the set of obstacles within the range of agent

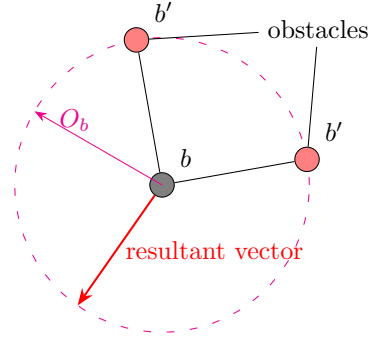


Figure 4: Repulsion from obstacles.

b . The obstacles are identified by comparing their Cartesian distance to the fixed obstacle repulsion field O_b , so $\forall o \in \mathcal{O}_b : |o| \leq O_b$. The applied repulsion is calculated by scaling the normalised sum of the normalised vectors \hat{o} by O_b . Note that $\hat{\cdot}$ is the equivalent of $\hat{v} = \frac{v}{|v|}$, the normalised vector.

$$\begin{aligned} v_o(b) &= O_b \hat{q}_o & (6) \\ \text{where } q_o &= \sum_{o \in \mathcal{O}_b} \hat{o} \\ v_o(b) &= O_b \left(\sum_{o \in \mathcal{O}_b} \hat{o} \right)^\wedge \end{aligned}$$

An agent's *movement vector* is defined as the sum of all the component vectors, as shown in Equation 2 (similar to that used by Hashimoto, *et. al.* (Hashimoto et al., 2008)). In order for a vector to be used for movement, it must be normalised before the agent's speed, s_b , can be applied. The resulting movement vector, m_b , is defined in Equation 7, and is calculated using unit time, speed and the normalised *movement vector*.

$$m_b = s_b \hat{v}(b) t \quad (7)$$

Over time, applying the calculations described in this Section to all agents in turn creates the global swarming effect. This provides the *baseline* algorithm for swarm movement. We now describe how the swarm may be *dynamically reconfigured*, which is the main novel contribution of this paper. After describing our new algorithm for void reduction, we show how it may be applied to a specific problem.

Perimeter Detection

In order to dynamically restructure a disorganised swarm, we must first identify the *perimeter* agents. This is due to the fact that anomalies occur at swarm boundary locations. With reference to Figure 5, these agents may form part of an outer (green) or inner (red) edge.

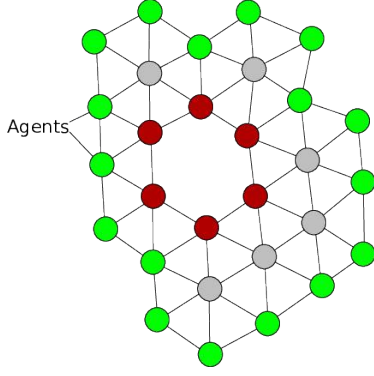


Figure 5: **Outer** and **inner** swarm perimeters.

Our detection mechanism detects both the outer edge of a swarm and any internal features (voids) that satisfy the same set of conditions (Figure 5). It is therefore possible to have both voids and “islands” of agents within the same swarm. Voids are best defined as perimeters that are both concave (Equation 15) in nature and which exist inside another perimeter. McLurkin (McLurkin and Demaine, 2009) describes two types of perimeters, convex and concave, where a convex perimeter is an edge where the average angle of the exposed faces of relevant agents is $> 180^\circ$, and a concave perimeter is one where the average exposed angle is $< 180^\circ$.

The set of neighbours, \mathcal{N}_b (Equation 1) is sorted into the sequence \mathcal{P}_a , in ascending order of bearing:

$$\mathcal{P}_a = \langle (r_0, \beta_0), \dots, (r_n, \beta_n) \rangle \quad (8)$$

such that $\beta_0 < \beta_1 < \dots < \beta_n$

This set of agents forms the perimeter of an enclosing polygon of agent b . Each consecutive pair of agents in the sequence defines an *edge*, which has length d and an angle θ given by the difference in bearings of successive neighbours. The sequence of edges that forms this polygon is:

$$\mathcal{P}_e = \langle (d_0, \theta_0), \dots, (d_n, \theta_n) \rangle \quad (9)$$

where

$$\theta_i = \beta_{i+1} - \beta_i \quad (10)$$

The index addition is modulo $|\mathcal{N}_b|$, making β_0 the successor bearing to β_n ($n+1=0$). The angles θ must lie in the range $0 < \theta \leq 2\pi$. This restriction on the values of θ enforce the condition that

$$\sum \theta_i = 2\pi \quad (11)$$

The length of a perimeter edge is given by the cosine rule

$$d_i^2 = r_{i+1}^2 + r_i^2 - 2r_{i+1}r_i \cos \theta_i \quad (12)$$

An agent is therefore on the perimeter of the swarm if it is not enclosed by the polygon defined in \mathcal{P}_e . Simple geometry shows that this is the case, given by the predicate in Equation 13.

$$\exists \theta_i \in \mathcal{P}_e : \theta_i \geq \pi \quad (13)$$

The polygon is considered to be “open” if two successive agents on the perimeter are unable to “see” one another; that is, their separation, d , is greater than the range of the attractive field. An open polygon does not enclose the agent b , so it is considered to be on the perimeter.

Formally, an agent, b , is on the perimeter of the swarm if the predicate in Equation 14 is true.

$$\exists d_i \in \mathcal{P}_e : d_i > C_b \vee \exists \theta_i \in \mathcal{P}_e : \theta_i \geq \pi \quad (14)$$

An agent is at the apex of a concave region of the perimeter if

$$\exists (\theta_i, d_i) \in \mathcal{P}_e : d_i > C_b \wedge \theta_i < \pi \quad (15)$$

The orientation is independent in so much as: if the agent b is rotated through an angle of γ then the bearings are rotated by $-\gamma$,

$$\beta_i \mapsto \beta_i - \gamma$$

The angle between successive agents is now

$$\theta_i = (\beta_{i+1} - \gamma) - (\beta_i - \gamma) = \beta_{i+1} - \beta_i - \gamma + \gamma = \beta_{i+1} - \beta_i$$

Void Reduction

In a static swarm, where there are essentially no *destination vectors*, void reduction will result in a restructuring motion that creates a more “rounded” swarm. Void reduction also creates a *surrounding* effect, as it removes voids from a swarm. This is discussed in more detail in the next Section. Although these effects improve the potential applications of swarms, negative effects may also be introduced (e.g., in some circumstances void reduction can create an artificial *destination vector*, in that the swarm will appear to have a directional movement).

In order to implement void reduction, full *perimeter detection* is required in order to identify candidate agents (Eliot, 2017). Void reduction does not require the perimeter *type* to be identified, and no communications infrastructure is required. Many existing swarm coordination algorithms require inter-agent communication (Jung and Goodrich, 2013; McLurkin and Demaine, 2009; Saldana et al., 2012; Navarro and Matía, 2009; Zhang et al., 2013), and this imposes a significant limitation on swarm size, due to the requirement for message propagation. Our method avoids the problems associated with this.

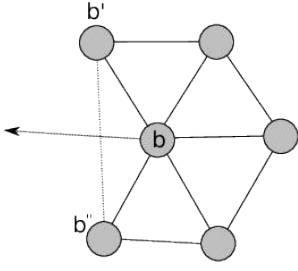


Figure 6: Agent void reduction motion: agents b' and b'' form a concave edge (depicted by the lighter line). Agent b must therefore move to remove this edge.

Void Reduction: Agent Movement

The addition of a further characteristic to the motion of a swarm means that we must augment the existing agent model (Equation 2). With void reduction, this revision is based on the identification of the agents that are connected by a concave edge, as shown in Figure 6 as (b', b, b'') .

When an agent is identified as being a component of a void characteristic (Equation 15), the normal *movement-direction vector* is replaced by a *void reduction vector*. This new vector causes the agent to move in a direction that will reduce or remove a concave edge, by moving the agent towards the identified gap. This either straightens an outer perimeter, or reduces/removes a void. The change in direction also affects the distance and magnitude variances. Figure 7 shows this effect in more detail; the top figure shows the initial positions of the agents before void reduction is applied, and the bottom part of the figure shows the effects on its relationship with its neighbours. The aggregate change is an increase in the inter-agent distances, and an increase in the resultant magnitude effects.

As part of the perimeter detection process, we may generate a set of agents, G_b , that produce a gap for a particular agent, b (that is, the first two agents identified as creating a “gap” in agent b ’s neighbours). Equation 16 is then used to calculate the centroid of the “gap” agents:

$$D_{pos}(b) = \frac{1}{2} \sum_{b' \in G_b} b' \quad (16)$$

The centroid $D_{pos}(b)$ is then used to calculate the *void reduction vector*:

$$D(b) = D_{pos}(b)b \quad (17)$$

$D(b)$ is the vector from the coordinates of agent b to the centroid coordinates, $D(b)$. This new vector is used as the void reduction vector in order to implement the necessary void reduction movement (Equation 17).

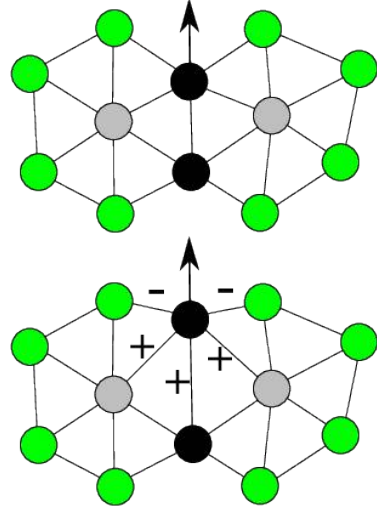


Figure 7: Initial position (top), and reduced position (bottom). +/- labels show relative changes in inter-agent magnitude.

In addition to agent proximity, the void reduction movement process must also include *obstacle avoidance* (Equation 18). As with the earlier vector-based calculations, a weighting, k_{cr} , is applied to the void reduction vector in order to allow the model to adjust the application of the effect. The resultant void reduction vector is normalised to produce a *directional vector*, as shown in Equation 18. This is then applied to the agent in order to effect movement.

$$V(b) = (k_{cr}D(b) + k_o v_o(b))^\wedge \quad (18)$$

Experimental Results: Oilslick Containment

In this Section we give the results of experiments to simulate a specific scenario; that of *oilslick containment* using a mobile robot swarm. Oil spills (from ships or drilling operations) can cause significant environmental, social and economic damage, and removing them can be hazardous and expensive. Several alternatives to traditional spill dispersal/containment procedures have been proposed, with some proposals relying on the use of robot swarms to surround a spill (Fritsch et al., 2007; Kakalis and Ventikos, 2008; Zhang et al., 2013) (details of remediation processes are outside the scope of this paper, but they may include skimming of the surface, deposition of a dispersal agent, or oil containment using a boom). However, these proposals all require the use of a communications infrastructure to facilitate message passing between agents. Our proposed method has the *significant benefit* of not requiring any such mechanism, relying only on local proximity detection.

The scenario is schematically depicted in Figure 8; we have an oil slick in some environment, and a swarm of robots

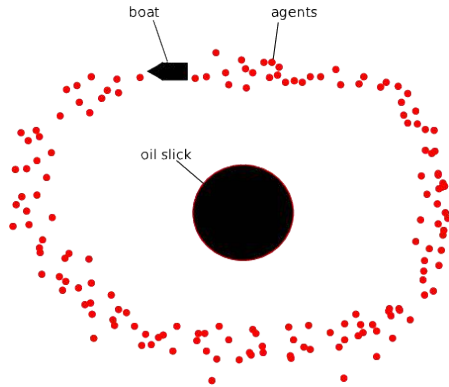


Figure 8: Oil slick containment scenario.

that are deployed by boat around the perimeter of the slick. Figure 9 shows the results of simulating the containment process using both the baseline movement algorithm (top) and the baseline method with void reduction (bottom). In our simulation, we use 200 agents, which is significantly greater than the number of agents than are generally simulated when inter-agent communication is required.

Without void reduction (i.e., simply using the field-effect-based movement algorithm) the swarm expands and then stabilises into a structure containing a void. The swarm “vibrates” slightly as cohesion and repulsion forces fluctuate to maintain the swarm’s structure, but the void does not fully close, and full and “tight” containment is not achieved. The agents that do come in contact with the obstacle are repelled by the obstacle repulsion field. If, however, we activate void reduction, then the swarm expands as expected, due to the field effects, but then the void is completely removed, achieving full and close containment of the slick.

Figures 10 and 11 show the effect of the void reduction on the distribution of agents compared to the baseline method. Figure 10 shows the distance distribution of the swarm for both the baseline method (grey/black) and the void reduction method (red). The baseline swarm initially expands, then settles after approximately 6 seconds (this is also the case for the void reduced swarm). Following the initial expansion, the baseline swarm remains relatively slow-changing with respect to distance and magnitude. However, the void reduced swarm is affected more significantly; after approximately 10 seconds the swarm’s internal void perimeter makes contact with the oil spillage (obstacle). This has the effect of disrupting the average distance and average *inter-agent magnitudes*. This effect diminishes slightly after approximately 18 seconds, when the swarm’s *void reduction vectors* cause the swarm to surround the spillage. The slick surrounding process is followed by a few remaining changes caused by the “snapping” of agents at the spillage perimeter, and then the containment process is complete.

Figure 11 compares *inter-agent magnitudes* for the base-

line and void reduction swarms. When initially deployed, the swarm is so dense that the average *inter-agent magnitude* is negative, indicating a high level of expansion. Within 2 seconds the expansion has reached a point where the average magnitude is positive, indicating the swarm is cohesive. This means that the swarm will remain as a single entity, and therefore be capable of surrounding an object without breaking apart.

When the swarm shrinks to surround the obstacle, we see an erratic change in the number of perimeter agents. Figure 12 shows the number of perimeter agents over the duration of the simulation. We see that the baseline swarm perimeter size decreases steadily and then settles (the swarm has not enclosed the spillage). The perimeter count has settled, but, as shown in Figures 10 and 11, the agents are still moving (magnitude variance and magnitude >0); however, the movement does not affect the overall structure.

When the void reduction swarm encounters the obstacle at approximately 10s there is a change due to “snapping”, as the agents “fold” around the obstacle. Snapping is an oscillation of relations between four agents (Eliot, 2017). The perimeter size then continues to fall gradually as the void percolates out of the system. The perimeter size then stabilises as the slick obstacle is fully surrounded.

Conclusion

In this paper, we have shown how the structure of a simulated swarm of robots may be controlled by the identification and removal of perimeter anomalies. Importantly, the identification of anomalies is achieved locally by individual agents using only proximity detection, without any need for an inter-agent communication structure. This could offer significant benefits in terms of cost, simplicity, and fault-tolerance. The technique works with arbitrary-sized swarms; here we use 200 agents, but we have successfully simulated swarms of up to 500 agents with no appreciable performance degradation.

This work demonstrates one possible application of our void reduction technique. Future work will focus on its use with mobile swarms (e.g., for reconnaissance) which must navigate past/around a number of obstacles whilst maintaining a coherent and compact structure.

References

- Andreou, P., Zeinalipour-Yazti, D., Andreou, M., Chrysanthis, P. K., and Samaras, G. (2009). Perimeter-based data replication in mobile sensor networks. In *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 244–251. IEEE.
- Barnes, L., Alvis, W., Fields, M., Valavanis, K., and Moreno, W. (2006a). Heterogeneous swarm formation control using bi-variate normal functions to generate potential fields. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 85–94. IEEE.

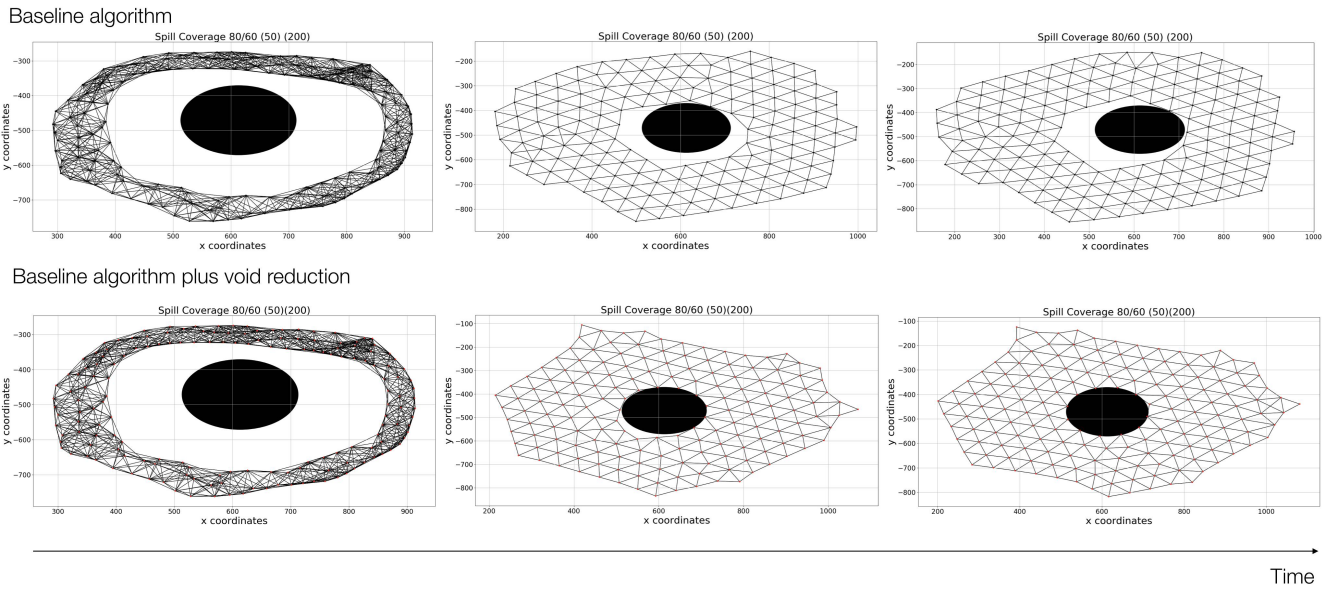


Figure 9: Simulation results for oil slick containment scenario. Top three frames show the evolution of the swarm using only the baseline movement algorithm; bottom three frames show the impact of adding our void reduction method.

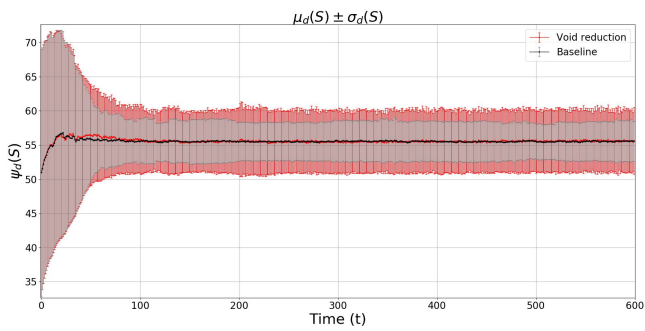


Figure 10: Oil spill containment distance (time shown in 10 millisecond slices).

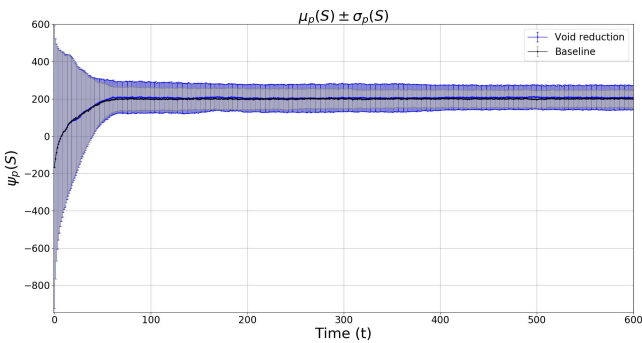


Figure 11: Oil spill containment magnitude (time shown in 10 millisecond slices).

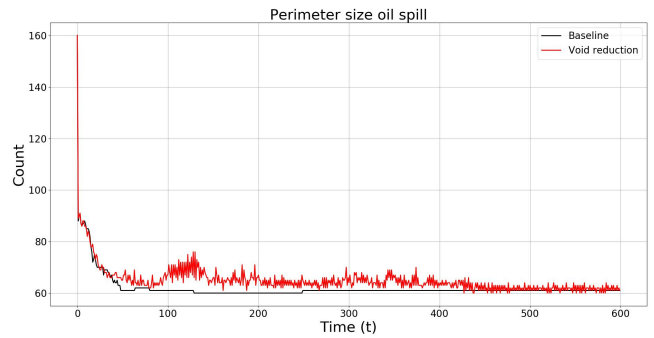


Figure 12: Swarm perimeter size comparison (time shown in 10 millisecond slices).

Barnes, L., Alvis, W., Fields, M., Valavanis, K., and Moreno, W. (2006b). Swarm formation control with potential fields formed by bivariate normal functions. In *14th Mediterranean Conference on Control and Automation*, pages 1–7. IEEE.

Bennet, D. and McInnes, C. (2009). Verifiable control of a swarm of unmanned aerial vehicles. *Journal of Aerospace Engineering*, 223(7):939–953.

Blum, C. and Merkle, D. (2008). *Swarm Intelligence: Introduction and Applications*. Springer.

Cao, Y., Ren, W., and Egerstedt, M. (2012). Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks. *Automatica*, 48(8):1586–1597.

- Dai, Y. S., Hinchey, M., Madhusoodan, M., Rash, J. L., and Zou, X. (2006). A prototype model for self-healing and self-reproduction in swarm robotics system. In *2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pages 3–10.
- Ekanayake, S. W. and Pathirana, P. N. (2010). Formations of robotic swarm: an artificial force based approach. *International Journal of Advanced Robotic Systems*, 7(3):173–190.
- Elamvazhuthi, K. and Berman, S. (2015). Optimal control of stochastic coverage strategies for robotic swarms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1822–1829. IEEE.
- Eliot, N. (2017). *Methods for the Efficient Deployment and Coordination of Swarm Robotic Systems*. PhD thesis, Northumbria University, UK. Available at <http://nrl.northumbria.ac.uk/32575/>.
- Eliot, N., Kendall, D., and Brockway, M. (2018). A new metric for the analysis of swarms using potential fields. *IEEE Access*, 6:63258–63267.
- Fritsch, D., Wegener, K., and Schraft, R. D. (2007). Control of a robotic swarm for the elimination of marine oil pollutions. In *IEEE Swarm Intelligence Symposium*, pages 29–36. IEEE.
- Gazi, V. and Passino, K. M. (2002). Stability analysis of swarms in an environment with an attractant/repellent profile. In *Proceedings of the American Control Conference*, volume 3, pages 1819–1824. IEEE.
- Gazi, V. and Passino, K. M. (2004a). A class of attractions/repulsion functions for stable swarm aggregations. *International Journal of Control*, 77(18):1567–1579.
- Gazi, V. and Passino, K. M. (2004b). Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):539–557.
- Gazi, V. and Passino, K. M. (2005). Stability of a one-dimensional discrete-time asynchronous swarm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):834–841.
- Gazi, V. and Passino, K. M. (2011). *Swarm Stability and Optimization*. Springer Science & Business Media.
- Gurfil, P. and Kivelevitch, E. (2007). Flock properties effect on task assignment and formation flying of cooperating unmanned aerial vehicles. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 221(3):401–416.
- Hashimoto, H., Aso, S., Yokota, S., Sasaki, A., Ohya, Y., and Kobayashi, H. (2008). Stability of swarm robot based on local forces of local swarms. In *SICE Annual Conference, 2008*, pages 1254–1257. IEEE.
- Hereford, J. (2011). Analysis of beelust swarm algorithm. In *IEEE Symposium on Swarm Intelligence*, pages 1–7.
- Hou, S. P., Cheah, C. C., and Slotine, J. J. E. (2009). Dynamic region following formation control for a swarm of robots. In *IEEE International Conference on Robotics and Automation*, pages 1929–1934.
- Ismail, A. R. and Timmis, J. (2010). Towards self-healing swarm robotic systems inspired by granuloma formation. In *IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 313–314. IEEE.
- Jung, S.-Y. and Goodrich, M. A. (2013). Multi-robot perimeter-shaping through mediator-based swarm control. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6. IEEE.
- Kakalis, N. M. and Ventikos, Y. (2008). Robotic swarm concept for efficient oil spill confrontation. *Journal of Hazardous Materials*, 154(1-3):880–887.
- Lee, G. and Chong, N. Y. (2008). Self-configurable mobile robot swarms with hole repair capability. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1403–1408.
- McLurkin, J. and Demaine, E. D. (2009). A distributed boundary detection algorithm for multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4791–4798. IEEE.
- Mohan, Y. and Ponnambalam, S. (2009). An extensive review of research in swarm robotics. In *World Congress on Nature and Biologically Inspired Computing*, pages 140–145. IEEE.
- Navarro, I. and Matía, F. (2009). A proposal of a set of metrics for collective movement of robots. In *Proc. Workshop on Good Experimental Methodology in Robotics*.
- Roach, J. H., Marks, R. J., and Thompson, B. B. (2015). Recovery from sensor failure in an evolving multiobjective swarm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):170–174.
- Saldana, D., Ovalle, D., and Montoya, A. (2012). Improved algorithm for perimeter tracking in robotic sensor networks. In *Conferencia Latinoamericana En Informatica (CLEI)*, pages 1–7.
- Timmis, J., Ismail, A., Bjerknes, J., and Winfield, A. (2016). An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems*, 146:60–76. Information Processing in Cells and Tissues.
- Vassev, E. and Hinchey, M. (2009). ASSL specification and code generation of self-healing behavior for nasa swarm-based systems. In *IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, pages 77–86.
- Zhang, G., Fricke, G. K., and Garg, D. P. (2013). Spill detection and perimeter surveillance via distributed swarming agents. *IEEE/ASME Transactions on Mechatronics*, 18(1):121–129.

The ARE Robot Fabricator: How to (Re)produce Robots that Can Evolve in the Real World

Matthew F. Hale^{1*}, Edgar Buchanan^{2*}, Alan F. Winfield¹, Jon Timmis², Emma Hart⁴, Agoston E. Eiben³, Mike Angus², Frank Veenstra⁴, Wei Li², Robert Woolley², Matteo De Carlo³ and Andy M. Tyrrell²

¹Bristol Robotics Laboratory, University of the West of England, Bristol, UK

²Department of Electronic Engineering, University of York, UK

³Department of Computer Science, Vrije Universiteit Amsterdam, NL

⁴School of Computing, Edinburgh Napier University, UK

matt.hale@brl.ac.uk, edgar.buchanan@york.ac.uk *These authors contributed equally to this paper

Abstract

The long term vision of the Autonomous Robot Evolution (ARE) project is to create an ecosystem of both virtual and physical robots with evolving brains and bodies. One of the major challenges for such a vision is the need to construct many unique individuals without prior knowledge of what designs evolution will produce. To this end, an autonomous robot fabrication system for evolutionary robotics, the *Robot Fabricator*, is introduced in this paper. Evolutionary algorithms can create robot designs without direct human interaction; the Robot Fabricator will extend this to create physical copies of these designs (phenotypes) without direct human interaction. The Robot Fabricator will receive genomes and produce populations of physical individuals that can then be evaluated, allowing this to form part of the evolutionary loop, so robotic evolution is not confined to simulation and the reality gap is minimised. In order to allow the production of robot bodies with the widest variety of shapes and functional parts, individuals will be produced through 3D printing, with prefabricated actuators and sensors autonomously attached in the positions determined by evolution. This paper presents details of the proposed physical system, including a proof-of-concept demonstrator, and discusses the importance of considering the physical manufacture for evolutionary robotics.

Introduction

This paper outlines a long-term vision towards robots that reproduce and evolve in real-time and real space as well as an ongoing research project concerned with the first tangible implementation of such robots; evolving physical robots will be a significant step towards robotic artificial life. Specifically, we discuss the challenge of robot (re)production and present our first results with the ‘Robot Fabricator’ for autonomously producing robot phenotypes.

The long-term vision behind this research has a two-fold motivation. Firstly, it is to create a new type of artificial evolutionary systems that depart from the evolution of digital artefacts—Evolutionary Computing—and realizes the evolution of physical artefacts: the Evolution of Things as introduced in (Eiben et al., 2012). Such systems will represent a third incarnation of Darwinian principles. To date, these principles can be observed and studied in wetware (Life on

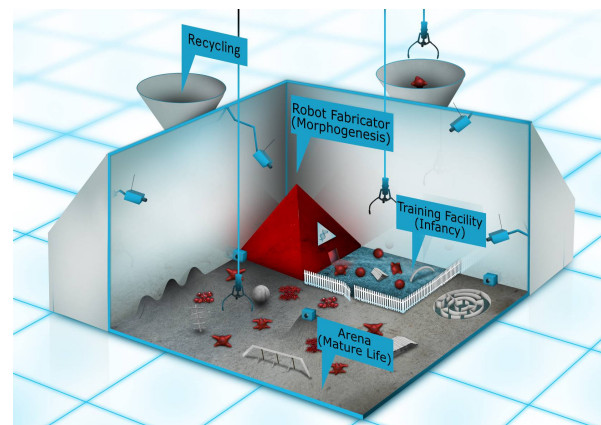


Figure 1: Illustration of the ARE environment, showing the three main stages of the Triangle of Life model and a recycling facility.

Earth) and software (Evolutionary Computing); the Evolution of Things will realize them in hardware, cf. Eiben and Smith (2015). Such hardware models of evolution will facilitate fundamental research into, for instance, the macro-level mechanisms of evolution, the emergence of (embodied) intelligence, and the simultaneous evolution of the body and the brain without suffering from the infamous reality gap (Jakobi et al., 1995).

Secondly, evolving robots is interesting from an engineering perspective. The fact that Life on Earth has populated practically all possible environmental niches demonstrates that natural evolution is very successful in producing specialised life forms. Hence, it is a reasonable hypothesis that artificial evolution will be capable of producing specialised robots for various environments and tasks. Furthermore, an autonomously evolving robot population has the ability to adapt to previously unknown and/or changing conditions, thus creating new types of machines that are able to adapt their form and behavior.

The field of Evolutionary Robotics has addressed the evolution of robot controllers (brains) with considerable suc-

cess but evolving the morphologies (bodies) has received much less attention (Nolfi et al., 2000, 2016; Doncieux et al., 2015). This is somewhat understandable, given the difficulty of implementing such systems in the real world, i.e. the lack of technologies for automated (re)production of robots. However, advances in robotics, 3D-printing, and automated assembly mean it is now timely to work on physical robot systems with evolvable morphologies (Winfield and Timmis, 2015). The Autonomous Robot Evolution (ARE) project¹ is concerned with developing the first such system, illustrated as a concept in Figure 1. The work will allow for radically new autonomous systems, where robots are designed and manufactured by algorithms and machines rather than by humans.

The main contribution of this paper is the proposal of a novel approach to robotic evolution, where an automated facility enables the (re)production of robot populations, and thus robot evolution in the physical world. After discussing previous related work on evolving physical robot morphologies, the paper outlines the overall system architecture for the ARE project. This is followed by a more detailed description of the required physical infrastructure, focusing on the proposed Robot Fabricator system (RoboFab) to achieve an automated assembly process, with a proof-of-concept demonstration presented.

Related Work

As of today there have been very few examples of autonomous robot fabrication for evolutionary robotics in the literature. Most approaches involve evolving the robots in different simulators and manually assembling the resulting robots. In addition, those robots that are autonomously assembled have generally been very simple with a single type of actuator and no sensors.

The important link between morphology and control (Pfeifer and Bongard, 2007) suggests that they should be evolved together in order to unlock the potential of evolutionary robotics; however since the breakthrough work of Sims (1994), the majority of brain and body evolution has occurred in simulation.

A logical progression from simulation studies has been the simulate-then-transfer paradigm, such as the Golem project (Pollack and Lipson, 2000; Lipson and Pollack, 2000), in which the evolution of simple robots (without sensors and with fixed controllers) occurred in simulation after which a select few individuals were physically manufactured. However this does not allow for any selection based on the physical robots, and so evolution stops at the point of physical manufacture. Although this paradigm has been shown to be successful for problems such as antenna design (Hornby et al., 2011), the complex interactions of mobile robots with their environment are difficult to simulate

¹see <https://www.york.ac.uk/robot-lab/are/>

accurately, leading to the reality gap (Jakobi et al., 1995), which could be bypassed by evaluating robots in the physical world.

Evolution of controllers where virtual models of the robots are updated according the performance of physical robots has been achieved previously, e.g. (Bongard and Lipson, 2004; Hwangbo et al., 2019).

One approach for creating populations of physical robots may be through self-replicating robots. Zykov et al. (2005, 2007) demonstrated modular robots capable of such reproduction. However the physical robots only created exact copies of themselves, without variation and therefore without evolution in hardware.

There has been some progress toward a fully automated system for creating physical robot phenotypes (which in ARE is termed the Robot Fabricator). Brodbeck et al. (2015) used a “mother robot” (a robot arm) to create physical individuals by gluing cuboid modules together. They demonstrated model-free morphological evolution, with 100 individuals for each evolutionary run physically created and tested; a major breakthrough. However, the robots used were limited to a single type of actuator (servo-motor) and the robots had no sensors—evolving more complex morphologies is likely to present further challenges.

3D printing is a rapidly developing field, offering exciting new opportunities to create complex robotics systems with minimal manual intervention (MacCurdy et al., 2016), with a particular interest in printed soft robots (Hiller and Lipson, 2012; Bartlett et al., 2015). However, 3D printing cannot yet be used to create mechatronic components, making essential robot functions of sensing and actuation impossible. This motivates a combined approach in the ARE project, where some components are 3D printed (allowing arbitrary shapes), while sensor and actuator ‘organs’, with electronics and motors, are hand designed and built.

The Triangle of Life concept (discussed below) has been demonstrated with a simplified setup by Jelisavcic et al. (2017), where physical robots were 3D-printed and hand-assembled, also using prefabricated organ, such as servo motors and Raspberry Pis. The ARE project will take this further through the automated Robot Fabricator, allowing larger numbers of individuals and a complete evolutionary system to be fabricated.

Overall System Architecture

A general architecture for evolving robots in real time and real space has been suggested in the conceptual framework named the Triangle of Life by Eiben et al. (2013), shown in Figure 2. A real-world implementation of this is envisaged by the notion of an EvoSphere as introduced and extensively discussed in Eiben (2015) and modified for the ARE project in Figure 1; this forms a design template for an evolutionary robot habitat and provides the basis of the physical environment in the ARE project.

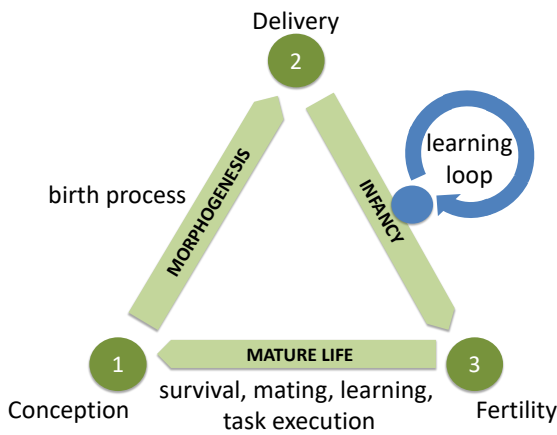


Figure 2: Generic system architecture for robot evolution conceptualized by the Triangle of Life. The learning methods in the Infancy stage are not necessarily evolutionary.

The Triangle of Life consists of three stages: morphogenesis, infancy, and mature life, as illustrated in Figure 2. Consequently, an EvoSphere consists of three components. The Robot Fabricator is where new robots are created (morphogenesis). The Training Facility provides a suitable environment for individuals to learn during infancy, providing feedback, perhaps via a computer vision system and/or a human user, so individual robots can learn to control their (possibly unique) body to perform some simple tasks. The Training Facility increases the chances of success in the Arena and plays an important role: it prevents reproduction of poorly performing robots and saves resources. It also enables refinement of controllers learned in a simulated environment that do not transfer properly due to the reality gap. If a robot acquires the required set of skills, it is declared a fertile adult and enters the Arena, which represents the world where the robots must survive and perform user-defined tasks, and may be selected for reproduction. The selection mechanism can be innate in the robots (by choosing “mates”) or executed by an overseer, which can be algorithmic or a human “breeder”.

An essential feature of the EvoSphere concept and the ARE system is the centralised, externalised reproduction. For reasons of ethics and safety we reject distributed reproduction systems, e.g. self-replicators or the robotic equivalents of cell division, eggs, or pregnancy, and deliberately choose for one single facility that can produce new robots. This facility, the Robot Fabricator, serves as an emergency switch that can stop evolution if the users deem it necessary.

The ARE system features deep integration of virtual and physical robot evolution. In essence, there are two concurrently running implementations of the Triangle of Life, one in a virtual environment and one in the physical environment. The Ecosystem Manager is a program to control the hybrid physical-virtual system, providing the link between

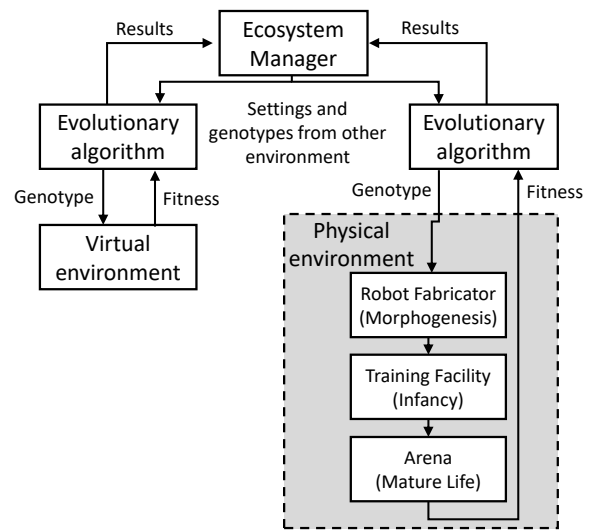


Figure 3: Diagram of the ARE system components: the virtual environment, the physical environment, and the Ecosystem Manager. Simulated and physical evolution are running concurrently, supervised and controlled by the Ecosystem Manager.

the two, as illustrated in Figure 3. This integration is made possible by using the same genetic representation in both worlds, enabling cross-breeding so a new robot in either environment could have physical or virtual parents, or a combination of both. An individual can also be copied between environments simply by transferring its genotype from one environment to the other. This integration must be controlled by the Ecosystem Manager, which optimises the working of the hybrid evolutionary system, maximising task performance. It reacts to flows of information from both subsystems, and to human-specified goals, either hand-directed or running autonomously.

This integration seeks to combine the advantages of the real and virtual worlds. Physical evolution is accelerated by the virtual component that can find good robot features with less time and resources than physical evaluation, while simulated evolution benefits from the influx of genes that are tested favourably in the real world. This means while a single physical robot is evaluated in the physical evolution, hundreds of simulated generations will be evaluated at the same time. As consequence of this a huge proportion of the population will be virtual. In order to compensate for this we will give greater weighting to results obtained from physical robots.

It is important to mention that the size of the population in the physical world will dynamically change over time, as new robots are added (from the virtual world or randomly created) and due to physical robots malfunctioning or being removed due to poor performance.

The evolutionary algorithms are kept separate, as shown in Figure 3, to allow for different configurations (of population size, mutation rate etc.) suited to these different environments, with settings dictated by the Ecosystem Manager.

Autonomous Manufacturing in the Physical Environment

This section describes the proposed method for the physical manufacture of robots in the context of the ARE project.

The *Robot Fabricator* is a system to automate the process of morphogenesis: the conversion of genotypes into physical robot phenotypes, as outlined in Figure 3. The final objective is to remove human intervention from this process as far as possible, creating a manufacturing system for the autonomous production of complete robots, which will allow the evaluation of physical individuals to form part of the evolutionary process.

The proposed design for the Robot Fabricator is described below.

The *organs* are defined as active components which individual robots can use to perform their task(s). It is expected each robot will have a single “brain” organ, which contains the electronic control hardware and a battery to provide the robot with power. Other organs provide sensing and actuation, with their type and location being specified in the genome written by the evolutionary algorithm. These other organs will typically each comprise a single sensor or actuator, such that evolution is free to select the quantity and arrangement of the active parts of the robot. In practice, once a robot is no longer needed, the physical organs can be removed and recycled to create new individuals. Examples of physical and virtual organs are shown in Figures 4 and 5.

The *skeleton* is the structure that physically connects the organs together. Specifically, the skeleton will be comprised of thermoplastic and made by additive manufacture (3D printing). The skeleton is generally expected to be non-functional except for serving as a scaffold to hold the organs in place relative to each other.

In this way, the robots created will feature organs which are designed to be re-usable for many different designs and the skeleton, which will be made specifically for a particular individual. This organ and skeleton approach is flexible enough to allow for a wide range of sensors and locomotion methods, including wheel organs and/or joint organs for constructing robots with limbs. Furthermore, certain organs could also be designed such that some part(s) can be individually 3D printed and attached autonomously at the time of assembling the robot, for instance the radius (or even the shape) of wheels, or the fingers of a gripper, so that these aspects of the organs can also be evolved. As such, the Robot Fabricator is intended to allow for experiments where the search space can be large and diverse enough for fundamental research into evolutionary robotics (and perhaps evolution in general), and the robots capable enough to perform



Figure 4: Prototype organs in the physical world. From left to right: a sensor organ, brain organ and wheel organ.



Figure 5: Models of the organs from Figure 4 for use in the virtual environment.

useful and interesting tasks.

One potential ethical issue is the waste produced by large numbers of bespoke plastic skeleton parts, which cannot be re-used. To minimise the environmental impact of this, the material chosen is a plant-based polymer (PLA), which is recyclable and biodegradable. After the useful life of an individual, the organs will certainly be removed and re-used for future individuals; it may also be possible to melt down the skeleton parts and make new filament, to be used for new individuals.

The Robot Fabricator System

The skeleton, which holds the organs in their positions, needs to take an arbitrary shape depending on the organ locations specified. To achieve this, it is made using a 3D printer (LulzBot TAZ 6) by the Fused Deposition Modelling (FDM) approach to form 3D shapes from a thermoplastic (polylactic acid, commonly known as PLA).

The organs, in comparison, are much more complex, as they require electrical and electronic components and many different materials to allow for a range of actuation and sensing technologies; these will be prefabricated and attached to the skeleton using a multi-axis manipulator (a robot arm, in this case a Universal Robots UR5e).

To allow the robot arm to easily pick up each required organ, they are stored in an *organ bank*, where each organ is held in a known and accessible position.

While the organs are attached, the semi-constructed robot

must be held securely, but then must be released once the assembly is complete. This will be achieved by a fixture in the *assembly area* which can mate to a feature on the bottom face of the brain organ which forms the core of each individual.

The 3D printing of the skeleton is likely to take several hours, much longer than the assembly, and so the production rate can be increased by having multiple printers operating in parallel, tended to by a single robot arm. A proposed layout for these components is shown in Figure 6.

To demonstrate some of the key steps in this process, a test setup has been created. This process is depicted in Figure 7 and the supplementary video². The manufacturing sequence is described as follows:

1. The Robot Fabricator receives the required coordinates of the organs and one or more mesh file(s) of the shape of the skeleton.
2. The skeleton is produced by the 3D printer (Figure 7-1).
3. The robot arm transfers the skeleton to the assembly area. This is currently done manually (Figure 7-2).
4. The organs are attached to the appropriate locations by the robot arm (Figure 7-3).
5. The robot arm connects the organs together by cables to provide power and communications (Figure 7-4).
6. Any remaining organs that cannot be attached by the robot arm, such as wheels, are attached manually (Figure 7-5). In the final design this step will be eliminated.
7. The robot is complete and ready to be transferred to the Training Facility (Figure 7-6).

From a Virtual Robot to a Physical Robot

The robot morphologies shown in this paper were initially evolved in the simulations using a steady state evolutionary algorithm (EA), i.e. with overlapping generations. Traveling distance is the measure of fitness. The parameters values used for the EA are the following: the population size of 20, generation number of 200 and mutation rate of 0.2. An example of equivalent virtual and physical phenotypes is shown in Figure 8.

The robots shown in this paper were generated with direct encoding. The position and orientation of each organ is explicitly specified in the genome. The organs are connected directly to the brain with vertical and horizontal segments. This representation is used because of its simplicity, but further work will evaluate and compare other methods. The code that generates such morphologies can be found in the supplementary material of this paper.

Once the best individual is found, the list of coordinates of each organ is sent to the Robot Fabricator together with the mesh file that represents the shape of the body. The mesh file is used to create the skeleton with a 3D printer and the

list of coordinates is used to attach the organs to the body using the robot arm.

Even though a large diversity of robots with different shapes can be generated with evolutionary algorithms, not all the robots can be built. The practical limitations of the Robot Fabricator impose various constraints on what can be physically produced. Therefore, before a genome is sent to the Robot Fabricator, it is very important to make sure that its phenotype can be manufactured. This evaluation is described below.

Viability Test

Before a genome is sent to the Robot Fabricator, it is subject to a *viability test* to make sure that its phenotype can be manufactured. Only viable robots will be produced, to avoid wasting time and resources attempting to manufacture only to find it cannot be made. Trying to produce a non-viable individual may also risk damage being caused to the organs or Robot Fabricator.

The viability test must check for violation of the limitations of the Robot Fabricator manufacturing process, which are constraints that would not exist for simulation, and would change depending on the assembly process chosen, such as those in Table 1.

The viability test may optionally be extended to cover not only individuals that *cannot* be manufactured, but also detect some cases of individuals with no chance of a decent performance, and therefore *should* not be manufactured. This could avoid wasting resources and speed up the overall evolution. For example, this could be a test of fundamental functionality—if a robot must sense its surroundings and act upon the information received, then it must possess at least one sensor and one actuator. Whether this type of addition to the viability test does significantly speed up evolution, or if they have negative side-effects, is an area to be explored.

In this paper we present some examples of robots generated with and without viability test from evolution. Robots are subjected to a series of checks when the the viability test is enabled, shown in Table 1; if a robot fails one of the checks it is considered as non-viable and a fitness of zero is assigned to this robot. All the checks are ignored and all the robots are considered viable when the viability test is disabled. An example of a robot that passes the viability test is shown in Figure 9.

Without the restrictions of the viability test, evolution will exploit any means of increasing the fitness function, which may not result in robots that can be manufactured. Figure 10 shows some examples of robots that evolved which would have failed the test, selected by manually inspecting the final generation to demonstrate different issues. Figure 10(a) shows an example of overlapping organs (although in this case a high fitness is achieved through a simulation bug, and is not the intended target of the viability test). With no limits to how many organs a robot can have, evolution can add

²Supplementary material available at: <https://www.york.ac.uk/robot-lab/are/alife2019/>

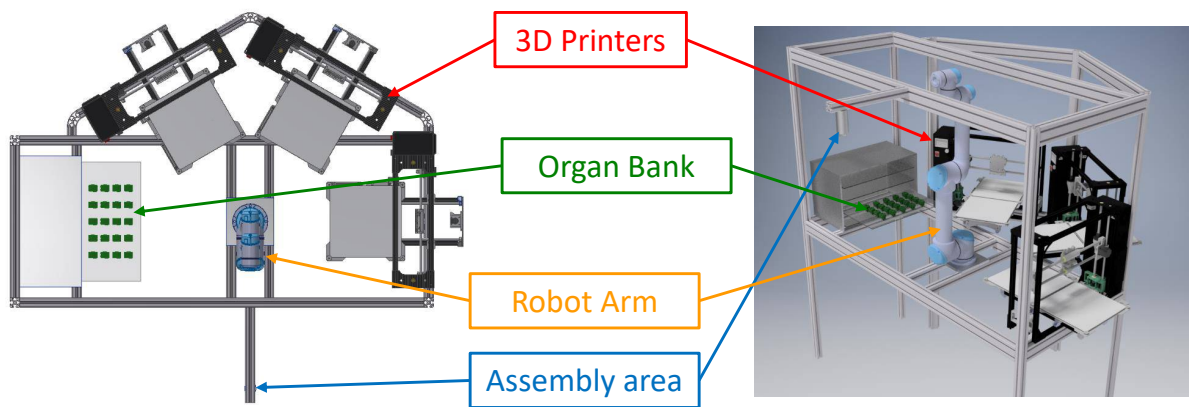


Figure 6: Concept layout for the final Robot Fabricator design to facilitate automated production of complete robots. The skeleton is printed on one or more printers, and assembled by the robot arm together with organs from the organ bank.

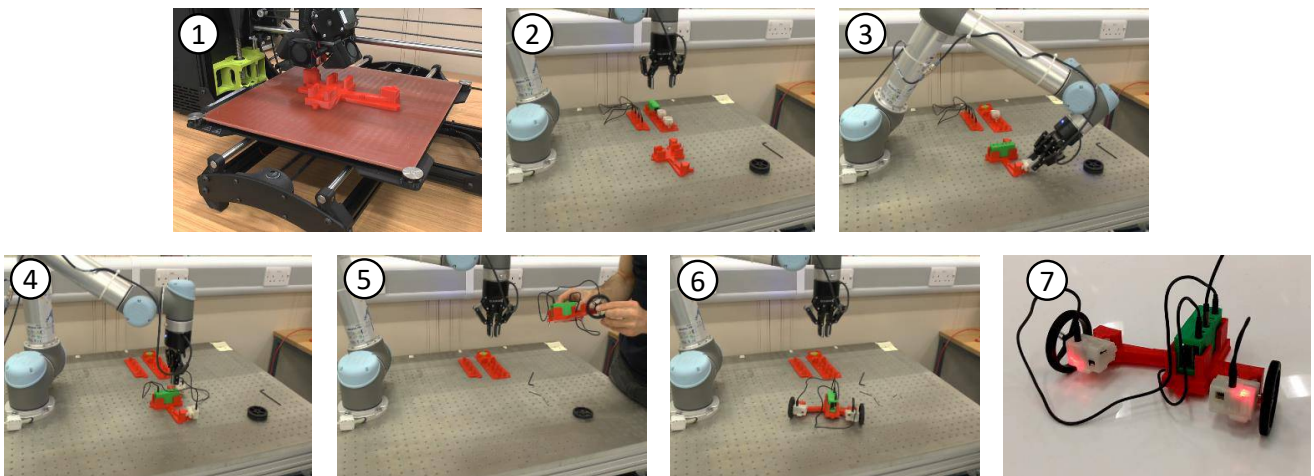


Figure 7: Robot production: (1) skeleton is 3D printed, (2), manual transfer to assembly area, (3) organs attached, (4) cables connected between organs, (5) manual wheel attachment, (6) robot is finished and (7) is tested. These images are from a video which is available online at www.york.ac.uk/robot-lab/are/alife2019/.

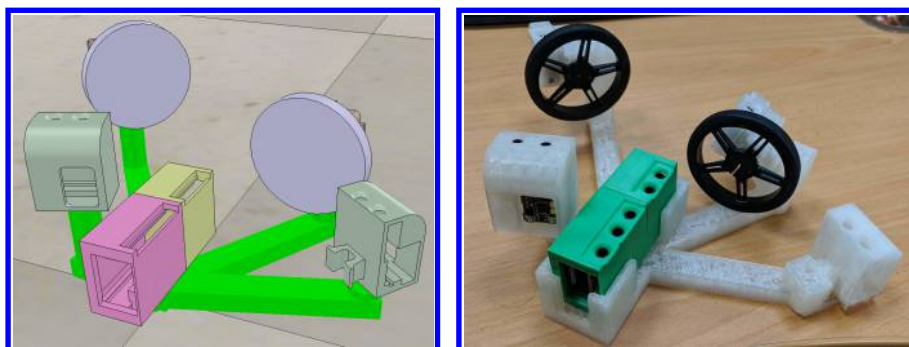


Figure 8: Equivalent virtual and physical phenotypes.

Constraint	Description
3D printer volume	The maximum size of a 3D-printed part (<i>i.e.</i> a piece of skeleton) is limited by the print volume of the 3D printer. This does not prohibit several parts (<i>e.g.</i> limbs) being made separately and then joined to form a complete robot larger than this.
Assembly area size	The overall size of any robot, after all the organs and limbs are connected, will be limited by the physical size of the assembly area.
Overlapping organs	In simulation organs can overlap and share space. In the physical world, this cannot be achieved.
3D Printing overhangs	3D printing works by building in layers; each layer must be supported by the layer below so there is a limit to the angle by which a face can overhang; additional support material would be difficult to remove autonomously.
Organ attachment and connections	The robotic arm requires access to attach each organ, so there must not be any 3D printed skeleton material, or a previously attached organ, in its path. Equivalent access is also required for each cable connection.
Number of Organs	The organs must be pre-made, and therefore there will be a fixed number available, limiting the quantity of organs that can be assigned to each individual.

Table 1: Manufacturing constraints to be considered by the viability test.

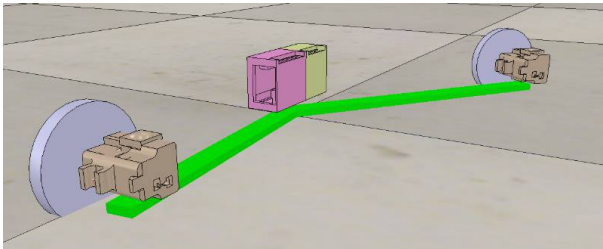


Figure 9: Robot evolved with viability test. This robot meets the physical constraints.

more than are available to the Robot Fabricator, such as the robot shown in Figure 10(b). In a similar way, evolution can exploit the size of the robot, resulting in skeleton sections larger than the print volume of the Robot Fabricator, as with the robot shown in Figure 10(c). Of course, some of these issues could be addressed by manipulating the fitness function or constraining the range of parameters within the genome, but it remains clear that there will be robots that are viable in a simulator but that cannot be manufactured for multiple reasons, therefore the viability test is essential.

It is worth mentioning that assigning a fitness of zero to a robot failing the viability test is not the only treatment for non-viable robots. Each robot could be repaired, or each robot could be kept in the population with low fitness, as it may be beneficial to allow movement through infeasible regions of the search space to find regions of higher fitness.

Conclusions

The ARE project envisions an environment where autonomous systems (robots) are not designed by humans (or indeed designed at all), but are created through a series of steps that follow evolutionary processes. These robots will

be “born” through the use of 3D additive manufacturing, with novel materials and a hybridised physical-virtual evolutionary architecture. Newly created robots will learn in a safe and controlled environment where success will be rewarded. The most successful individuals will make available their genetic code for reproduction and for the improvement of future generations. Such a process may ultimately lead to a change in the way things are designed and manufactured.

This paper describes the first step towards this vision, creating the Robot Fabricator system in which physical individuals can be fabricated autonomously. This is a challenging task, but the Robot Fabricator operates within the constraints of current technologies, to create a feasible system for the automatic production of robot bodies, while maximising the diversity of possible morphologies. Automated manufacture of evolved robots in the real world will allow us to address interesting and important questions around morphological evolution in hardware, the reality gap, and how these systems can be implemented, with potential for fundamental advances in the field.

Acknowledgements

The work reported in this paper is funded by EPSRC under the ARE project: EP/R03561X, EP/R035679, EP/R035733.

References

- Bartlett, N. W., Tolley, M. T., Overvelde, J. T., Weaver, J. C., Mosadegh, B., Bertoldi, K., Whitesides, G. M., and Wood, R. J. (2015). A 3d-printed, functionally graded soft robot powered by combustion. *Science*, 349(6244):161–165.
- Bongard, J. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, pages 57–62.

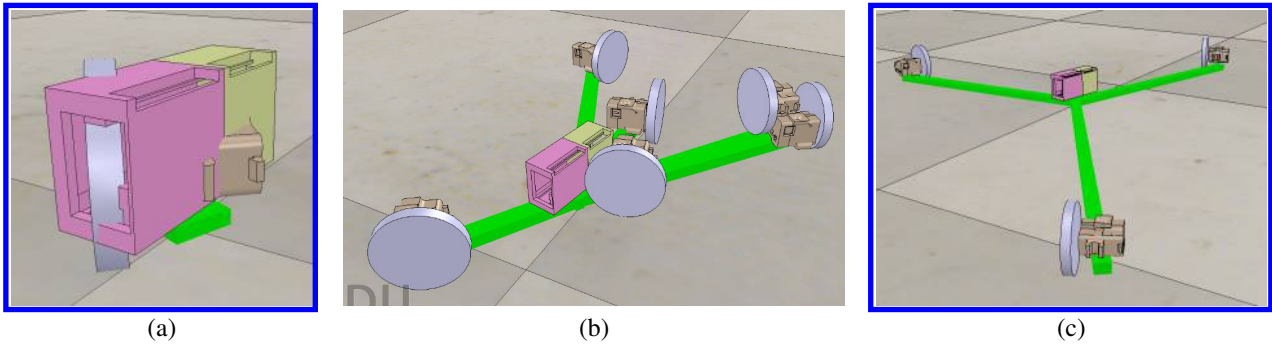


Figure 10: Examples of robots evolved without viability test: (a) robot with overlapping organs, (b) robot with high number of organs and (c) robot bigger than build volume.

- Brodbeck, L., Hauser, S., and Iida, F. (2015). Morphological evolution of physical robots through model-free phenotype development. *PLoS one*, 10(6):e0128444.
- Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. (2015). Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI*, 2:4.
- Eiben, A. (2015). EvoSphere: The World of Robot Evolution. In Dediu, A.-H., Magdalena, L., and Martín-Vide, C., editors, *Proc. of the Theory and Practice of Natural Computing 2015*, LNCS 9477, pages 3–19. Springer.
- Eiben, A., Bredeche, N., Hoogendoorn, M., Stradner, J., Timmis, J., Tyrrell, A., and Winfield, A. (2013). The triangle of life: Evolving robots in real-time and real-space. In Lio, P., Miglino, O., Nicosia, G., Nolfi, S., and Pavone, M., editors, *Proc. of the 12th European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)*, pages 1056–1063. MIT Press.
- Eiben, A., Kernbach, S., and Haasdijk, E. (2012). Embodied artificial evolution: Artificial evolutionary systems in the 21st century. *Evolutionary Intelligence*, 5(4):261–272.
- Eiben, A. and Smith, J. (2015). From evolutionary computation to the evolution of things. *Nature*, 521(7553):476.
- Hiller, J. and Lipson, H. (2012). Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466.
- Hornby, G. S., Lohn, J. D., and Linden, D. S. (2011). Computer-automated evolution of an x-band antenna for nasa’s space technology 5 mission. *Evolutionary Computation*, 19(1):1–23. PMID: 20583909.
- Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., and Hutter, M. (2019). Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872.
- Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer.
- Jelisavcic, M., De Carlo, M., Hupkes, E., Eustratiadis, P., Orłowski, J., Haasdijk, E., Auerbach, J. E., and Eiben, A. E. (2017). Real-world evolution of robot morphologies: A proof of concept. *Artificial life*, 23(2):206–235.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974.
- MacCurdy, R., Katzschmann, R., Kim, Y., and Rus, D. (2016). Printable hydraulics: A method for fabricating robots by 3d co-printing solids and liquids. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3878–3885. IEEE.
- Nolfi, S., Bongard, J., Husbands, P., and Floreano, D. (2016). Evolutionary robotics. In *Springer Handbook of Robotics*, pages 2035–2068. Springer.
- Nolfi, S., Floreano, D., and Floreano, D. D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press.
- Pfeifer, R. and Bongard, J. (2007). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Pollack, J. B. and Lipson, H. (2000). The golem project: Evolving hardware bodies and brains. In *Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware*, pages 37–42. IEEE.
- Sims, K. (1994). Evolution of Virtual Creatures. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*.
- Winfield, A. and Timmis, J. (2015). Evolvable robot hardware. In Trefzer, M. and Tyrrell, A., editors, *Evolvable Hardware*, pages 331–348. Springer.
- Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Self-reproducing machines. *Nature*, 435(7039):163–164.
- Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H. (2007). Evolved and designed self-reproducing modular robotics. *Robotics, IEEE Transactions on*, 23:308 – 319.

Reinforcement Learning Agents acquire Flocking and Symbiotic Behaviour in Simulated Ecosystems

Peter Sunehag, Guy Lever, Siqi Liu, Josh Merel, Nicolas Heess,
Joel Z. Leibo, Edward Hughes, Tom Eccles, Thore Graepel
DeepMind, London UK, sunehag@google.com

Abstract

In nature, group behaviours such as flocking as well as cross-species symbiotic partnerships are observed in vastly different forms and circumstances. We hypothesize that such strategies can arise in response to generic predator-prey pressures in a spatial environment with range-limited sensation and action. We evaluate whether these forms of coordination can emerge by independent multi-agent reinforcement learning in simple multiple-species ecosystems. In contrast to prior work, we avoid hand-crafted shaping rewards, specific actions, or dynamics that would directly encourage coordination across agents. Instead we test whether coordination emerges as a consequence of adaptation without encouraging these specific forms of coordination, which only has indirect benefit. Our simulated ecosystems consist of a generic food chain involving three trophic levels: apex predator, mid-level predator, and prey. We conduct experiments on two different platforms, a 3D physics engine with tens of agents as well as in a 2D grid world with up to thousands. The results clearly confirm our hypothesis and show substantial coordination both within and across species. To obtain these results, we leverage and adapt recent advances in deep reinforcement learning within an ecosystem training protocol featuring homogeneous groups of independent agents from different species (sets of policies), acting in many different random combinations in parallel habitats. The policies utilize neural network architectures that are invariant to agent individuality but not type (species) and that generalize across varying numbers of observed other agents. While the emergence of complexity in artificial ecosystems have long been studied in the artificial life community, the focus has been more on individual complexity and genetic algorithms or explicit modelling, and less on group complexity and reinforcement learning emphasized in this article. Unlike what the name and intuition suggests, reinforcement learning adapts over evolutionary history rather than a life-time and is here addressing the sequential optimization of fitness that is usually approached by genetic algorithms in the artificial life community. We utilize a shift from procedures to objectives, allowing us to bring new powerful machinery to bare, and we see emergence of complex behaviour from a sequence of simple optimization problems.

Introduction

Our natural world is the ultimate example of a self-organizing system (Ashby, 1947). Species and individuals

adapt to each other in competition and cooperation, often as predators and prey in food chains. One ubiquitous example of cooperative group behavior is flocking, which can be found on land, sea and air, and numerous benefits from flocking for both predators and prey have been discussed in the literature (Handegard et al., 2012; Ruxton, 2012). For instance, if predators are sparse, a flocked group of prey is not much more likely to be detected than any single individual. Thus, if the predator eliminates at most one individual per detection, it follows that fewer prey will be eaten if they stick together. Further, if the prey are collectively more likely to detect the predator and thus to avoid predation, this improves individual survival chances. Flocking is not only used by prey species but also by predators: it can enable predators to cut off escape routes for a group of prey (e.g. seatrout hunting juvenile gulf menhaden) (Handegard et al., 2012); enable species to jointly capture larger prey, e.g. humans hunting whales (Alvard, 2003); or reduce individual nutritional variability by sharing captures. A second example of group behavior are symbiotic partnerships between species, for example humming bird nests are safer from jay predation when a hawk, which threatens the jays, is situated on top of the same tree (Greeney et al., 2015).

We hypothesize that group strategies like flocking and symbiosis can result in response to very generic predator-prey pressures and opportunities in a spatial environment with range-limited sensation and action, and we test this hypothesis experimentally by deploying independent reinforcement learning (RL) agents in generic simulated ecosystems. RL agents, like e.g. genetic algorithms, learn across the full (evolutionary) history and not primarily during episodes (life-times) and is primarily here viewed as a powerful way to optimizing the sequence of optimization problems posed by the ecosystem including the policies of the other species at the relevant times.

Our environments have three trophic levels (prey, predators and apex predators) and thus enable the emergence of partnerships within and across species. Further, unlike prior work (Morihiro et al., 2006; Hung, 2015; Yang et al., 2018) we do not shape the dynamics, actions or rewards to specif-

ically encourage or facilitate particular group behaviors, Instead we show that such behaviors can emerge in simple general contexts. We use two simulation platforms for this study, to highlight the generality of the findings and show that implementation details are not important. First, the Mujoco physics engine (Todorov et al., 2012) in which RL agents control a spherical body with continuous steer and roll actions, which sense the position and velocity of nearby others and their own physical state. These agents are rewarded (or penalized) according to their proximity to prey (or predator) agents. Second, we use a 2D gridworld with partial observations (agents locally sense a window of pixels around themselves) as used in Leibo et al. (2017). The former environment allows richer movement patterns while being more challenging to learn in, while the latter allows for very large numbers of agents.

One choice to make in both platforms is the population size at each trophic level. We believe that flocking is more likely when there is a spatial concentration of the predators and/or prey of the species in question. For a species to seek protection from an apex predator we believe it must be hard to escape from its predators because of its density or abilities. Due to these considerations, we opt for a relatively large middle population while a few individual agents at the bottom level represent a large amount of food (e.g. a group of individuals). The apex predators will be the fewest in numbers but have the most impact (on reward).

We find that the agents of the middle trophic level learn a coordinated hunting (flocking) strategy that makes them more successful at hunting their prey. There are at least two benefits from collective hunting; reducing the set of escape trajectories for the prey and collective navigation including information gathering. Cross species collaboration patterns are one of the new possibilities that arise with more than two trophic levels. We observe this in both the 3D physics simulation and the 2D gridworld. The agents at the bottom of the food chain learn to seek out the top apex predator (the hawk) for protection (from jays) and even form a sort of “partnership”.

Further, to investigate if at a large scale, like Yang et al. (2018), we also see population dynamics of a form that in some ways resemble nature (e.g. oscillations around a mean), we introduce a variation with spawning and vanishing (from predation) agents. This enables the population levels to reflect the success of the species (policy). We observe several learning phases with lasting equilibria in between quicker changes when superior strategies are discovered. Within each episode, population levels fluctuate regularly around the average, which changes between episodes as all agents learn. Further, we are able to see the aforementioned group behaviours playing a pivotal role. Most interestingly, we see first the failure and then success of group defence without direct individual reward, and it is strengthened by a partnership.

In summary, in our food chain simulations we observe several instances of sophisticated spatial coordination strategies emerging without having shaped the environment dynamics or rewards. For example, we see flocking strategies for predators. While this kind of pattern was also seen recently by Yang et al. (2018) in a grid world, they relied on an explicit “join group” action and introduce prey explicitly requiring sufficiently large groups to hunt.

Related work

Besides the prior work that has been reviewed above, we here review further relevant literature in Reinforcement Learning, Ecology and Artificial Life.

Predator-Prey dynamics have been widely studied (Levin, 2009), both through data gathering in nature and with mathematical modelling and simulation (Harfoot et al., 2014). Often these models are defined at the population level and deal primarily with numbers or densities in an area. Also, research generally focuses on two trophic levels, a predator species (e.g. foxes) and its prey species (e.g. rabbits). A more intricate line of work (Fretwell, 1987) has considered three or more trophic levels, which permit *trophic cascade effects*. All of the above are explicit mathematical models and do not involve agents that learn.

A famous example of a trophic cascade is the green world hypothesis (Hairston et al., 1960), which explains the richness of plant life on earth as resulting from predation keeping herbivore population size in check. A more recently discovered example (Greeney et al., 2015) is the aforementioned partnership between humming birds and hawks. The natural world contains a tremendous diversity of other intelligent group strategies, including how ants search for food, which has inspired the ant colony optimization class of algorithms (Dorigo, 1992). Flocking has also inspired a long line of work for robot navigation (Reynolds, 1987) enabling drones with weak individual sensors to reach their target more robustly together. Other work has replaced the explicit flocking model with reinforcement learning in an MDP constructed so as to learn flocking (Morihiro et al., 2006), e.g. to fly a group of UAVs in formation to a location (Hung, 2015).

Artificial ecosystems have been studied for a long time (Conrad and Pattee, 1970; Packard, 1987; Ray, 1991; Hraber et al., 1994; Yaeger, 1993; Adami and Brown, 1994). Many of these do not contain an element of spatial navigation. Polyworld developed in Yaeger (1993) is the clearest example that does contain navigation in two dimensions. However, none of these works have reported the emergence of flocking behaviour. Although symbiosis has been a possibility from the earliest models (Conrad and Pattee, 1970), interactions between individuals has not been a main concern (Pachepsky et al., 2002). In their continuation, such as Lenski et al. (2003); Yaeger (2009), these lines of work focused more on the evolution of individual complexity. In the area of artificial life, work on swarm intelligence (Bonabeau

et al., 1999) usually involves explicit models as reviewed above. Further, in this area as in this article, work has also focused on generic ecosystems aiming to capture essentials and not biological specifics (Bedau, 2007). Many of these works focus on genetic algorithms due to them being inspired by genetic evolution, while we shift the attention to the sequence of optimization problems addressed by those procedures, and deploy state-of-the-art deep reinforcement learning that have seen much recent success.

Simple predator-prey inspired environments have also been used as test problems for multi-agent reinforcement learning, but not in the same way as explored here. For example, Lowe et al. (2017) model two trophic levels and do not closely investigate the solution strategies, instead comparing algorithms based on accumulated reward.

Reinforcement Learning in Ecosystems

As is common in RL (Sutton and Barto, 1998), we rely on an agent-environment framework (Russell and Norvig, 2010) where an agent interacts sequentially with an environment over a sequence of time steps. The agent selects actions and the environment returns observations and rewards. The agent’s performance is measured by cumulative reward, possibly using a discount to encode a preference for the near-term. *Multi-agent reinforcement learning* (MAREL) models a collection of agents interacting with an environment and learning, from these interactions, to optimize individual cumulative reward. MAREL is typically modelled as a *Markov game* (Littman, 1994). The special case of a Markov game with one agent is a partially-observed Markov decision process (POMDP) (Sutton and Barto, 1998).

The 3D-physics based environment proposed here, features continuous actions, but discrete time. The environment has underlying smooth (classical physics) dynamics with continuous time. 3D physics and continuous control of forces, provide a rich world allowing for more realistic and explicit behaviours, but can be more difficult to learn. For example, it requires a long sequence of actions to perform an apparently simple maneuver.

Ecosystem training We use an ecosystem training (Figure 1) approach where we keep three species (sets of policies), one set each for prey, the predators and apex predator, and for each episode we create a habitat by sampling one policy from each species and use it for all the relevant players. We do this in many parallel threads. Hence, at all times experience is gathered for each policy in many different combinations and for several instances of itself in each ongoing episode. The experiences are gathered and sampled from for each policy, which is learned independently through updates performed to its network weights using state-of-the-art RL algorithms; Maximum a-posteriori Policy Optimization (MPO) (Abdolmaleki et al., 2018) for the continuous case and Impala (Espeholt et al., 2018).

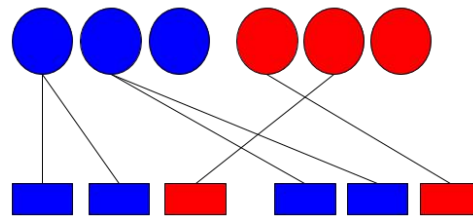


Figure 1: Simple example of ecosystem training: Sampling from two species (blue and red) of three policies (here circles) each, to be placed in habitats (here row of three rectangles) for two identical blue and one red.

A Physics-Based Food Chain Environment

We introduce a physics based food chain environment (see Figures 2 and 4) and describe its observations and rewards, as well as how agents process these and learn in an ecosystem training framework with three species. We base our environment on the MuJoCo physics engine (Todorov et al., 2012), utilized in much recent continuous reinforcement learning work including Brockman et al. (2016); Heess et al. (2017); Bansal et al. (2017); Abdolmaleki et al. (2018). In this environment, each agent controls a sphere with a two dimensional action space; acceleration forward/backward and rotational to steer. We have three different roles in the environment; apex predator, predator and prey, so we have three agent types or species. For visualization we render each agent type with different colors; green(pre), blue(predator) and red(apex predator). These spheres travel on a square floor bounded by walls on each side. The environment further contains two large square blocks which serve as physical barriers and introduce additional structure in the environment. The predators always spawn randomly within a large square in the middle. The apex predator and prey spawns according to two equally likely patterns. In the first they spawn in the same central square as the predator, or they each spawn independently in (a square in) a uniformly random corner. Both spawn patterns are displayed in accompanying videos¹ and are simply chosen to force the learning of varied behaviours, but are not designed to generate any specific outcome. In one, the predators (and the apex predator) has to start with searching, in the other the prey has to start with escaping.

Proximity based rewards The agents receive rewards based on proximity to other agents. Predators receive positive reward for being near prey while prey agents receive negative reward. Similarly, the apex predator receives a positive reward when it is sufficiently close to a predator agent which receives a negative reward in turn. The reward function is only dependent on distance between the agents and

¹<https://docs.google.com/presentation/d/1u86oapziZ35MfphcrIC3zbMMg9It-Bhf6fJEqyoeBwg/edit?usp=sharing>

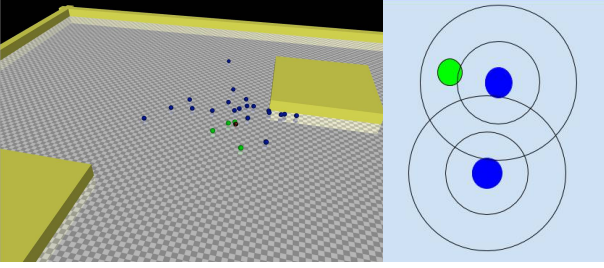


Figure 2: Physics based ecosystem environment. Left: An example of close red-green partnership in the middle of a large floor. Right: An illustration of two predators (blue) with a larger conspecific radius and a smaller by which it can see the prey (green).

we choose a sigmoid with a cut-off threshold. We define,

$$\phi(d) = \begin{cases} 1 - \tanh(0.5d) & \text{if } d < \text{radius} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and if d_i is the distance (at time t) from a certain predator agent to the prey agent i , and if e_j is the distance to the apex predator agent j , then this predator agent receives the instantaneous reward $\sum_i \phi(d_i) - 2.5 \sum_j \phi(e_j)$. The factor 2.5 for the term that represents being predated on by the apex predator, is to not make being close to one or two prey more important than avoiding the predator. A prey agent’s reward only depends on its distance to the predator agents, and if the distance to the predator agent l is d_l , then its instantaneous reward is $-\sum_l \phi(d_l)$. An apex predator’s reward only depends on its distance to the predator agents, and if the distance to the predator agent l is e_l , then its instantaneous reward is $\sum_l \phi(e_l)$.

Observations Every agent observes their own position, velocity and accelerometer information as well as the vector to each corner of the two blocks that can be seen in the figure. Each agent further observes egocentrically represented positions and velocities of other agents within its sensor radius. Next, we introduce how the agents process these observations and map onto actions.

Perception Network As is common with swarm agents, motivated by both biological inspiration as well as learning complexity, we only want our agents to take the species of another agent into account and not individual identity. Further, we let each agent in the environment of the same species have the same policy. We want agents to have the capacity to generalize across different numbers of sensed other agents and potentially scale to very large numbers. We achieve this, as shown in Figure 3, by first applying a two layer feed forward neural network to every other agent’s position and velocity (within the focal agent’s sensor radius), and then we can combine these for each agent type by either summing or computing the mean, with similarity to

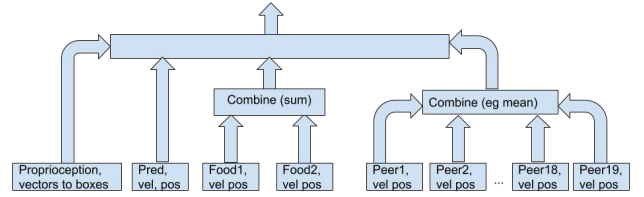


Figure 3: Diagram showing how the perception network first process the different parts of an observation separately and then combines the resulting representations.

Hüttenrauch et al. (2017).

Policy Network The policy that produces the continuous action vector is constructed as follows. The network that encodes the position and velocity of each other agent within the relevant sensor radius, is a two-layer feed forward network with 8 hidden units in each layer. For the predator agents (as many as 40 in the physics based experiments), we combine these representations by computing their mean. For the prey and apex predator agents, which are few (5 and 1), we use the sum to distinguish different numbers of agents in the same place. After this, we concatenate the result for the three types of other agents as well as the agent’s own proprioception and the vectors to the corners of the boxes on the floor. The resulting total representation is first processed by a two layer feedforward network with 128 and 64 units (with tanh activation) and then a recurrent network, an LSTM (Hochreiter and Schmidhuber, 1997) with 32 units. The final layer produces Gaussian distributed actions. Note that while parameters are shared between agents of the same type within an episode, agents are entirely independent in terms of action selection.

Locally observed grid worlds

In this section, we introduce a grid world ecosystem that is similar to the 3D physics based world, in the abstract ecological sense. The agents environment is a square map and they can rotate 90 degrees left or right, step forward or backward, or launch a yellow beam that represents predation. The predation beam is a difference to the physical 3D simulation where an agent only has to be near its prey to predate. In the grid world the agent has to be near and directed towards the prey, and choose this action, which is supplied by the platform Leibo et al. (2017). It is our strategy to only make as minimal and obvious design choices on top of the generic platforms as possible. However, it also comes with new possibilities including the possibility of enabling defense against predation and it makes a form of capture of prey even more important. Again, we have agents of three different varieties; prey(green), predator(blue) and apex predator(red). The apex predator gets reward +1 if predated on a predator (meaning that the predator is in the apex predator’s yellow beam), while the predator gets -1. Sim-

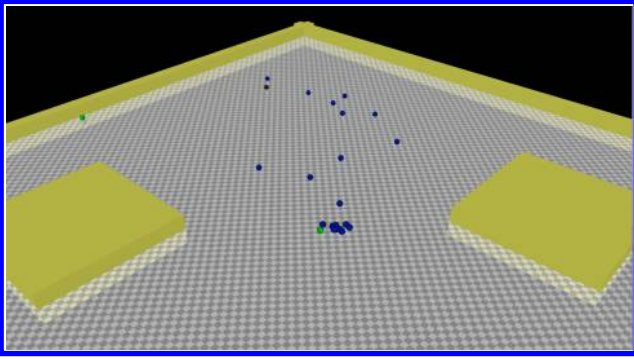


Figure 4: One prey (green) getting surrounded by most of the predators (blue).

ilarly a predator gets +1 for predating on prey, which then gets -1 . These are the rules that defines the environment shown in Figure 5.

The agents are trained using a ecosystem training protocol like in the 3D physics case, while differing by the learning update used. We replace MPO with impala learning updates (Espeholt et al., 2018) for this case in which the action space is discrete. The agents observe a small 9×9 window around the agent. Figure 5 shows screens from the resulting simulations when using a modest 5 apex predators, 10 prey and 50 predators. In the next section, we will also consider agents that spawn and vanish during very long episodes with thousands of agents, yielding population dynamics reflecting the relative success of each species over time.

Experiments

This section presents results from experiments with the introduced ecosystems that test our hypothesis that flocking and symbiosis can result in response to very generic predator-prey pressures in a spatial environment with range-limited sensation and action. We also investigate how the emergence of the relevant strategies depends on levels of predation pressure and the range of conspecific (within species) sensing among the predators.

Physical worlds

Our first range of experiments in ecosystems with 3D-physics, is varying the radius (5 vs 10 in a square with side length 48) within which predator agents can see each other, to see how well they make use of that information and what the consequences for the ecosystem are. All experiments features a radius of 5 both for sensing agents of other types, for the apex predator and the prey to see their conspecifics and the environment reward cut-off radius. We measure both the reward achieved for each type of agent during training, which was performed with 200 parallel habitats, and during regular evaluation against fixed pretrained prey and apex predator agents.

conspecific radius	5	10
number of predators	20	20
evaluation reward	2.7 ± 0.1	5.1 ± 0.2
symbiosis	0.6 ± 0.2	1.5 ± 0.4
predator group size	5.5 ± 1.4	9.3 ± 2.5
apex predator rewards	29.7 ± 5.8	51.8 ± 5.1
predator rewards	5.6 ± 1.5	9.3 ± 0.4
prey rewards	-36.0 ± 4.6	-64.0 ± 13

conspecific radius	10	10
number of predators	40	10
symbiosis	1.7 ± 0.3	0.96 ± 0.22
predator group size	16.8 ± 2.8	4.2 ± 1.3
apex predator rewards	119 ± 20	25.9 ± 8.7
predator rewards	12.9 ± 8.7	8.5 ± 5.2
prey rewards	-128 ± 18	-22.2 ± 9.1

Table 1: Results in 3D-physics ecosystems (at the end of training) for evaluation predator reward against fixed apex predator and prey, as well as on symbiosis measured by the average number of prey near the apex predator, predator group size measured by the average number of other predators near a predator and training reward for each species. The result are from three full runs of each of four experiments, pooling results for each species (3×10 policies) and calculating averages and standard errors. For reward, it is just the standard error for the three means. The ecosystems features 1 apex predator, 10, 20 or 40 predators, 5 prey and conspecific radius 5 or 10.

Unlike the results against the prey and the apex predator that the predators are learning with and that keeps changing, the results against the pre-trained agents provide a consistent well-defined evaluation. Further, during training we also measure distances between pairs of predator agents as well as between the apex predator and prey pairs of agents. From these we can judge to what extent predator agents flock and to what extent prey and the apex predator stay together. The results can be found in Table 1 based on species averages over the last $0.5e10$ of $3.0e10$ training steps.

Predator coordination: Predators perform better when able to sense other predators with a larger sensor radius as can be seen in Table 1 ($5.1 > 2.7$). We also see larger predator groups ($9.3 > 5.3$) when the conspecific sensor radius is larger. We believe that where there is a larger concentration of predators, there is likely to be a prey, and it is easier to join such groups if one can sense further.

Apex predator-prey coordination: Another observation is that the number of prey on average within a circle around the apex predator is much higher ($1.5 > 0.6$) when the predator has the wider conspecific sensor radius, which makes them a more effective hunter and increases the incentive of protection for prey. The smaller number (0.6) can

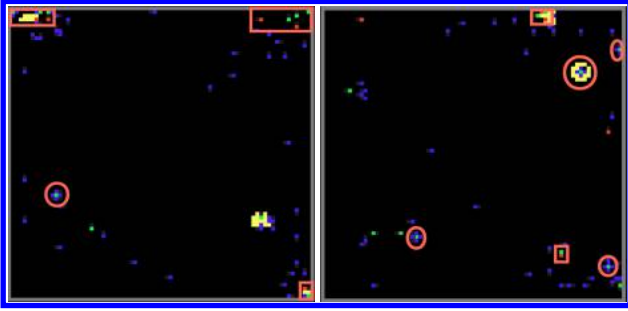


Figure 5: Grid world ecosystem showing group captures (predators of prey, highlighted by orange ellipses) and the apex predator-prey partnerships (orange rectangles).

largely be due to the apex predator chasing a group of predators that is chasing a prey and thereby, keeping the prey and the apex predator near each other. The stronger pattern (1.5) is clearly a more direct partnership pattern as can be seen in the videos¹, from which Figure 2 was taken. When the prey is chased by a flock of predators it cannot escape, it heads to the middle where it meets the apex predator that breaks up the pursuing predators. The videos¹ also show an example with a smaller world where the two prey and the apex predator find each other and the three stay together.

It is interesting that the apex predator earns substantially more reward when this partnership pattern with prey has emerged. The end result of predators getting better at hunting is that its own predator (the apex predator) is a big winner, via the adaptation by the prey. This is an example of the fascinating and indirect possibilities that arise from modeling more than two trophic levels. Figure 2 shows an example situation at the end of the learning for this case, while Figure 4 shows a situation generated by the very same agents (same weights) as in Figure 2, but here a lone prey is surrounded by a very large number of predators and does not find an escape route until the apex predator agent arrives. The prey here suffers catastrophic reward.

Varying numbers of predators: We compare varying numbers of predators, all with conspecific sensing radius 10. We see (Table 1) that the apex predator-prey partnership emerges more with 40 predators when the pressure on prey is obviously higher, and much less with 10 predators.

Grid worlds

In a first grid world ecosystem experiment with 5 apex predators, 10 prey and 50 predators, we consistently see groups of predators capturing prey agents, both in the middle of the floor (by 4 predators) and against walls (3 predators) or in corners (2 predators). We also see prey learning the defence strategy of sheltering near the apex predator, and the apex predator staying with the prey as it can enjoy reward for predating on approaching predators. These two strategies are visible multiple times in Figure 5.

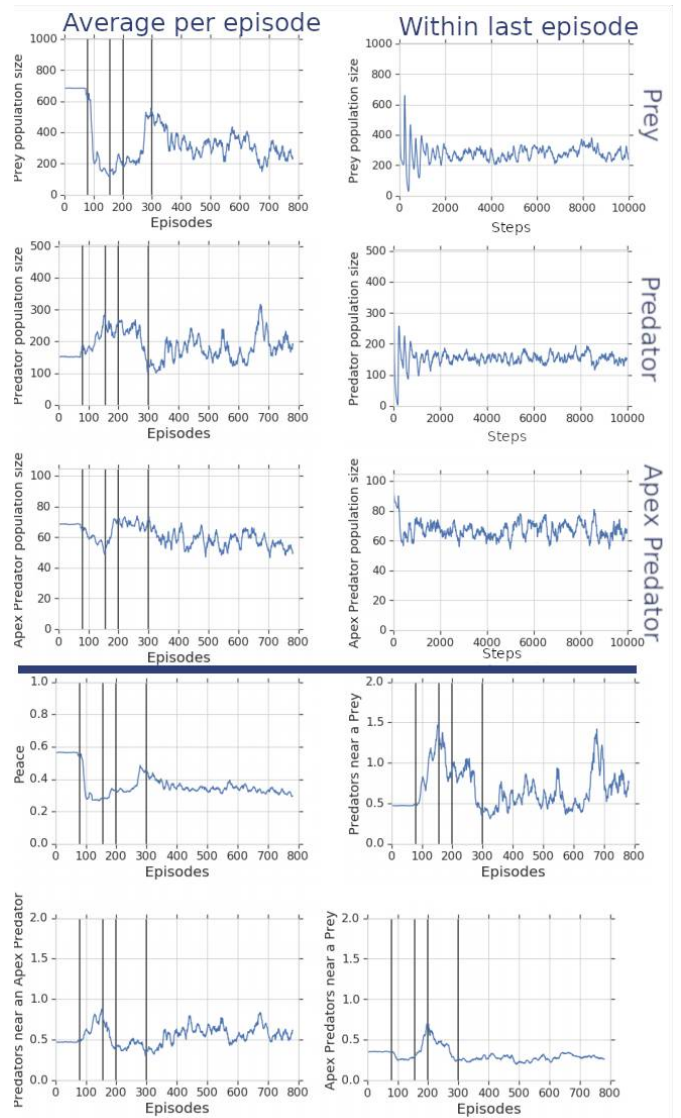


Figure 6: Top three on the left: The number of prey (top), predators (middle) and apex predators (bottom) present on average per episode. Top three on the right: The number of prey (top), predators (middle) and the apex predator (bottom) during the last episode. Bottom two rows: Peace (fraction of time agents on average is present in the environment (not vanished), numbers of predators near (within 3 cells on each side) prey, apex predators near prey (sheltering) and predators near the apex predator.

Large scale worlds with spawning and vanishing agents. In our final experiment, we extend our experiments both in scale and to introduce population dynamics, in the simplest way available. We give each agent a total amount of health (5) at the start, which is depleted by one unit each time they are preyed upon. Upon depletion, agents sit out

of the game for a certain amount of time (200) and then are respawned. While neither this, nor a fixed spawn rate as in Yang et al. (2018), represents the multiplicative nature of biological population dynamics, we do still get an environment where the relative population size indicates how well a certain agent type performs at a point in learning history. Within episodes we get a fluctuating curve with a significant oscillatory component around this mean, as is also familiar from ecology (Levin, 2009) if not in an identical manner.

In addition, our design allows for defence by prey species. That is, a predator can fire its yellow beam at the apex predator to reduce its health by one, and similarly prey can defend against predators. The structure of this defence indirectly encourages group strategies, since it takes a while for defence to eliminate the predator. While better for all agents if predators are eliminated, each individual can selfishly optimize reward by running away and allowing one's peers to defend them. While reminiscent of the sequential social dilemmas of Leibo et al. (2017), here a solution involving a familiar partnership with the third species is found.

From the perspective of the prey, we see a progression from annihilation to initiating defence to seeking shelter near the apex predator and jointly decimating the predator, which decrease its predation and peace increase. After this, the apex predator must hunt the predator more actively, and the predator gradually increase its predation on the prey.

Figure 6 shows the result of this experiment. Each (very long 10000 step) episode starts with 1000 prey, 500 predators and 100 apex predators, in terms of population level at different times of learning, and how the numbers varies within an episode. After the first 100 (parallel) episodes (24 hours on the compute cluster), we see a drop in prey numbers to near extinction, and in the within episode results, we see a complete annihilation early in the episodes. Predators have learnt early on to efficiently hunt prey. Closer inspection shows that the prey mostly individually flee, while they also start using their beam at the predator. After a period of nearly 300 episodes (3 days on cluster), prey numbers quickly increase while predator numbers now plummet before they adapt to this situation and gradually improve at the expense of both prey and apex predators. In Figure 6 (bottom right) we can see that the number of apex predators on average around a prey, climbs up to a peak just at the time when the change starts around episode 200. After this, we can see that this is followed by peace (low predation) to increase. We see predation levels between all species decrease substantially and we also see in Figure 6 that the species are now less frequently near each other. Our interpretation is that the predators decreased their pursuit of prey as a response to an apex predator-prey partnership that made it face a combination of predation and defence it was decimated by.

Conclusions

In this work, we have approached emergence of complex group behaviours in ecosystems as a sequence of optimization problems where each species is optimizing its fitness based on the current policies of other species. We use state-of-the-art deep reinforcement learning methods to address these optimization problems and consistently found the hypothesized emergence of strategies like flocking and symbiosis. These group patterns, which appear widely in different forms and contexts in nature, emerge through interactions of independently incentivized self-interested reinforcement learning agents acting in simple ecosystems that are not shaped to encourage the emergence of these particular strategies. Taken together, our results demonstrate that state-of-the-art reinforcement learning agents combined with our open-ended ecosystem training protocol can generate interesting coordinated behaviours familiar from nature.

References

- Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In *ICLR 2018*.
- Adami, C. and Brown, T. C. (1994). Evolutionary Learning in the 2D Artificial Life System Avida. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV*, pages 377–381, Cambridge, MA. MIT Press.
- Alvard, M. S. (2003). Kinship, lineage, and an evolutionary perspective on cooperative hunting groups in Indonesia. *Human Nature*, 14 2:129–63.
- Ashby, W. (1947). Principles of the self-organizing dynamic system. *The J. of General Psychology*, 37(2):125–128.
- Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mor-datch, I. (2017). Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*.
- Bedau, M. A. (2007). Artificial life. In Matthen, M. and Stephens, C., editors, *Philosophy of Biology*, Handbook of the Philosophy of Science, pages 585 – 603. North-Holland, Amsterdam.
- Bonabeau, E., de Recherches Du Fnrs Marco Dorigo, D., Dorigo, M., Théraulaz, G., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Proceedings volume in the Santa Fe Institute Studies in the Sciences of Complexity. OUP USA.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *CoRR*, abs/1606.01540.
- Conrad, M. and Pattee, H. (1970). Evolution experiments with an artificial ecosystem. *Journal of Theoretical Biology*, 28(3):393 – 409.
- Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy.

- Espeholt, L., Soyer, H., and Munos, R. e. a. (2018). IM-PALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1407–1416.
- Fretwell, S. D. (1987). Food chain dynamics: The central theory of ecology? *Oikos*, 50 3:291–301.
- Greeney, H. F., Meneses, M. R., Hamilton, C. E., Lichter-Marck, E., Mannan, R. W., Snyder, N., Snyder, H., Wethington, S. M., and Dyer, L. A. (2015). Trait-mediated trophic cascade creates enemy-free space for nesting hummingbirds. *Science Advances*, 1(8).
- Hairston, N. G., Smith, F. E., and Slobodkin, L. B. (1960). Community structure, population control, and competition. *The American Naturalist*, 94(879):421–425.
- Handegard, N. O., Boswell, K. M., Ioannou, C. C., Leblanc, S., Tj, D., and Couzin, I. D. (2012). The dynamics of coordinated group hunting and collective information transfer among schooling prey. *Current Biology*, 22:1213–1217.
- Harfoot, M. B. J., Newbold, T., and Tittensor, D. P. e. a. (2014). Emergent Global Patterns of Ecosystem Structure and Function from a Mechanistic General Ecosystem Model. *PLoS Biol*, 12(4):e1001841+.
- Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S. M. A., Riedmiller, M. A., and Silver, D. (2017). Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Hraber, P. T., Jones, T., Forrest, S., and Ds, K. (1994). The ecology of echo. *Artificial Life*, pages 165–190.
- Hung, D. S. M. (2015). *Reinforcement Learning Approach to Flocking with Fixed-Wing UAVS in a Stochastic Environments*. PhD thesis, Royal Military College of Canada.
- Hüttenrauch, M., Sosic, A., and Neumann, G. (2017). Guided deep reinforcement learning for swarm systems. *CoRR*, abs/1709.06011.
- Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. (2017). Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Levin, S. A. (2009). *The Princeton Guide to Ecology*. Princeton University Press.
- Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163. Morgan Kaufmann.
- Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR*, abs/1706.02275.
- Morihiro, K., Isokawa, T., Nishimura, H., and Matsui, N. (2006). Emergence of flocking behavior based on reinforcement learning. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 699–706, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Pachepsky, E., Taylor, T., and Jones, S. (2002). Mutualism promotes diversity and stability in a simple artificial ecosystem. *Artificial life*, 8:5–24.
- Packard, N. H. (1987). Evolving bugs in a simulated ecosystem. In Langton, C., editor, *Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87)*, Santa Fe Institute Studies in the Sciences of Complexity, pages 141–156.
- Ray, T. (1991). An approach to the synthesis of life. In *Artificial Life II*, pages 371–408. Addison-Wesley.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *SIGGRAPH Computer Graphics*, 21(4):25–34.
- Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 3rd edition.
- Ruxton, G. (2012). Group dynamics: Predators and prey get a little help from their friends. *Current Biology*, 22(13):R531 – R533.
- Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.
- Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.
- Yaeger, L. (1993). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a new context. In *Artificial Life III, Vol. XVII of SFI Studies in the Sciences of Complexity, Santa Fe Institute*, pages 263–298. Addison-Wesley.
- Yaeger, L. S. (2009). How evolution guides complexity. *HFSP Journal*, 3(5):328–339.
- Yang, Y., Yu, L., Bai, Y., Wen, Y., Zhang, W., and Wang, J. (2018). A Study of AI Population Dynamics with Million-agent Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18*.

Building a survivable protocell for a corrosive digital environment

David H. Ackley

University of New Mexico, Albuquerque, NM 87131
ackley@cs.unm.edu

Abstract

A computer produces outputs from inputs, and to do so reliably, its internal noise and variability must be managed effectively. Traditional computer architecture requires *hardware determinism*, but such perfect repeatability is increasingly incompatible with large-scale and real-world systems. Natural living systems, without the luxury of deterministic hardware, manage variability across the computational stack—and using such principles, *soft artificial life* offers a route to much larger and safer manufactured computers. This paper describes the engineering development of *C214*, a next-generation self-constructing digital protocell. *C214* struggles to survive in a challenging environment that, while not literally malicious, goes well beyond merely non-deterministic to deliberately destructive. Improved self-repair mechanisms, as well as active defenses in depth, give the new cell’s membrane a median survival time more than ten times greater than that of the earlier *C211*. A new grid-based cytoplasm is also presented, standing to offer a more stable environment for future layers of the ‘living computation stack’, and some basic cellular software engineering techniques are highlighted.

Soft artificial life and society

The ALIFE 2019 conference theme asks us: “How can Artificial Life help solve societal challenges?” Indeed, while making even software-based alife is a grand challenge on its own—and the results can be fascinating purely as objects of study—this author agrees that alife is also called to help address urgent problems in human technological society.

As an example, the author’s research and development program seeks to show that ‘soft’ alife, deployed on a new computer architecture, offers a radical but coherent solution to one of technological society’s ‘wicked problems’: The inability to make computer systems securable and robust.

This paper is a progress report for ongoing work with “digital protocells”—spatially-extended software constructs designed for the indefinitely scalable “Movable Feast Machine” (MFM) computer architecture (see, e.g., Ackley, 2013). These protocell designs, as well as performing life-like tasks such as growth and repair, are intended to serve as foundational components for future “multicellular” computations that are *inherently* robust—able to survive, even flourish, without global hardware determinism.

While the original protocells presented in Ackley (2018) operated in an otherwise empty universe, these new *C214* cells struggle under the “DReg and Res” (DR&R) computational regime—in which any atom adjacent to a DReg atom may be erased at random without notice, no matter the victim’s possible importance in any ongoing computation.

In the rest of this introduction, we provide motivation and context for the MFM architecture and the soft alife approach to robust-first computing, then outline the rest of the paper.

Computer scalability and securability

Since the 1940s, manufactured digital computers have revolutionized many areas of human society. Untold billions of units are now deployed in myriad uses and physical forms—but virtually all of them presume *global hardware determinism*: The hardware is to produce perfectly repeatable software execution, over the entire computer, however much time or available memory the software consumes. Any potential failures are to be eliminated *during system design*, for example by error-correcting codes or other fault tolerance.

The resulting determinism guarantee makes computations easier to implement, because application software need not worry about processor errors, or check its work, or confirm that values in memory are still good. Deterministic execution is a powerful and clean separation of concerns between hardware and software—but *guaranteeing* it becomes just unaffordable as computational systems grow. As the supercomputer community already knows (Cappello et al., 2009), for example, if you run enough hardware long enough, it *will* deliver undetected errors to software—and... Then What?

Because of software’s relentless focus on correctness and efficiency only, computing today has virtually no answer to that question. When, for whatever reason, something *does* go wrong *in a delivered system*, we simply have no idea how badly the ongoing computation will be damaged; we have no general way to estimate *how wrong* its output is likely to be.

Another—at present much worse—problem is that even when some task does fit comfortably within global hardware determinism’s limited ‘globe’, software’s systemic lack of redundancy also means it has *no structural defense* against

unexpected changes caused not by hardware failures but by malicious attacks. While redundancy, recomputation, and cross-checking are no *guarantee* of ‘cybersecurity’, if applied across the computational stack they can make crafting attacks much more expensive and their payoffs much more modest. By deploying redundancy effectively, we can also use *statistical* reasoning to estimate damages, long after our *logical* reasoning has collapsed due to perfection lost.

If deterministic execution gives way to indefinitely scalable *best-effort computing*, the ‘exotic’ techniques of artificial chemistry and soft alife—spatial processing and local memory, self-organization and repair, mobile and reproducing structures, etc.—will eventually become computation as usual. Getting there will take much work, but facing the abysmal state of computer security given traditional architecture, this author contends that society sorely needs artificial life researchers to help accelerate that transition.

Outline of the paper

The next section briefly places the present work into the context of some existing work, in computation broadly, and artificial life more specifically. The following sections summarize the goals and mechanics of the DR&R physics, then present the new *C214* protocell, focusing on its design improvements over the *C211* cell previously reported Ackley (2018), and then offer data on the relative impacts of the improvements. A short discussion and conclusions follow.

Related Work

Though this project uses a novel architecture, discussed further below, there is of course much work—both in and out of alife—employing similar goals, concepts, or mechanisms.

For example, work in distributed systems routinely handles non-deterministic execution and many kinds of errors, with decades of empirical and theoretical results (e.g., Saltzer et al., 1984; Clement et al., 2009). Self-stabilizing systems (e.g. Dijkstra, 1974; Dolev, 2000) are also related, as is work in spatial and amorphous computing (e.g., Abelson et al., 2000; Orhai and Black, 2012, is very relevant). One view is the present work applies such distributed systems principles to traditionally ‘single host’ computations.

Also, although the spatialization and other details differ, the goal of using programmable soft alife to engineer useful computations is shared between the present effort and alife work such as ‘chemical networking protocols’ built with lovely *Fraglets* language (e.g., Meyer and Tschudin, 2009).

Indefinitely scalable artificial chemistry

Traditional computer architecture specifies dedicated and highly-differentiated spatial structures for CPU, RAM, their interconnections and input and output channels, and that’s it. Though there are bigger and smaller CPUs and other components, they are all necessarily limited in size—RAM, for example, cannot extend too far from CPU, or access time

suffers. About systems much bigger than those limits, traditional computer architecture has little to say.

The trap of deterministic cellular automata That fundamental boundedness stands in stark contrast to *cellular automata* (CA) computational models. Their dimensionalities and details vary, but CAs virtually always specify *spatially isotropic processing*: At some level of description, the *same* function is performed at *all* locations in computational space. Though any CA instance must be finite, and spatial isotropy may be violated at its edges, the powerful implication is that *any finite size* could be chosen.

But CA models that require deterministic execution lack indefinite scalability and violate that assumption. As architectures, deterministic CAs are doomed to lose exactly that spatial unboundedness that made them so appealing at first. Much has been learned from them, but for truly large-scale computations, deterministic cellular automata are a trap. They are climbing a tree to get to the stars.

Artificial chemistries and the MFM Alife work based on *artificial chemistry* (Banzhaf and Yamamoto, 2015, is a good overview) begin with a physics model, where the primitive objects and processes represent ‘atoms’ or ‘molecules’ and ‘reactions’, from which ‘biological’ agents are constructed.

Unlike CAs more generally, artificial chemistries typically avoid the trap of determinism—for example, the exact order of reactions is usually random—but other assumptions can limit the scalability as well. Models that critically depend on floating-point accuracy are problematic, for example. Similarly, some dimensional assumptions—such as Fontana (1992)’s classic ‘0D’ well-stirred reactor—must be reexamined as well: Bigger beakers take longer to stir well.

The present research program focuses on chemistries that are implementable on the 2D ‘Movable Feast Machine’ (MFM) (e.g. Ackley and Ackley, 2016; Ackley et al., 2013). The MFM is designed first and foremost to be an *indefinitely scalable computer architecture*, and it jettisons everything that might prevent such open-ended scalability. Though it uses discrete sites and local transition rules like typical CAs, the MFM offers only best-effort reliability and guarantees essentially nothing about the cellular automata site update orders, or even the rates of receiving update events over time.

Those assumptions and others make MFM programming much more complicated than traditional software—but that added effort up front stands to pay off *indefinitely*, as the underlying hardware expands and improves.

An energetic, corrosive environment

The basic ‘laws of physics’ examined in this paper begin with DReg and Res, two of the oldest MFM elements, first reported in Ackley and Cannon (2011), and considered at length in Ackley (2013). DReg and Res form a homeostatic mechanism designed to help manage free space and con-

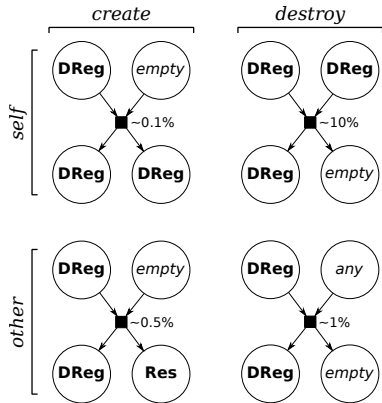


Figure 1: DReg reactions overview, with inputs (circles with out-bound arrows), outputs (circles with in-bound arrows), and approximate reaction probabilities (small black boxes). DReg may create Res or (more) DReg, or destroy DReg or *any other type of atom*. Empty sites are explicitly modeled in these pseudo-chemical reactions because the Movable Feast Machine is based on *conservation of space* rather than of *matter*. See text. (From Ackley, 2013).

trol crowding. Res (R) represents a ‘generalized Resource’ element that does nothing, on its own, except diffuse randomly into empty sites—but it can be employed or consumed however a computation wishes. By contrast, DReg (DR)—a ‘Dynamic Regulator’ element—randomly creates Res (and more DReg), while also randomly consuming *everything*—including, in particular, pieces of whatever other computation is running. Figure 1 depicts the DReg reactions.

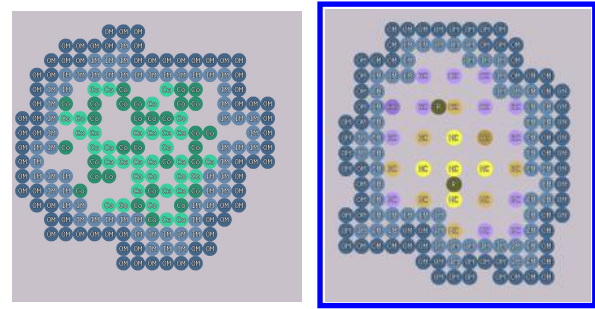
It’s important to note that this DR&R mechanism is not inherent in or enforced by the underlying MFM architecture—it’s purely ‘opt-in’ from the point of view of the element programmer. We don’t have to include DR&R in the ‘table of elements’ of our program, and if we do, we can easily write state transition code that simply erases every DReg atom that our element encounters. Similarly, we don’t have to wait for a Res to create new atoms if we don’t want to.

But if we *choose* to opt-in to the DR&R execution environment, we play along with its ground rules:

- Don’t create or destroy DR; only DR itself does that.
- Don’t grow unless we can consume R to do so. And,
- DR creates R ‘de novo’, but we can create R only if we destroy something previously made from R.

The current protocell violates DR&R rules in one major way: The membrane grows and shrinks as it pleases, without consuming or shedding R. But the risk of runaway growth is minimized by the membrane clinging to the cytoplasm—which mostly *does* obey DR&R rules—and by the previously-developed membrane internal consistency checks.

If our computation can persist and make progress while co-existing with DReg, we will have a limited but legitimate basis for calling it ‘robust’. The goal of the present work is a simple but robust protocell worthy of the name.



(a) *C211* (b) *C214*
Figure 2: Sample *C211* and *C214* protocells. See text.

A tough protocell for a harsh environment

The original *C211* protocell, and the ‘SPLAT’ spatial programming language used to implement it, both debuted at last year’s Artificial Life conference (Ackley, 2018). *C211* provided a range of useful ‘cell-like’ properties such as spatial isolation, mobility, and fission and fusion. It has a two-layer ‘membrane’ made of InnerMembrane (‘atomic symbol’ IM) and OuterMembrane (OM), plus an amorphous ‘cytoplasm’ element called Content (Co). It had some abilities to repair damaged membranes, but it was not explicitly designed for nor extensively tested in the DR&R regime.

The new *C214* protocell addresses that limitation. *C214* uses updated versions of IM and OM, plus two new cytoplasm elements called ‘HardCyt’ (HC) and ‘SoftCyt’ (SC). Figure 2 provides samples of the old and new protocells, and the rest of this section discusses the changes between them.

A more robust membrane for controlled isolation

C212 membrane development

Initial tests of *C211* cultured in DReg/Res revealed, among other things, some uncovered edge cases in its spatial transition rules. For example, a configuration such as in Figure 3 proved to be a ‘pinning state’ because, as it happened, no rules matched that shape. That didn’t matter originally, because such an ‘exposed IM’ state cannot arise during error-free growth of a *C211* in an empty universe—but given DReg’s random depredations, such frozen shapes do arise. The rest of the *C211* membrane continues to move around it, but eventually further DReg damage breaches the membrane entirely.

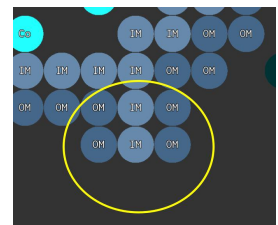


Figure 3: An unmatched *C211* state.

This lacuna was handled via a new SPLAT rule for IM:

```

160 | == Rules: IM management (shrink and die)
161 | given @ isa InnerMembrane
162 | given i isa InnerMembrane
165 | given c isa OuterMembrane
168 | given o isa OuterMembrane
169 | let x = i|o

```

```

182 | ---  ->  ...
183 | c@c  ->  .c.      # Cap off
184 | oxo   ...

```

(where the specific line numbers refer to the *C214* version of QMembrane.splAt, which incorporates these changes). The excerpts in lines 161–169 declare a variety of ‘keycodes’ for matching purposes; more than one keycode is sometimes declared for the same type (e.g., lines 165 and 168) to make the state transition rule more specific (discussed below.)

The left-hand side (LHS) of the spatial pattern rule at lines 182–184 matches states including the yellow-circled one in Figure 3: Three empty sites (line 182), then an OM, IM, OM sequence (line 183, with the @ keycode defining the center of the event), then finally two OMs with either an IM or and OM between them (line 184).

When that pattern matches (as it could, 180° rotated, in Figure 3), the right-hand side (RHS) of the spatial pattern rule is performed. On the RHS the keycode . means do nothing, while the keycode c means “write a copy of the ‘winning’ c chosen from the LHS.” Both of the cs on the LHS must have matched to reach this RHS, so one of them is chosen at random and copied to the event center, with the net result is that the center IM becomes a copy of an adjacent OM, and the *membrane invariant* (see Ackley, 2018) is restored.

One might wonder why that rule was chosen, rather than a simpler one such as:

```

| ---  ->  ...
| o@o  ->  .o.      # Cap off
| oxo   ...

```

which would also restore the membrane invariant in cases such as Figure 3. The more complex rule ensures that the copied OM is from an immediately adjacent site, while with the latter rule any of the four LHS os might have been chosen. Although the result is the same if all OM instances are identical, in a complex physics the atoms involved may actually be subclasses of OuterMembrane possessing additional state that differentiates them.

Minimizing spatial damage Using the rule as shown ensures that—should an ‘emergency repair’ to the membrane invariant be necessary—it will be done in a *least spatially damaging* way, by copying from a nearest neighbor of the appropriate type, chosen at random to preserve spatial isotropy. Such nearest-neighbor copying may or may not be what some higher-level computation might want—but it is defensible as the best choice to make absent any other information, purely on the basis of geometry—and we suggest that this is an example of what *best-effort spatial software engineering* looks like (but see Section Selective permeability via virtual stigmergy below for additional discussion).

A few other discovered corner cases were handled similarly. For example, although *C211* had a simple rule to correct a missing OM:

```

8 | given @ isa OuterMembrane
9 | given i isa InnerMembrane
81 | i_      .@
82 | i@  ->  ..      # Square off (outer)

```

which copies the @OM into an inappropriately empty adjacent site, a somewhat similar rule for IM needed to be added:

```

84 | _i      i.
85 | i@  ->  ..      # Square corner (inner)

```

The result of these membrane self-repair modifications was dubbed *C212*. Its DR&R median survival time (tested below) was a clear improvement over *C211*, but one could certainly hope for better still, considering that each of these protocells represents a significant investment in construction time and space, possibly embodying considerable state information of use to some higher-level computation.

C213 membrane development How can membrane survivability be improved further, after the locally-detectable membrane self-repairs are done? One might add a third ‘OuterOuterMembrane’ layer, for example. That would allow more time to repair DReg-induced damage before the membrane was completely breached, but it would also incur significant area costs as well as a major codebase redesign.

A more feasible alternative holds that the ‘real problem’ is the DReg itself. The DR&R regime makes it a sin to kill a DReg, but says nothing about trying to *keep them away* from the membrane. To that end, the rule shown in Figure 4 was developed. The spatial rule at lines 39–40 divides the world into three categories: Ds, Xs, and the OM itself (@). D, defined purely in SPLAT, is a vote for a DReg. While SPLAT **given** declarations introduce constraints that *all* must be true for a rule LHS to match, **vote** declarations, by default, will succeed if *any* LHS appearance of the given keycode receives more than zero votes. The rule in Figure 4 will match if one or more of the three Ds is a DR, at which point the expression **\$D.\$winsn** will represent a *site number* containing a DR.

The **vote X** declaration (at lines 30–35) uses a block statement, in {}’s, using *ulam* code (Ackley and Ackley, 2016) to express a relatively complex weighted voting criterion: Any **X** site containing a DR or any subclass of QMembrane (which includes both IM and OM), receives zero votes and so cannot be selected; otherwise votes are assigned based on the *squared distance* from the site being voted upon (**\$cursn**) to the event center (@). If the **X** in line 39 matched, for example, that site, at (-2,-1), would receive five votes $((-2)^2 + (-1)^2)$, while the leftmost **X** in line 40, at (-3,0), would score nine.

If the rule overall matches—meaning at least one **X** received votes and at least one **D** matched a DR—the RHS changes are performed. A limitation of the SPLAT spatial rule syntax is that it has no direct way to express a transformation like “Swap *whichever* sites won the **D** and **X** votes”, so that desire is expressed sententially instead, via

```

7 | == Rules: OM management Part 1 (miscellaneous business)
8 | given @ isa OuterMembrane
28 | # Fight DReg
29 | vote D isa DReg
30 | vote X {
31 | .   return ($curatom is DReg           // Swapping DReg inward doesn't help
32 | .       || $curatom is QMembrane)     // and moving membrane could be disruptive
33 | .       ? 0u                           // So no votes for them
34 | .       : (Votes) ew.getCoord($cursn).euclideanSquaredLength(); // Else farther is better
35 | .}
36 |
37 | change D { ew.swap($X.$winsn,$D.$winsn); }
38 |
39 |     XD     ..
40 |   XXD@ -> D...
41 |     XD     ..

```

Figure 4: A rule to push DReg away from the membrane, excerpted from the *C214* QMembrane.splat. See text.

the **change D** declaration at line 37.¹

Note this ‘DReg-fighting’ rule is quite aggressive. It’s willing to swap a DR with *anything* suitably farther away that isn’t a cell membrane or already a DR—and that absolutely could disrupt some significant structure that happened to be near a protocell’s outer membrane. Wearing a traditional deterministic ‘top-down programming’ hat, one would want to ask if that could actually happen, and handle it ahead of time somehow—but again, from a best-effort, bottom-up, spatialized point of view, that isn’t strictly necessary. If the need is great enough, from the point of view of whoever is having the event (here, OM, with the potential destruction of its protocell at stake), that is sufficient grounds for action, full stop. If some other structure thereby gets disrupted, well then it’s that structure’s fault for being too close to OM and DR.

Overall, the *C213* membrane is identical to *C212*’s but for that single added rule, and it makes a dramatic improvement in membrane lifetime, as discussed below.

C214 membrane development Pushing DReg is a great help, but it is not a panacea for at least two reasons:

1. Each event happens with a randomly-chosen symmetry. A DReg-pushing opportunity may be missed because the OM was looking in the wrong direction.
2. The MFM makes no guarantees about event delivery order. It’s unlikely but possible that a DReg could approach and burn all the way through a membrane, before the nearby membrane atoms got even one chance to react.

Bad luck can *never* be completely eliminated—and that is almost a mantra for best-effort computing—but it can be minimized statistically. And perhaps the most fundamental way to swing the statistics your way is to *control more*

¹It is (perhaps unfortunately) necessary that **D** appear *some-where* in the RHS to have any effect, but its specific location is ignored when an explicit **change D** declaration is provided.

space—which is what the *C213* DReg-pushing rule tries to do. The *C214* membrane, in turn, moves the battlefield even farther from the protocell, by deploying ‘cilia’ (Ci), which in this case are free-floating individual atoms, rather than literally hair-like structures. Unlike all other protocell components, the cilia live outside the outer membrane, but they remain near and in service to OM. Ci are created by transmuting available R, using an OM rule like:

```

# Deploy cilia
given Z : !($curatom is Cilium ||
           $curatom is QMembrane)
.
vote Z isa Res
change Z {ew[$Z.$winsn]=Cilium.instanceof;}

    ZZZ     ...
  ZZZ@ -> Z...
    ZZZ     ...

```

which combines **given**, **vote**, and **change** declarations with a spatial pattern to express essentially this transition: “If there are no Ci, IM, or OM nearby, but there is at least one R, change such an R into a default instance of Ci.”

A Cilium atom, in turn, has a variety of rules to push away DR, referencing a nearby OM to decide which direction is “away”, as in these excerpts from Cilium.splat:

```

10 | given @ isa Cilium
22 | # Push DReg
23 |
24 | vote o isa OuterMembrane
25 | vote d isa DReg
26 |
27 | change d { ew.swap($_.$winsn,$d.$winsn); }
28 |
29 |   _ddd@oooo -> .d.....
37 |     @oo -> ...
38 |     _doo   d...

```

where keycodes **o** and **d** use multiple sites and voting to help the rule match more often. Ci also have rules to diffuse near

the membrane, and to set their own site to empty (E) if no OM can be located—and such $DR \rightarrow R \rightarrow Ci \rightarrow E$ reaction chains are one way that R are ‘metabolically consumed’ by protocell operations. Overall, Ci is the signature addition differentiating *C213*’s membrane from *C214*’s, and experimental data shows that deploying Ci produced a qualitative jump in the membrane survival time, as discussed below. Figure 5 depicts a small *C214* protocell, with a developing cloud of Ci around it.

Although the Ci were developed, initially, solely to provide defense-in-depth against DR, they have since proved useful for other purposes. For example, some protocell operations consume R to obey the DR&R regime—and just as Ci can push DR away, these excerpts from *Cilium.splat* show how they can also, if deemed appropriate, pull R in:

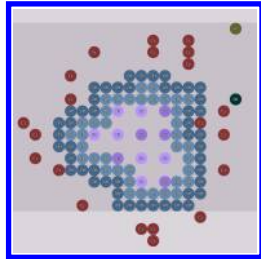


Figure 5: A young *C214* protocell developing cilia.

```

6 |== Data members
7 | u Bool mPullRes = true;
62 == Rules (Pull res)
63 given @ isa Cilium : $self.mPullRes
64
65 vote o isa OuterMembrane
66
67 vote r isa Res
68 change r { ew.swap($_.$winsn,$r.$winsn); }
69
70 rrr@_oo    -> ....r..
71
72 rrr@oooo   -> .r.....

```

where the ‘: \$self.mPullRes’ **given** condition on line 63 depends on the value of the data member declared (in *ulam* code) at line 7. Ci’s mPullRes defaults to true, but after that it is turned off and on by an additional *C214* OM rule—and how that rule knows whether another nearby R is likely to be a help or a hindrance, at the moment, is discussed in Section Selective permeability via virtual stigmergy, below.

In the near future we expect Ci will gain yet more useful abilities, such as assisting with protocell mobility and perhaps various types of environmental sensing. Its notable immediate utility during *C214* development was a reminder of yet another bottom-up mantra: Space is the place.

A quieter cytoplasm for bounded centralization

The protocell membrane is a crucial component, but of course a container with nothing to contain is only half a story. A version of the Mob element (originally discussed in Ackley and Ackley, 2015), formed the protocell interior ‘cytoplasm’ for *C211*. That Mob-like element—called Content (Co)—performed controlled growth from a seed, was collectively mobile, and used gossiping to disseminate movement commands. Those movement commands could

originate from anywhere inside the cell, because Co was completely also decentralized—which was satisfying from a first-principles robustness and ‘bottom-up purist’ point of view, but it also proved a challenge when it came to adding further functionality and programmability to the cell.

With that experience in mind, the new cytoplasm design goes a different way. Rather than an amorphous and centerless cloud akin to Mob, the *C214* cytoplasm is basically a semi-rigid grid spaced on two site centers (see Figures 2b and 5), more like the Router elements discussed in Ackley (2016). The grid is constructed from atoms of HardCyt (HC), which grow opportunistically by R transmutation starting from a single HC created (in the *de novo* case) by a Seed atom, which sprouts, as its only transition, into the configuration shown in Figure 6.

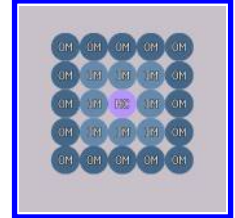


Figure 6: The *de novo* *C214* configuration, created by a Seed atom.

Once created, HC atoms do not move—although they can ‘melt’ into movable SoftCyt (SC) atoms—and HC’s relative rigidity underpins a cell-spanning local coordinate system. Each HC works to localize itself within the bounding box of all HC inside the membrane, by maintaining a four number array, as seen in these excerpts from *HardCyt.splat*:

```

17 == Data members
19 | u typedef Distance DistanceArray[4];
20 | u DistanceArray mDistances; // WNSE

```

where Distance is defined as a four-bit unsigned integer with a maximum value of 15. These distances are updated, during most HC events, by observing either another HC or a membrane atom in a randomly-chosen direction. When we (as the HC having an event) see a membrane atom, that is (the current) ground truth, and we set our distance in that direction to zero. When we see another HC, we set our distance in that direction to one more than theirs (though saturating at 15), and we set our distance in one of the two orthogonal directions to the max of their distance and ours.

If the HC happen to fill an axis-aligned rectangular shape, this process quickly finds consistent positions for all. More commonly, HC estimates partially settle and then jitter, depending on whether an HC listens to its neighbors or trusts its own eyes. But, in another signature of best-effort computing, although position estimates may be unstable or wrong, other *C214* processes use them *as if* they are correct.

In particular, once the estimated protocell volume has reached a minimum level, HC that find themselves near the center begin competing to be the ‘leader’ of the cell. Once some such HC is strong enough, in one fell event it declares itself the leader, and begins suppressing the neighboring HC. After a leader has arisen, an estimated distance-from-leader gradient forms—in addition to the ongoing position estimates—across all the HC. With leader competitions spa-

tially limited in this way, we virtually never see multiple separate leaders rising within a connected HC pool.

Having a single leader—like having a single queen bee—is a great help for coordinating protocell actions, and we look forward to fleshing out such uses soon. Of course, a single leader is also a single point of failure, but since the hierarchy’s stability depends on the leader (like the queen) actively suppressing competitors, if the leader *is* somehow lost, central competition soon resumes and another rises.

Evaluation of membrane survival times

Here we present the results of a small experiment to assess the effects of the membrane changes discussed above.

An experimental fixture To evaluate survival time changes due to the membrane modifications from *C211* to *C214*, a new Alien Explosion (AX) element was created. AX inherits from QContent, so it is recognized as ‘self’ by IM, but its behavior is simple and drastic: If an AX ever finds itself next to anything *other* than a QContent or an Empty, it erases itself after ‘triggering’ all nearby AX to do the same.

In effect, any contiguous cluster of AX vanishes almost immediately once any member of the cluster encounters an ‘alien’ atom. Using the ‘mfms’ simulator’s command line arguments ‘--halt-if-extinct AX’ and ‘--haltafteraeps 500000’, the time that a membrane was breached, up to a maximum of 500KAEPS², can be assessed with good accuracy.

Figure 7 shows the test fixture used in the experiment. A short distance from a membrane-enclosed block of AX, four DR are positioned, which gradually populate the extracellular environment with a mix of DR and R, as discussed above.

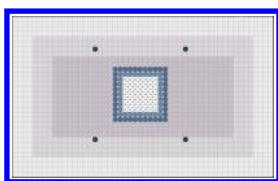


Figure 7: The membrane survivability experimental fixture is a single size ‘H’ tile (52 × 32 active sites), with 49 AX (white), 4 DR (black), 32 IM (light blue-green) and 40 OM (dark blue-green) of the particular membrane being tested. See text.

Results Membranes of the *C211–C214* protocells were tested 25 times each, and the membrane failure times recorded. Figure 8 presents all the data, with median survival times for each membrane called out.

Over the trials the original *C211* membrane rarely survived for 100KAEPS and never reached 200KAEPS; *C212* did notably better in most cases but never reached 200KAEPS either. *C213*, on the other hand, exceeded 200KAEPS over 30% of the time and once even reached the 500KAEPS test limit. Finally, *C214*’s cilia-waving monster membrane exceeded 500KAEPS on more than 50% of runs.

²1 KAEPS is an average of 1,000 events per site.

At present, longer runs remain to be performed, so no credible estimate of *C214*’s unbounded median membrane survival time is yet known. Of course death eventually comes to us all, and the details remain to be developed, but intuitively at least, the *C214* membrane *feels* survivable enough to support protocell growth and maturation and a productive computational lifetime, before that happens.

Cellular software engineering

It is unusual to see much real code in an life paper, but programming for an indefinitely scalable machine is different and more challenging than for traditional architecture. Code matters. Programmers think in code, and the metaphoric and literal *shape* of code subtly but strongly influences what future developmental directions will seem feasible. Here we focus briefly on indefinitely scalable software engineering.

‘Metabolic’ regulation *C211* used a ‘telomere’ counting mechanism to decide when its Co cytoplasm should stop growing, where *C214* now uses a combination of ‘R metabolism’—as dictated by the DR&R regime—plus cytoplasmic volume estimation via the bounding box described above. If the cell seems big enough, HC production shuts down and R import (discussed below) slows. Though this mechanism does not strictly conserve energy or matter or anything, it is quite effective, and—unlike the internally-facing telomere mechanism—it automatically adjusts, at least crudely, to a changing environment.

Selective permeability via virtual stigmergy *C214*, under the DR&R rules, grows by importing R into the cell. At first we used an IM data member to specify an importable element type—but that was limited to one type, and it was unclear how or when IM should reset that type, and it consumed 16 precious state bits. A much better solution was to have QContent specify a function ‘virtual Bool shouldImport(Atom)’ and have the IM call it *during its event* to determine if a given atom should be imported.

This way, subclasses can easily customize shouldImport to their needs, and it costs *zero* state bits in IM, and it’s never obsolete. It’s so nice that a shouldExport method was also defined, and *C214*’s IM can actually import and export simultaneously in a single event, using only one SPLAT rule.

This shows a third way to handle context, between defining explicit state to internalize it, and simply ignoring it like the repair rules in Section Minimizing spatial damage. The key to it is *spatiotemporal contiguity*: Having QContent reliably close enough that IM can run code on it during an event. Cilium, by contrast, uses a stateful mPullRes data member, despite its problems, to move freely as a forward fighter.

Towards living computation for society

Obviously, this is all just a beginning. There is a tremendous amount of work to do before we can demonstrate robust con-

Comparison of Membrane Robustness in Standard DReg & Res Environment

Survival times up to 500KAEPS; 7x7 AX fixture in single size H tile; 25 runs per condition

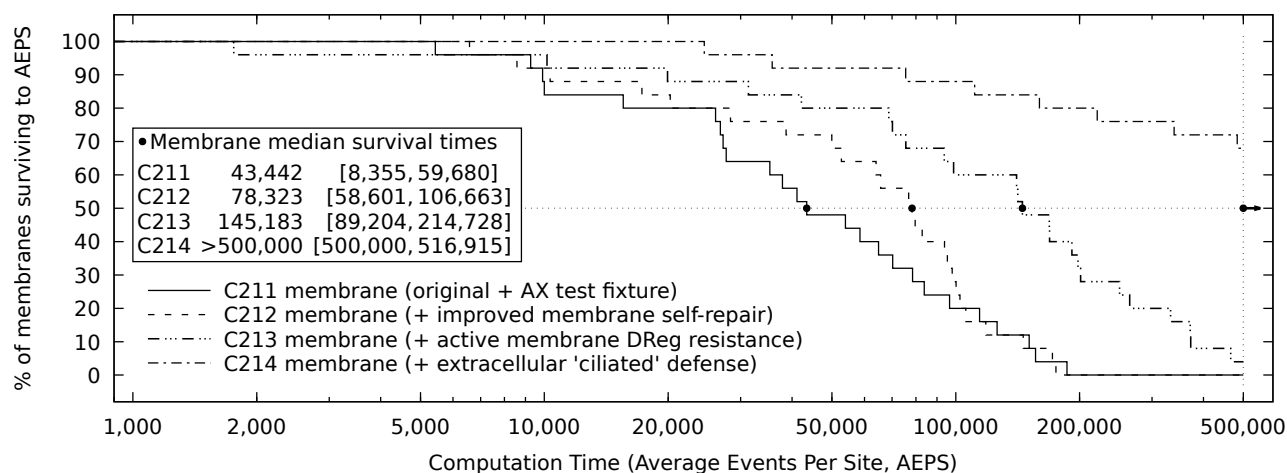


Figure 8: Survival time distributions of the four tested membranes, with medians and their 95% confidence intervals. See text.

crete system control using these indefinitely scalable, living computation methods. In at least one way, in fact, the *C214* protocell takes a step backwards: Its mobility is quite limited compared to *C211*. Some blame goes to the rigidity of HC compared to Co, but, in retrospect obviously, it is also due simply to the existence of DReg and Res, compared to the utterly empty environment around *C211*. Now, blind motion attempts soon create a ‘shockwave’ of R, DR, and Ci that impedes forward progress. Current work is developing dynamic control methods to alter the protocell aspect ratio, and we expect more advanced Ci will also help clear the way.

Yes, there is much to do, but also look how far we have already come. We have a growing palette of spatial and distributed programming patterns (topological invariant maintenance, gradient formation, leader election, etc.)—all familiar in other contexts, but now united under an *indefinitely scalable* architecture. And the goal of *robust multicellular programmability* for that architecture is closer than ever.

Indefinitely scalable soft alife is coming.

Acknowledgments Elena S. Ackley implemented the *ulam* compiler underlying SPLAT. Dan Cannon demonstrated a ‘cloud around a protocell’ mechanism before the current simulator even existed, but this author didn’t fully appreciate its power. *T2 Tile Project* subscribers provided valuable focus during the writing of the paper, and anonymous reviewer comments improved its quality.

References

- Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, Jr., T. F., Nagpal, R., Rauch, E., Sussman, G. J., and Weiss, R. (2000). Amorphous computing. *Commun. ACM*, 43(5):74–82.
- Ackley, D. H. (2013). Bespoke physics for living technology. *Artificial Life*, 19(3.4):347–364.
- Ackley, D. H. (2016). Indefinite scalability for living computation. In *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 4142–4146.

- Ackley, D. H. (2018). Digital protocells with dynamic size, position, and topology. *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 83–90.
- Ackley, D. H. and Ackley, E. S. (2015). Artificial life programming in the robust-first attractor. In *Proc. of the European Conference on Artificial Life (ECAL)*, York, United Kingdom.
- Ackley, D. H. and Ackley, E. S. (2016). The *ulam* programming language for artificial life. *Artificial Life*, 22(4):431–450. <https://dx.doi.org/10.1162/ARTL.a.00212>.
- Ackley, D. H. and Cannon, D. C. (2011). Pursue robust indefinite scalability. In *Proc. HotOS XIII*, Napa Valley, California, USA. USENIX Association.
- Ackley, D. H., Cannon, D. C., and Williams, L. R. (2013). A movable architecture for robust spatial computing. *The Computer Journal*, 56(12):1450–1468. <http://dx.doi.org/10.1093/comjnl/bxs129>.
- Banzhaf, W. and Yamamoto, L. (2015). *Artificial Chemistries*. MIT Press.
- Cappello, F., Geist, A., Gropp, B., Kal, L. V., Kramer, B., and Snir, M. (2009). Toward exascale resilience. *IJHPCA*, 23(4):374–388.
- Clement, A., Wong, E., Alvisi, L., Dahlin, M., and Marchetti, M. (2009). Making Byzantine fault tolerant systems tolerate Byzantine faults. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, NSDI’09, pages 153–168, Berkeley, CA, USA. USENIX Association.
- Dijkstra, E. W. (1974). Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644.
- Dolev, S. (2000). *Self-stabilization*. MIT Press, Cambridge, MA, USA.
- Fontana, W. (1992). Algorithmic chemistry. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 159–210, Redwood City, CA. Addison-Wesley.
- Meyer, T. and Tschudin, C. (2009). Chemical networking protocols. In *Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII)*.
- Orhai, M. and Black, A. P. (2012). Approximate parallel sorting on a spatial computer. In *Proceedings of the 2012 ACM Workshop on Relaxing Synchronization for Multicore and Manycore Scalability, RACES ’12*, pages 61–66, New York, NY, USA. ACM.
- Saltzer, J. H., Reed, D. P., and Clark, D. D. (1984). End-to-end arguments in system design. *ACM Trans. Comput. Syst.*, 2(4):277–288.

An Object Oriented Implementation of the MetaChem framework

Penelope Faulkner Rainford^{1,3,4}, Angelika Sebald^{1,4} and Susan Stepney^{2,4}

¹Department of Chemistry, University of York, UK

²Department of Computer Science, University of York, UK

³Department of Mathematical Sciences, Durham University, UK

⁴York Cross-disciplinary Centre for Systems Analysis

penelope.s.rainford@durham.ac.uk

Abstract

Our MetaChem framework supports the definition and combination of artificial chemistries. Here we describe an implementation of MetaChem in an object oriented language. We briefly define MetaChem, and provide an example in the form of a toy AChem: StringCatChem. We present the class hierarchy used to define MetaChem such that the implementation can run directly from a graph description of some AChem. This matches the description given by the formal framework definition. We also describe some generic functions of MetaChem that have been implemented and used in StringCatChem. This implementation is available on GitHub.

Introduction

Artificial chemistries (AChems), like other areas of artificial life, are predominantly software based. Mathematical models of our systems provide rigour in our definitions, yet they must also be implementable. We have developed MetaChem, a mathematically rigorous framework for defining single AChems, and for compositing different AChems into larger systems; it aims to provide a unified description language for all AChems. The formal MetaChem framework is defined in (Rainford, 2018); here we give a brief summary.

Our framework is designed to replace the only earlier framework for describing AChems presented in (Dittrich et al., 2001). That system is not designed with implementation in mind; rather, it is a tool to help describe AChems and draw comparisons. It splits an AChem definition into three parts, using the triplet (S, R, A) : a set of particles S , rules for reactions R , and the algorithm A . All non-particle aspects of a system are combined in A as part of the algorithm, including spatiality, rule application, global variables, timing, and logging.

That framework is helpful and complete for describing the well-mixed tank-based systems of early work, but since its introduction AChems have changed considerably. For example, many systems have spatial elements (Ono and Ikegami, 2001; Hutton, 2007). There are subsymbolic chemistries where particles have internal structure (Faulconbridge, 2011; Faulkner et al., 2018), and automata chemistries with instruction sets packaged with processors

for particles (Hickinbotham et al., 2010; Ofria and Wilke, 2004). There are even AChems without direct interaction between particles (Sayama, 2011). These new AChems pose difficult questions for the old framework: is position an aspect of the particle S or the algorithm A ? Is the processor in S or A ? If we have no physical linking, what is in R ?

Our new framework encompasses this increased complexity by considering the entire system as a sequence of events occurring on objects in an environment, rather than as a function of rule application to particles. With this we are able to better describe and compare the diverse set of AChems.

A generic description language such as this should also aim to make implementation easier. The ability to build tools that work for multiple systems is one of the primary goals for making these AChems compatible. This means that we need to be able to implement the framework in order to fully leverage its potential power. Previous work on implementation of AChem systems has been done (Bersini, 1999). However that work focused on a single AChem and does not generalise.

We have implemented our MetaChem framework and used it to combine two quite dissimilar AChems into a single system (Rainford et al., 2018a). In this paper we focus on implementation issues: how to convert the formal mathematical framework into a generalised reusable object-oriented implementation. We illustrate and examine the implementation of MetaChem here using StringCatChem, a toy AChem based on string concatenation. A Python implementation of the MetaChem framework, including StringCatChem, is available at github.com/faulknerrainford/MetaChem.git

A key requirement is that our implementation matches our framework as closely as possible. This makes it easy for different users to translate from one format to the other. Additionally, we want our implementation of the high level MetaChem structure to be independent of the implementation of the lower level AChem-specific particles and algorithms, to allow rapid integration.

	Primary Focus	Auxiliaries
Objects:	Particles	Variables
Containers:	Tanks	Environment

Table 1: Common parts of AChem Systems

MetaChem

Modularising Artificial Chemistries

There are axiomatic concepts in all AChems that we build on. We work on the basis of small components interacting to generate our systems. We are interested in the emergent properties and behaviours of these systems. To differentiate an AChem from an Individual Based Model we add requirements for simplicity and tractability in our particles and their interactions. The intention is that these systems work over large collections of individuals over long time periods, though most are currently limited by computational capability. To consider computational issues in our models we need our frameworks and models to consider implementation.

From these axiomatic concepts we identify many common elements of AChem systems. We use these as the basis for a bottom-up approach to systematic modularisation of AChems. Small, simple individuals and their interactions are our primary focus. We call these individuals *particles*. These exist in all AChem systems. Systems also have other variables, properties and values; we describe these in the *environment*. Much like in real chemistry, we separate the description of the “glassware” from our consideration of its particle contents. We have multiple *containers* in our system, which allow us to isolate particles and move them (analogous to the “beakers”, “pipes”, and “valves” comprising the “glassware”). This splits the dynamic parts of our system as shown in Table 1.

These components handle the “things” in our systems. There are also commonalities in the algorithms of AChems (and often their implementations) that we abstract out in our framework. Control flows, related to time and generations, occur in most systems. Some systems update across all objects in the system at once; others continuously update objects at random. If we can identify the modularised control that produces these timing systems, designers could switch between them. This would then allow designers to focus on the new AChem-specific features of their design, whilst using pre-existing elements to implement less unique aspects of their systems.

Having divided our “things”, we define our control flow in relation to these divisions. We *modify particles*, similar to reactions and interactions in chemistry. We *record observations* of our system. We *modify the environment*, such as by changing the temperature of the system. We move particles around our system. We *decide* which of these things we should do next. These control flow actions form the building

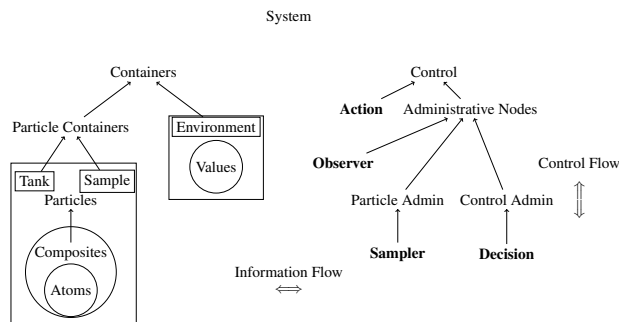


Figure 1: Conceptual structure of modularisation of AChems

blocks of our MetaChem.

We conceptualise these components into the structure shown in Figure 1, which we use to build a graph-based formalism. We have the overarching concepts of the System, made up of the elements formalised as graph *nodes* (Containers, Control), and as graph *edges* (Information Flow, Control Flow).

Control items are static nodes in the graph: their location and connectivity defined at the start, and remains unchanged as the AChem executes. These control nodes are connected by Control Flow edges, which together define the system’s specific algorithm.

Containers are also static nodes in the graph. They map to (“contain”) the dynamic particles and environmental values in the system.

Information Flow edges allow control nodes to influence the connected containers’ states (contents). Information can flow in either direction along an edge: *read* or *pulled* from containers’ state to the control node, and *pushed* from control nodes in order to update containers’ state.

We now describe these nodes and edges in sufficient detail to explain their implementation. For full formal definitions, see (Rainford, 2019).

Control nodes and Edges

The control flow of our system defines the AChem’s algorithm. The control evaluates a node’s definition, and moves to the next node. In the implementation, it iteratively executes a node’s transition function, and moves on to the next node; by traversing the graph in this manner it performs the relevant computation.

All control nodes execute the same basic state transition: the state changes, then control moves to the next node. The overall transition is defined through five individual transition functions: *read()*, *check()*, *pull()*, *process()*, *push()*, executed sequentially:

$$transition = read \ ; check \ ; pull \ ; process \ ; push$$

where \ddagger indicates strict ordering of function application from left to right.

Each of these transition function components plays a different role in the transition and thus uses a different aspect of the state.

read(): the current node collects information from (connected) external containers into temporary local containers, for used by the remaining transition functions. This action does not modify the external containers in any way. One can think of it copying the read particles and values.

check(): the current node uses its local information to generate a threshold probability value p , which it uses to determine if the rest of the transition function (the part that actually alters containers) occurs. In the implementation, it generates a uniform random number r ; if $p < r$, execution continues, otherwise it exits and moves to the next node.

pull(): remove particles and change information in external containers where appropriate. Any information so pulled must already been copied to local containers by *read()*, where it is available for local processing. Note that read information does not have to be pulled (that is, it can be copied, rather than moved).

process(): the main computation for the node, where the “chemistry” happens. It modifies the state of local particles and variables, including creating new particles and variables and destroying old ones.

push(): push variables and particles from the local variables into external containers; wipe the local containers’ contents.

Transition functions operate on local state, which exists only for the duration of the transition. Local particle containers and local environment containers are destroyed as soon as the transition function is completed, so the control nodes have no lasting state or memory. Any information used by a control node must come from containers at the start of a transition by using *read()* or *pull()*; any information or objects that need to remain in the system are written back to a container by *push()*.

These operations are summarised in Figure 2 and discussed in the context of specific node types below.

Control node subtypes

Our control nodes are partitioned into four subtypes: action, decision, sample and observer. We define these node subtypes by requiring some of the transition function parts to be null (identity), or by limiting the types of containers they can interact with during the transition, Table 2. The constraints on these subnodes help control the complexity of the system definition.

Action: read in information, check if an interaction occurs, process the particles in the system for the reaction to happen. It is not limited by which transition functions it executed, but it is limited by which containers it can *push()*

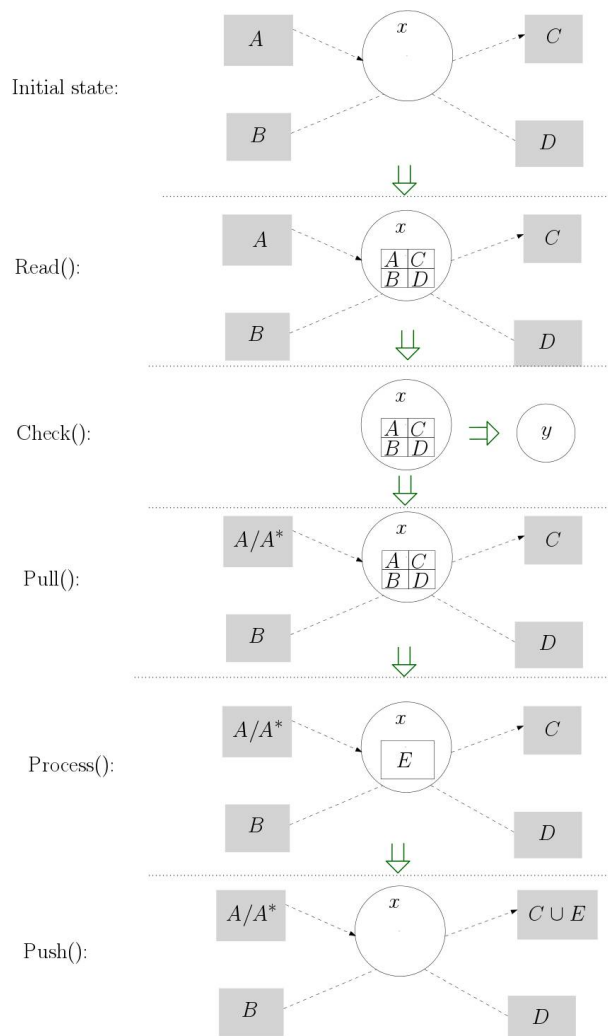


Figure 2: Summary of movement and processing of information done by transition functions in a node. A, B, C, D are container contents; $A^* \subseteq A$.

to, see Table 3. The limit to modify only samples allows parallelisation, and encourages controlled modification. The designer is required to consider what they wish to modify before they modify it, as they must first *sample* it from the tanks.

Decision: process the information from its containers and return a choice of the possible next nodes. It is limited to just *read()* and *process()*, so it cannot change the contents of any of the containers.

Sampler: move particles between containers. It does not compute or process the information, and it does not modify any particles or environment variables. It therefore has only a *read()*, *pull()* and *push()* function.

	action	decision	sample	observer
read	✓	✓	✓	✓
check	✓			
pull	✓		✓	✓
process	✓	✓		✓
push	✓		✓	✓

Table 2: Transition functions used by different types of control nodes; unchecked functions always return their default behaviour

	action	decision	sample	observer
tank			✓	
sample	✓		✓	
environment	✓			✓

Table 3: The set of container nodes that can be modified (push()/add() or pull()/remove()) by a control node, read() can always be performed on any container

Observer : observe but do not modify particles; modify the environment. It has a *read()* to view containers, and a *pull()* to allow it to edit environment variables only. It has a *process()* that allows it to compute summary statistics and changes to the environmental variables, and a *push()* to commit those changes back to the environment.

Container nodes

Container nodes are partitioned into two subtypes: Particle nodes and Environment nodes.

Particle nodes : mappings that take the node and the state of the system, and returns the set of particles in that container at that state. When the system is in a particular state the set of mappings of all the containers forms a partitioning of all the particles in the system. There are two types of particle nodes: samples and tanks. Tanks are protected containers. Particles in tanks can be moved in and out but cannot be changed; any changes must be made over samples, so the designer has to decide what will be changing.

Examples : A beaker being used for an experiment, a pipette, a petri dish.

Environment nodes : similar to particle nodes, except that they contain non-particle objects and information in the system. The system can have multiple environments, to make reference to the things in the environment easier. For example, one might want to store a time record separately to summary statistics or log information. These are all still accessed via a mapping from the node and state of the system to the dynamic information and objects.

Examples : Temperature readings, Bunsen burner, stirrer, observation log.

Use of mappings. It is key that containers work using mappings. Any container has three ways to access the information stored in them: *read()*, which returns a copy of all the information in the container; *remove([particle list])*, which removes all particles (or variables in the case of environment containers) in the list from the container (this is called by *pull()* during transitions so should be preceded by a *read()*); and *add([particle list])*, which adds the particles to the container (or adds a variable to the environment in the case of environment containers).

These mappings are a way to keep nodes independent of implementation. They define an interface. As long as the implementer can provide these three functions, they can store the items in what ever way they wish. This also means that control nodes that move objects and read them can be independent of the implementation of their storage. This allows for greater reuse of control nodes across systems.

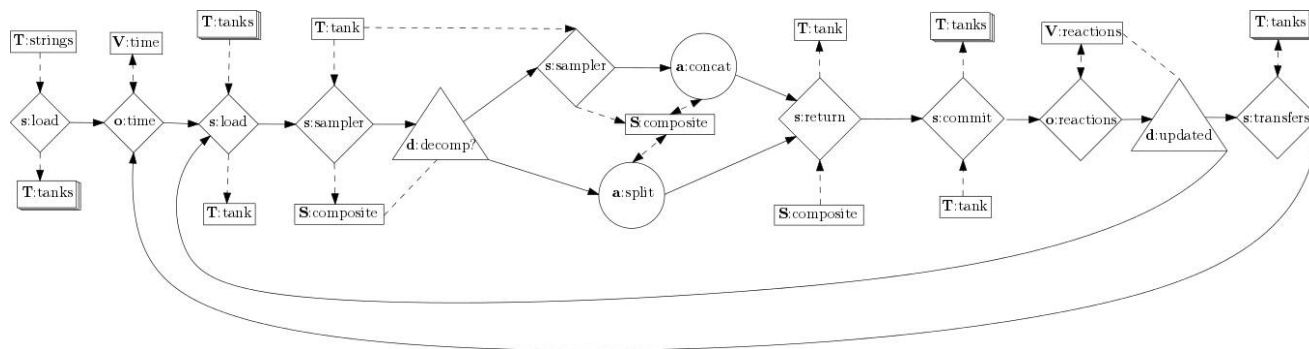
The limits on access to containers placed on control nodes is given in Table 3. The *read()* action is always possible for a control node of any container node (information is always knowable). However, we limit the modification of containers by nodes to make it easier to track activity in the system. This should also encourage limiting the scope of individual nodes to a basic action that may be reusable.

Example: StringCatChem

To help illustrate the use and power of MetaChem we introduce StringCatChem, a toy AChem. Its atoms are characters, the standard 26 lower case letters of the English alphabet. Composite particles are strings formed via concatenation. StringCatChem is situated in a collection of well-mixed (aspatial) tanks. When particles combine or split they remain in the tank. When we select a string we check if it contains any adjacent repeated letters; if so we split it between them. If not we select a second string at random and concatenate the two. We also randomly transfer strings between tanks.

The simplicity of this system means StringCatChem will continue to run until everything has formed a small number of large particles in each tank, all with matching letters at the starts and ends of them and no internal repeated letters. After this the system will not change. StringCatChem is therefore not a good choice of AChem if one wishes to study open-endedness or the transition to life. However, it makes a good example of implementation: the whole system can be implemented with 4 container nodes and 13 control nodes. The graph representation (and code) of the system is given in Figure 3.

The graph is a formal definition of the system itself; it is not merely a helpful visualisation of it. The code below it is a different concrete syntax of the same graph: it is the textual form for input to our interpreter. This is a description of the



```

1 edges = [[Sload, Otime], [Otime, Ssamplertank], [Ssamplertank, Ssamplertank],
2           [Ssamplertank, Ddecomp],
3           # Ddecomp splits control
4           [Ddecomp, Ssamplertankdecomp],
5           [Ddecomp, Asplit], [Ssamplertankdecomp, Aconcat],
6           # Control merges again at Sreturn
7           [Asplit, Sreturn], [Aconcat, Sreturn],
8           [Sreturn, Scommit], [Scommit, Oreaction], [Oreaction, Dupdate],
9           # Dupdate splits control again
10          [Dupdate, Stransfers],
11          [Dupdate, Ssamplertank], # Loop to the start of reaction
12          [Stransfers, Otime]] # Loop to the start to the generation
13

```

Figure 3: Graph for StringCatChem in visual and code form. For a key to the graph node shapes, see Table 4.

graph in terms of edges as pairs of nodes. The only further information the interpreter needs to run the StringCatChem is a limit on the number of transitions, and the specific implementations of the relevant transition functions.

The MetaChem framework is based on transitions rather than generations, as this is the natural step component; the MetaChem system has no built-in knowledge of generations or number of reactions.

Implementation

Now that we have overviewed the components of our graph-based MetaChem, here we discuss the implementation of the framework in Python. The hierarchical framework (Figure 1) translates readily to a class diagram, Figure 4.

Control nodes

The basic control node is defined as a transition function (Listing 1), which uses the five sub-functions. The transition function itself is used by almost all the subclasses. We also provide default null versions (which immediately *return*) of all of these five functions for use as null behaviours.

Our subclass nodes then use the default function in the top-level class for those functions not used by that type of node. The remainder of the functions are overwritten with default functions that perform basic functionality for that type of sub-node. For example, in the subclasses the default behaviour for *read()* is to read all information from all connected containers.

```

1 def transition(self):
2     self.read()
3     if self.check() < random.random():
4         self.pull()
5         self.process()
6         self.push()
7     pass

```

Listing 1: Transition function as defined in ControlNode class

```

1 def transition(self):
2     self.read()
3     return self.process()

```

Listing 2: Transition function as defined in Decision class

Decision does overwrite *transition()*, Listing 2, as in the case of Decision the transition needs to return a value back to the graph handler so it can transition to the correct node.

The graph handler checks if the node is running is a Decision and expects it to return a value.

Container nodes

Our container subclasses are very similar to the ContainerNode superclass. As long as a subclass implementation of a container provides a *read()*, *add()* and *remove()* function, the actual storage method is not important at this level.

For example, for StringCatChem we implement a list-based container. Listing 3 shows the *initialiser* and *read()* functions for this class.

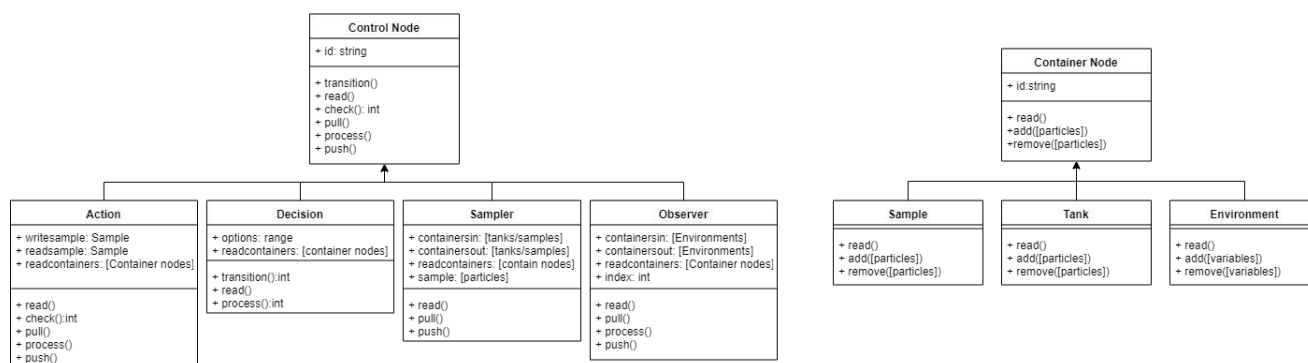


Figure 4: Class diagram of the implementation of MetaChem

```

1 class ListTank (node.Tank) :
2
3     def __init__(self) :
4         super(ListTank, self).__init__()
5         self.list = None
6         pass
7
8     def read(self) :
9         return self.list[:]
  
```

Listing 3: ListTank class definition including initialiser and read function

We could replace this list with a dictionary. Or we could use a database, with *add()* adding new records and *remove()* deleting them or modifying them so they are not recorded as part of the tank. Or we could use an array or file instead. We can use any such implementation, as long as we can reduce the interactions with it to *read()*, *add()* and *remove()*. So we can choose an appropriate implementation for any particular system at a particular size without needing to have this affect our control nodes and algorithm.

Given our containers are all so similar, why do we implement these subclasses? We do this so that we can type-check containers to ensure the different types of control nodes are modifying only allowed nodes.

Generic common nodes

There are common nodes that we will use multiple times in a system and in many different systems. We implement such types of nodes in their own two modules, one for containers and one for control nodes.

One obvious such common node is a counter node, which increments a particular variable that is later used by a decision node to control looping. Such a counter and decision, implemented as *ClockObserver* and *CounterDecision*, are given in Listings 4 and 5. The *ClockObserver* requires that the *pull* and *push* actions modify the same value, and takes an amount by which the value is incremented. The *CounterDecision* is set to look at the same variable, and is

```

1 class ClockObserver (node.Observer) :
2
3     def __init__(self, containersin, containersout,
4         readcontainers=None, increment=1) :
5         if containersin != containersout:
6             raise ValueError("Clock must read and write
7             to same variable")
8         else:
9             super(ClockObserver, self).__init__(
10                containersin, containersout, readcontainers)
11             self.increment = increment
12             self.clock = 0
13             self.variable = self.containersin
14             pass
15
16     def read(self) :
17         self.clock = self.variable.read()[0]
18         pass
19
20     def pull(self) :
21         self.variable.remove(self.clock)
22
23     def process(self) :
24         self.clock = self.clock + self.increment
25         pass
26
27     def push(self) :
28         self.variable.add(self.clock)
29         pass
  
```

Listing 4: ClockObserver node code

initialised with a threshold that determines next node.

Both of these generic nodes are used in *StringCatChem* to track and respond to the number of reactions that have happened. Their use in *StringCatChem* is given in Listing 6. Here we loop until 200 reactions have occurred.

Edges

The graph module interprets our graphs, given in textual form, and runs them. It stores the set of nodes into a dictionary. Each node is a key; the value is the node or list of nodes to which it has outgoing edges. This allows the interpreter to move on to the next node. In most cases there is only a single control node as the value; in the case of decisions there is a list of control nodes.












Element	Description
Containers	
	Tank containing particles.
	Sample containing an editable subset of particles.
	Environment containing non-particle variables and information in the system.
Control	
	Information administration node moves particles between containers by sampling them
	Observes particles and produces summary statistics (saved in the environment), also an information administration node.
	Control administration node makes weighted decisions on control flow based on the state of the particles and the environment
	Performs actions on particles based on state of particles and environment
Information Flow	
	Reads information from container into control node.
	Pull moves information out of a container into the control nodes local storage during it's operation.
	Writes information from control node into a container.
Control Flow	
	Arrow between Sampling, observer and action nodes to indicate control flow in system.

Table 4: Legend of types nodes and edges used in MetaChem graphs

```

1 class CounterDecision(node.Decision):
2
3     def __init__(self, options=2, readcontainers=None,
4                 threshold=1):
5         if isinstance(readcontainers, list):
6             raise TypeError("CounterDecision takes only
7                 a single readcontainer")
8         if options == 2:
9             super(CounterDecision, self).__init__(
10                options, readcontainers)
11            self.threshold = threshold
12            self.check = 0
13        else:
14            raise ValueError("CounterDecision takes
15                exactly two control options")
16        pass
17
18    def read(self):
19        self.check = self.readcontainers.read()
20
21    def process(self):
22        if self.check >= self.threshold:
23            return self.options[1]
24        else:
25            return self.options[0]

```

Listing 5: CounterDecision node code

```

1 Oreaction = control.ClockObserver(Vreactions, Vreactions
2 )
3 Dupdate = control.CounterDecision(2, Vreactions, 200)

```

Listing 6: Use of clock and counter nodes in StringCatChem

Discussion

We have successfully implemented our mathematical MetaChem framework in a reusable generic manner. With the correct set of system-specific nodes this allows us to define our graph structure and run our algorithm.

We don't provide results based on StringCatChem here. This system is very simple and over time the particles form into a very small number of large particles and stay that way. There isn't much to analyse in this. For results from other systems built on the basis of MetaChem see Rainford (2018); Rainford et al. (2018b).

The independence of our containers' implementation from our control node definitions allows us to implement generic nodes, which can be interchanged between different chemistries and experiments.

As an extension of this we can use subgraphs for common control systems. This immediately includes spatial and time systems e.g. the clock and counter combination described above implements a generational time system.

Our container implementation allows us to use simple storage, such as lists or dictionaries, for smaller/ simpler systems. For larger systems implement a database interface that could handle larger numbers of particles, longer runs, and different particle types.

MetaChem can be used to implement any AChem. A reasonable level of Python knowledge is needed to implement specific unique nodes and particles for a particular AChem.

Once the nodes have been implemented the chemistry itself can be instantiated and run without anything more than basic Python.

In our current implementation, available on GitHub, this is done by implementing the module `stringcat_nodes.py`, containing class specifications for all nodes required by `StringCatChem`. We then can run an experiment in this `ACChem` using a script, `stringcat_graph.py`, that generates instances of nodes and a graph, then runs it for a number of transitions.

Further Work

The implementation of core `Static Graph MetaChem` code described here means we can start porting other systems to run in our framework. An earlier version of `MetaChem` is used to combine `SwarmChem` and `JA-ACChem` (Rainford et al., 2018a). Porting further existing `ACChem`s would be the beginning of building a comprehensive code base for `ACChem`s. It could be expanded to include analysis tool that work across different systems, allowing direct comparisons of a range of `ACChem`s.

Next, we will expand the existing code to allow the graph to change at run time, allowing our `ACChem`s to be more dynamic: this is analogous to changing the “glassware” as a chemistry experiment runs, based on the results of reactions. This will use graph transformation rules to allow the graph to change and develop in response to the state of the system.

The naming of `MetaChem` is not just to imply that it is a meta system for handling `ACChem`s. It is capable of being an `ACChem` itself. Nodes are atomic particles, edges are links, and an `ACChem` graph is a composite particle in the `MetaChem`. Such an approach could allow us to evolve and generate `ACChem`s as products of a `MetaChem`.

Acknowledgements

PFR was funded by a University of York, Department of Chemistry Teaching Studentship.

References

- Bersini, H. (1999). Design patterns for an Object-Oriented computational chemistry. In *Advances in Artificial Life*, pages 389–398. Springer.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial Chemistries—A review. *Artif. Life*, 7(3):225–275.
- Faulconbridge, A. (2011). *RBN-World: Sub-Symbolic Artificial Chemistry for Artificial Life*. PhD thesis, University of York.
- Faulkner, P., Krastev, M., Sebald, A., and Stepney, S. (2018). Sub-Symbolic artificial chemistries. In Stepney, S. and Adamatzky, A., editors, *Inspired by Nature*, pages 287–322. Springer International Publishing, Cham.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010). Specification of the stringmol chemical programming language version 0.2. Technical report, Technical Report YCS-2010-458, Univ. of York.
- Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif. Life*, 13(1):11–30.
- Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artif. Life*, 10(2):191–229.
- Ono, N. and Ikegami, T. (2001). Artificial chemistry: Computational studies on the emergence of Self-Reproducing units. In *Advances in Artificial Life*, Lecture Notes in Computer Science, pages 186–195. Springer Berlin Heidelberg.
- Rainford, P. F. (2019). *MetaChemistry: an algebraic approach to artificial chemistries*. PhD thesis, Department of Chemistry, University of York, Heslington, York YO10 5DG.
- Rainford, P. F., Sebald, A., and Stepney, S. (2018a). Modular combinations of artificial chemistries. In *ALife 2018, Tokyo, Japan, July 2018*, pages 361–367. MIT Press.
- Rainford, P. F., Sebald, A., and Stepney, S. (2018b). Modular combinations of artificial chemistries. *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 361–367.
- Rainford, P. S. M. F. (2018). *Algebraic approaches to artificial chemistries*. PhD thesis, University of York.
- Sayama, H. (2011). Seeking open-ended evolution in swarm chemistry. In *2011 IEEE Symposium on Artificial Life (ALIFE)*, pages 186–193.

A Meta-Atom Based Sub-Symbolic Artificial Chemistry

Isaac Watson^{1,2}, Angelika Sebald^{1,3} and Susan Stepney^{1,4}

¹York Cross-disciplinary Centre for Systems Analysis, University of York, UK

²Department of Electronic Engineering, University of York, UK

³Department of Chemistry, University of York, UK

⁴Department of Computer Science, University of York, UK

iw596@york.ac.uk, angelika.sebald@york.ac.uk, susan.stepney@york.ac.uk

Abstract

We introduce a new sub-symbolic Artificial Chemistry, called the Meta-Atom Artificial Chemistry. It treats composite particles (composites of random boolean networks, RBN) as a new type of higher level atom, a *meta-atom*. These complex structures, together with a new kind of link, then form even larger, multi-level, structures. We show that Meta-Atom Artificial Chemistry exhibits rich behaviour, including reaction pathways that resemble catalytic reactions.

Introduction

We wish to use Artificial Chemistries to build and investigate open-ended systems. As such, we wish to minimise the number of explicit rules and properties needed, yet still get rich behaviours. For this reason we have developed *sub-symbolic artificial chemistries* (Faulkner et al., 2018), where the properties and dynamics of atomic and composite particles emerge from the underlying structure of the particles themselves. This is in contrast to more traditional AChems (Dittrich et al., 2001; Banzhaf and Yamamoto, 2015), in which the atomic particles have no underlying structure.

Spiky-RBN AChem (Krastev et al., 2016, 2017) is one such sub-symbolic artificial chemistry. In the original Spiky-RBN AChem, the lowest level structure, an atomic particle, is a random boolean network (RBN) (Gershenson, 2004; Drossel, 2008). An RBN has N nodes; at initialisation each node is assigned K inputs selected at random from K of the N nodes (possibly including the node itself), and a randomly generated Boolean function of these inputs; each node has a Boolean-valued state. The node's function is used to update the state of the node from the states of its input nodes at each timestep. When started from some initial state, the dynamics of the RBN goes through two phases: a *transient* phase of progressing through states, followed by an *attractor cycle* where states that are periodically revisited (Wuensche, 1998). The transient phase and attractor cycle have a rich variety of emergent properties; for $K = 2$ both these phases are relatively short, so calculating the properties is computationally tractable.

In Spiky-RBN, the RBN nodes are assigned to functional groupings known as *spikes*, which act as *bonding sites* for

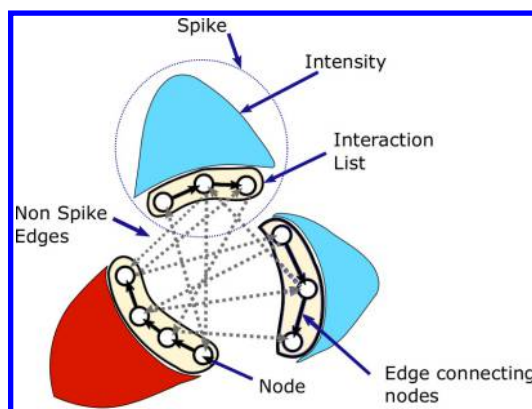


Figure 1: Spiky-RBN atomic particle. The specific node functions and connections of the underlying RBN result in emergent properties captured as *spikes*. Blue spikes have a positive intensity value and red spikes have a negative intensity value. After (Krastev et al., 2016)

the particle. Spikes consist of an interaction list, which is a list of all the nodes in the spike, and an intensity, which is a signed integer value related to the states of the nodes in the interaction list. These spikes, and hence the particle's bonding properties, are emergent properties of the underlying structure and dynamics of the underlying RBN; for details of how they are calculated from the properties of the RBN, see Krastev et al. (2016). An atomic particle is shown in Figure 1.

Particles can form a link (bond) if they have spike intensities of equal and opposite magnitude; the nodes in each particle have their inputs redirected to link together to form a composite particle (Figure 2). A composite particle may decompose, or may react with another particle. This leads to reaction pathways in which novel composite particles form from the reactions of an initial set of particles in a reactor. Krastev et al. (2016, 2017) show that the Spiky-RBN AChem has a rich behaviour and is able to form composite particles of varying sizes and structures through complex reaction pathways.

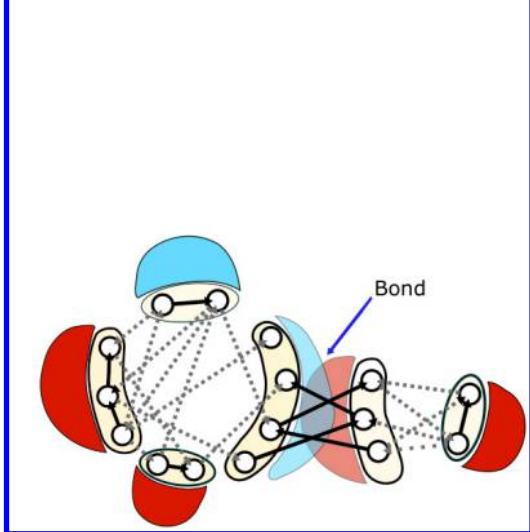


Figure 2: A composite particle composed of two atomic particles. After (Krastev et al., 2016)

Here we introduce two variants of the original Spiky-RBN AChem. Firstly, we define Frozen-Node-Spiky-RBN (FN-SRBN), in order to explore how further emergent properties of RBNs can be exploited as bonding properties. Secondly, we use this new variant as the basis of Meta-Atom Artificial Chemistry. This Meta-Atom AChem reacts composite particles through a higher level kind of bond, using ‘dangling nodes’ in lower level bonds as the bonding criterion. As with Spiky-RBN AChem, the aim is to build an AChem where the bonding properties emerge from its underlying structure and dynamics. Here the underlying structure is that of the lower level bonds in composite particles formed in FN-SRBN.

The Frozen-Node-Spiky-RBN AChem

In this section we introduce a modified version of the Spiky-RBN AChem, the Frozen-Node-Spiky-RBN (FN-SRBN) AChem. We discuss why this AChem has been developed, and describe the structure of atomic particles and the new bonding process. We conclude the section with the introduction of a special composite particle called a *ring*, and an experiment to generate ring structures, which will be used as the basis of the later Meta-Atom AChem experiments.

Atomic Particles in the FN-SRBN AChem

FN-SRBN AChem is used to explore if other emergent properties of RBNs, here *frozen nodes* (see later), can be used as part of an AChem, and to see how other approaches to generating the interaction lists affect the behaviour of the AChem. This allows us to explore new Spiky-RBN AChems and to see if they have the rich behaviour which is required for a useful AChem.

Spiky-RBN AChem generates interaction lists by following the connections of the nodes in the RBN ordered by their ‘influence’ (number of outgoing connections). In the FN-SRBN AChem we instead follow the connection of randomly selected nodes. The algorithm which builds the interaction list is shown below in Algorithm 1.

Spiky-RBN AChem calculates the intensity of the spikes as a function of the state changes of the nodes over the

```

Data: N: List of all nodes in the RBN
while N is not empty do
  Create new empty Interaction List  $IL_i$ ;
  Remove random node  $n$  from N;
  Append  $n$  to  $IL_i$ ;
  while  $\exists n' \in N$  where  $n'$  is an input to  $n$  do
    Remove  $n'$  from N;
    Add  $n'$  to  $IL_i$ ;
     $n \leftarrow n'$ ;
  end
   $i++$ ;
end

```

Algorithm 1: Building interaction lists for an RBN

RBN’s attractor cycle. In the FN-SRBN AChem the intensity of the spike is instead dependent on the states of the ‘frozen’ nodes in the attractor cycle. Frozen nodes are nodes whose states remain constant over the attractor cycle, and are a prominent feature of RBNs. Each node n in a spike S is assigned a weighting n_w :

$$n_w = \begin{cases} 1 & \text{if node frozen in True state} \\ -1 & \text{if node frozen in False state} \\ 0 & \text{if node is not frozen} \end{cases} \quad (1)$$

The intensity of an FN-SRBN spike S_i is the sum of weightings for each node in the interaction list IL_i of the spike:

$$S_i = \sum_{n \in IL_i} n_w \quad (2)$$

As with Spiky-RBN AChem, this gives a spike with a magnitude and a sign. The intensity is constrained by the length of the interaction list:

$$- \#IL_i \leq S_i \leq \#IL_i \quad (3)$$

FN-SRBN AChem introduces a new parameter called spike type S_T which influences how easy it is for two spikes to bond. The spike type ranges from 1 to 3 and is a function of the length of the interaction list of the spike

$$S_T = \begin{cases} 1 & \text{if } \#IL_i < 5 \\ 2 & \text{if } 5 \leq \#IL_i < 10 \\ 3 & \text{if } 10 \leq \#IL_i \end{cases} \quad (4)$$

The attractor cycle is found from an initial state of all nodes being false, and the number of inputs each node takes is $K = 2$ in order to obtain stable rather than chaotic behaviour (Wuensche, 2008).

Bonding Atomic Particles

FN-SRBN AChem forms bonds in the same manner as Spiky-RBN AChem, with two spikes from different atomic

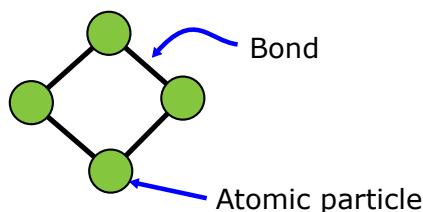


Figure 3: A *ring* of four atomic particles. The green circles are atomic particles, each with exactly two spikes, and the lines between atomic particles are bonds.

particles linking together as in Figure 2. Spiky-RBN AChem requires that the intensity of both spikes must sum to zero before bonding for the bond to form, and after bonding for the bond to remain stable (the intensity may change on bonding since the linked RBNs have a different dynamics from the unlinked ones; this decomposition property arises naturally from the emergent intensity property).

In FN-SRBN AChem the bonding condition depends on both the intensity and the spike type. Let the spike from the first particle be S^1 with intensity S_i^1 and type S_T^1 , and the spike from the second particle be S^2 with intensity S_i^2 and type S_T^2 . Then the two particles can bond if the spikes meet the appropriate condition specified in eqn. 5:

$$|S_i^1 + S_i^2| = 0 \text{ and } S_T^1 = 1 \vee S_T^2 = 1 \quad (5a)$$

$$|S_i^1 + S_i^2| \leq 1 \text{ and } 2 < S_T^1 + S_T^2 < 6 \quad (5b)$$

$$|S_i^1 + S_i^2| \leq 2 \text{ and } S_T^1 = 3 \wedge S_T^2 = 3 \quad (5c)$$

This condition reduces to the Spiky-RBN AChem behaviour if either IL is very short (just one node), but becomes more relaxed as the ILs get longer. This more relaxed behaviour is needed to ensure sufficiently rich behaviour, to compensate for the different distribution of spike intensities in FN-SRBN.

If the condition continues to be met after bonding, then the bond is stable. If the bond is unstable, it breaks.

Rings of Atomic Particles

A *ring* is a special composite particle consisting of atomic particles each of which have two bonded spikes and no un-bonded spikes. A ring of four atomic particles is shown in Figure 3.

This ring structure can be considered an ‘inert’ composite particle. Since all its spikes are bonded, it cannot bond to any other particles. We use this structure for our initial Meta-Atom AChem studies here, since we do not need to consider any lower level reactions between these inert rings.

Experiment: Growing a Set of Rings

The aim of this experiment is to generate a set of rings using FN-SRBN AChem. The set can then be used as the basis for a higher level AChem.

```

while true do
  // Phase 1 – find atomic set
  chain := empty
  while empty chain do
    R := 20 randomly created atoms
    tries := 0
    while empty chain and tries < 95 do
      a1, a2 := R
      if a1, a2 can bond then
        | chain := bonded (a1a2)
      end
      ++tries
    end
  end
  // Phase 2 – build a ring
  if head,tail of chain can bond then
    | ring := bonded chain
    | exit
  end
  tries := 0
  while tries < 700 do
    a := R
    if a, chain can bond then
      | chain := bonded (a,chain)
      if head,tail of chain can bond then
        | ring := bonded chain
        | exit
      end
    end
    ++tries
  end
end

```

Algorithm 2: Building a ring from random atoms

This is done using a well stirred reactor with no spatial component. The reaction contains a set of 20 atomic particles. To form this set, atoms are randomly generated with a size N chosen uniformly between 2 and 15 nodes. An atom is kept provided it has exactly two spikes, which it needs to be a link in a ring. Atoms are randomly generated in this manner until there is a set of size 20.

The algorithm to build a ring has two phases, see algorithm 2.

Phase one finds a suitable atomic set, by producing a short chain of two atoms. The reactor is initialised with 20 randomly generated atoms. Two atoms are selected at random, and a bond is attempted. Selection of pairs of atoms continues until a bond is formed. If after 95 attempts no initial bond is formed, another set of atomic particles is generated.

Phase 2 generates a ring. It starts with the initial pair of bonded atoms, and the atomic set, from Phase 1. A bonding attempt is made between the head and tail of this chain of two atoms, attempting to form a minimal ring. If no ring forms, a further atom is selected at random from the set,

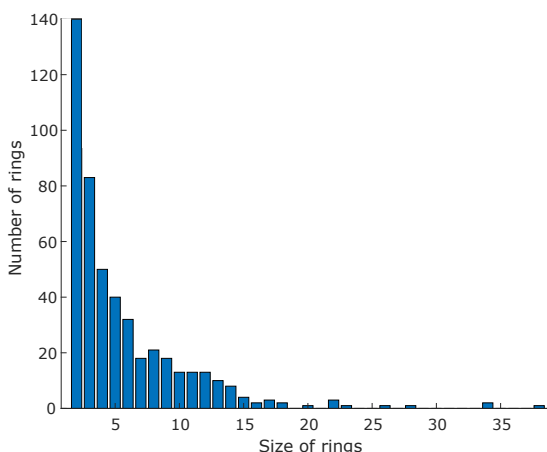


Figure 4: Chart showing size of rings against number of rings of this size. Total number of rings is 625, smallest size is 2, largest is 38.

and a bond is attempted with the short chain. This random selection is continued until the chain is lengthened by one atom. Each time the chain is lengthened, there is an attempt to bond the head and tail to form a ring. If no ring forms, the chain is further lengthened. If no ring has been created after 700 lengthening attempts, then entire process is started again.

Using this algorithm, 625 rings were generated. The distribution of ring sizes for atomic particles consisting of up to 15 nodes and an attractor cycle length of 3 is shown in Figure 4. The figure shows that the majority of rings consist of less than five atomic particles.

It is possible to obtain much larger rings, up to size 38 in this experiment, although this is rarer. There are two reasons for this rarity. Firstly, the algorithm is biased towards smaller rings, since it incrementally increases ring size, and stops once a stable ring forms. Secondly, larger rings have more bonds, increasing the likelihood of an existing bond becoming unstable and breaking when a further bond is formed during lengthening.

The Meta-Atom Artificial Chemistry

In this section we introduce the Meta-Atom AChem.

Meta-Atoms

FN-SRBN AChem can be used to bond particles with one kind of bond, as demonstrated above. Meta-Atom AChem introduces a new kind of bond, to bond composite particles formed in FN-SRBN in a new manner. Meta-Atom AChem has its own ‘atomic particles’, called *meta-atoms*. A meta-atom’s behaviour and dynamics emerge from its underlying structure, which here is that of a ring of bonded FN-SRBN AChem atomic particles. We can consider the Meta-Atom AChem as a higher level sub-symbolic artificial chemistry,

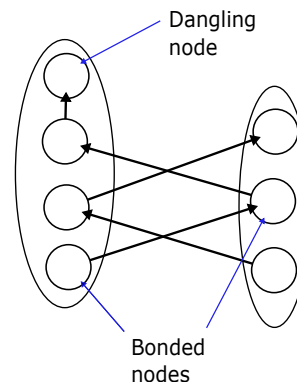


Figure 5: Two spikes bonded together resulting in a single dangling node. The colours of the spikes and connections to nodes outside of the interaction list have been removed for clarity.

as its properties and dynamics emerge from a lower level sub-symbolic artificial chemistry.

We use FN-SRBN rings as meta-atoms in this initial study, as they are inert at the FN-SRBN level, and so reactions between low level particles does not need to be considered. In principle, composite particles that are not inert could be used as a basis for meta-atoms. Meta-atoms of reactive composite particles will be added in future iterations of our Meta-Atom AChem, where we will investigate the interaction of the two bonding mechanisms. Such interaction could result in new reaction pathways and new products emerging.

Dangling nodes

Figure 2 shows a composite particle composed of two atomic particles. Figure 5 shows a zoomed view of the bond between the two particles. Figure 5 shows that when two spikes bond, not all of the nodes in a spikes need to be rewired to connect to the other spike. Such nodes play no direct part in the bond; we refer to these as *dangling nodes*. If there are multiple dangling nodes in a bond, we refer to this as a *dangling tail*. Nodes that are directly involved in the bond are referred to as *bonded nodes*.

Dangling nodes and tails are an emergent property of the bonding process between atomic particles. Dangling nodes and tails arise when the lengths of the ILs of the two bonded spikes are unequal. The number of dangling nodes, γ , in a bond is the difference in the lengths of the ILs of the two spikes:

$$\gamma = |\#IL_1 - \#IL_2| \quad (6)$$

Each dangling node is associated with a spike, which has an intensity and type. We assign the dangling node an intensity and type equal to the intensity and type of its spike.

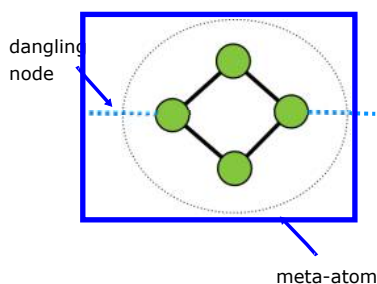


Figure 6: A meta-atom consisting of a ring of size four and two dangling nodes. The green circles are atomic particles; black lines are bonds between atomic particles; the light blue dashed lines emerging from two of the atomic particles represent bonding possibilities via dangling nodes, or dangling tails.

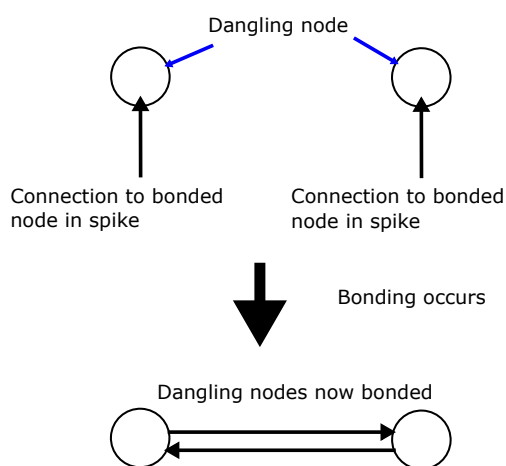


Figure 7: Dangling nodes bonding. This causes bonds in the low level structure of the atomic particle to break.

Meta-Atom bonding

As with the FN-SRBN AChem, there needs to be a mechanism that allows meta-atoms to bond in order to generate larger structures. To bond meta-atoms, we use the emergent dangling nodes within bonded spikes. We represent a meta-atom diagrammatically as a ring of atomic particles with dangling nodes (Figure 6).

Here FN-SRBN AChem is used as the lower level AChem to generate rings for meta-atoms. Future work will extend Meta-Atom AChem so that the low level structure of meta-atoms can be made up of composite particles generated by other Spiky-RBN AChems (Krastev et al., 2016).

For dangling nodes to attempt to bond, as with bonding spikes, we need to see if the intensity and type obey the bonding criterion (eqn. 5). If the condition is met, the dangling nodes swap connections (Figure 7). We refer to bonded meta-atoms as a meta-molecule.

The swapping of connections alters the underlying struc-

tures of the atomic particles in the ring. This means that higher level bonding alters the lower level structure of both meta-atoms involved in the bonding. Hence the properties of every spike in the ring need to be recalculated after a higher level bond has occurred. Bonding at the higher level can lead to intensities of the spikes changing (and thus the intensity of dangling nodes). This can cause the bond between dangling nodes to break and can also cause bonds in the low level structure to become unstable and break. A change in higher level bonding can perturb the low-level structure and alter it. A change in the low-level structure can then affect the higher level bonds in turn. Thus after meta-atoms bond to form a meta-molecule, there can be a period of instability in which the meta-atom's internal structure and the higher level bonds can change.

This period of instability post meta-atom bonding means that the structure of a meta-molecule needs to be reanalysed until all of its meta-atoms are stable. A stable meta-molecule is one in which all of its composite meta-atoms bond. Both high-level and low-level particles must meet the appropriate stability criteria as defined in eqn. 5. If the higher-level bonds between meta-atoms break, the meta-molecule is checked for meta-atoms which no longer have high-level bonding with other meta-atoms. If a meta-atom with no bonds to the meta-molecule is found, it is removed from the meta-molecule as it has 'broken away' from the meta-molecule.

Experimental Implementation

The FN-SRBN and Meta-Atom AChem were implemented using the Python programming language and the NumPy numerical library. An object-oriented design approach is used and the software is available on GitHub¹.

Analysis of Meta-Atom Bonding

The simplest case of meta-atom bonding is two meta-atoms M_1 , M_2 , each with one dangling node, coming together and attempting to bond. The eight possible outcomes of this reaction are shown in Table 1. In this table, R1 (Ring 1) Stable shows whether the underlying structure of M_1 is affected by the higher-level bonding between the meta-atoms. If R1 Stable is false, the underlying structure is affected by the bonding, and the ring will break one or more low-level bonds. MB Stable indicates whether the Meta-Bond is stable. For example, a type 5 reaction results in the low level structure M_1 changing, but the bond between M_1 and M_2 is stable and the low level structure of M_2 is stable. An example of a possible meta-molecule formed through a type 5 reaction is shown in Figure 8.

From Table 1 we can see that there is a symmetry between reaction types 3 and 5, and that there is also a symmetry between reaction types 4 and 6. Figure 8 is a pictorial repre-

¹github.com/iw596/Meta-AChem

Type	R1 Stable	R2 Stable	MB Stable
1	True	True	True
2	True	True	False
3	True	False	True
4	True	False	False
5	False	True	True
6	False	True	False
7	False	False	True
8	False	False	False

Table 1: Reaction types between two meta-atoms with a single bond between them.

sentation of Table 1, showing the resulting product of each reaction type.

Type 1 Reaction

A type 1 reaction results in a meta-molecule as there is stable bond between the two meta-atoms. The internal structure of both meta-atoms is unchanged by the reaction. This reaction type is analogous to a standard successful collision in the lower-level FN-SRBN AChem where two atomic particles react together to form a composite particles. Instead of atomic particles forming composite particles, now meta-atoms form meta-molecules.

Type 2 Reaction

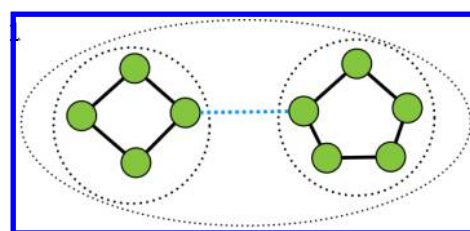
Type 2 reactions are elastic collisions between meta-atoms. This reaction results in no stable bond between the two meta-atoms and the internal structure of both meta-atoms are not altered by this elastic collision. This is analogous to an elastic collision in the lower level FN-SRBN AChem.

Type 3 & 5 Reactions

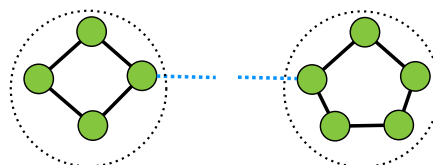
Both type 3 and type 5 reactions result in a meta-molecule consisting of two meta-atoms. One of the meta-atoms is unchanged by this bond but the other meta-atom is altered by the bond. The alteration to the meta-atom could be small, e.g a single low level bond breaking. The alteration could be large, e.g many atomic particles breaking away from the ring. This reaction type results in the previously inert lower-level structure becoming reactive. This means that it is now possible to have further reactions in the lower-level structure. With meta-atoms with more than one higher level bonding site, a type 3 or 5 reaction could result in a meta-molecule which is reactive at both the higher and lower levels.

Type 4 & 6 Reactions

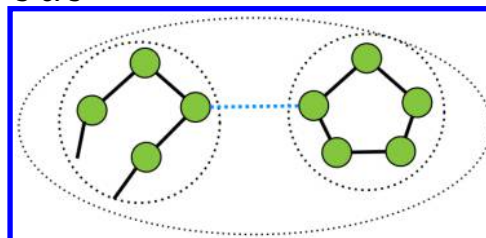
Both type 4 and 6 reactions result in the meta-atoms being temporarily bonded. This in turn results in one meta-atom being altered while the other is unchanged. This change one



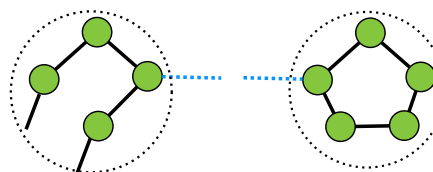
2



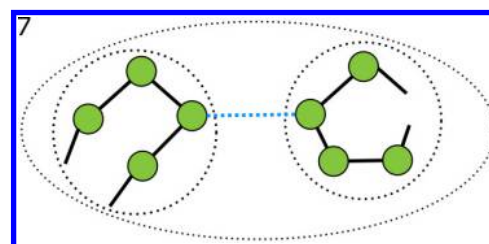
3 & 5



4 & 6



7



8

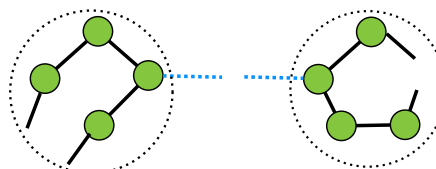


Figure 8: A potential product from each reaction type 1 to 8 listed in Table 1. Symmetric reaction types are grouped together.

meta-atom's structure causes the high-level bond between the meta-atoms to become unstable and break. A type 6 reaction is shown in Figure 9 in more detail. These reactions are interesting as they can be viewed as analogous to reactions between a reagent and a catalyst. The reagent is changed but the catalyst is not. Catalysts are thought to be important in the origin of life (Belmonte and Mansy, 2016; Russell, 2018), so having analogous behaviour in the Meta-Atom AChem shows that the AChem has some emergent properties which are potentially similar to real world systems. It further shows that this AChem has rich behaviour and warrants further exploration. These reaction types are a partially destructive collision, as no meta-molecule is formed and one of the meta-atoms has been altered by the reaction.

Type 7 Reaction

Type 7 reactions are similar to type 3 and 5 reactions in they result in a stable meta-molecule. The underlying structure here of both meta-atoms has been altered, and it is now possible to have further reactions in the lower-level structure of the meta-atoms.

Type 8 Reaction

A type 8 reaction is a fully destructive collision as the meta-atoms do not form a stable bond and both meta-atoms have their internal structure altered by the attempt to bond. This reaction type moves from having two meta-atoms that are inert at the lower level to having two meta-atoms that are reactive at both the lower and the higher level. This reaction type could be useful as it provides a way of creating very reactive meta-atoms, which could be used as a building block for generating complex meta-molecules.

Analysis

Our experimental tests have generated all of the reaction types in Table 1. These tests were performed using the set of rings shown in Figure 4. This shows that the Meta-Atom AChem has a rich behaviour as it exhibits all the possible reaction paths. These results also show that the AChem is capable of generating new structures.

Further tests, of bonding dangling tails rather than single dangling nodes, show that meta-atoms with dangling tails also can exhibit all the reaction pathways shown in Table 1. Reactions between meta-atoms that have two dangling nodes (or tails) on separate spikes result in products such as the one shown in Figure 10. One of the dangling nodes has bonded, which here has resulted in one of the rings in a meta-atom to break. Since there are still dangling nodes available, this meta-molecule could continue to bond with other meta-atoms to generate yet more complex products. With each meta-atom made up of composite particles, large composite particles consisting of many atomic particles can be generated in this way.

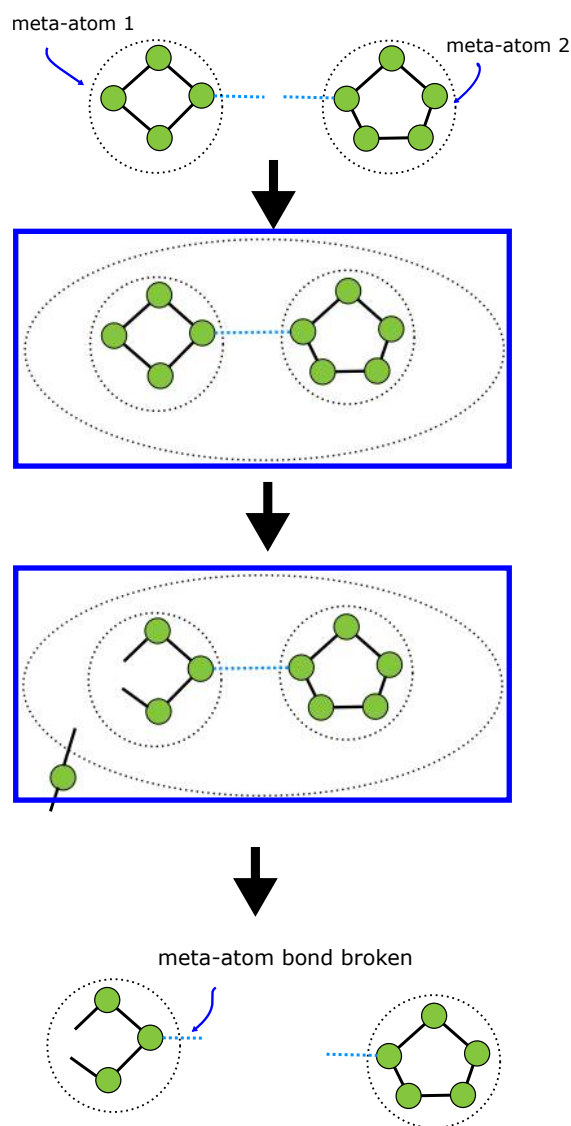


Figure 9: Reaction pathway for a type 6 catalytic reaction.

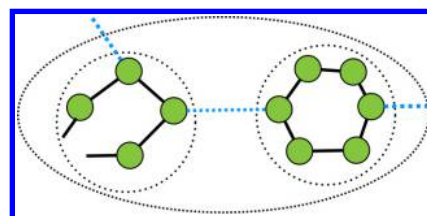


Figure 10: A meta-molecule with unbonded dangling nodes.

Conclusion & Outlook

We present a novel multi-level sub-symbolic AChem, Meta-Atom AChem, in which the structure and dynamics emerge from a lower level Spiky-RBN AChem (Krastev et al., 2016, 2017). Bonding in this Meta-Atom AChem exhibits a wide range of reaction pathways (Figure 8). Examples of these pathways include a catalytic reaction pathway. Meta-Atom AChem may be a way to analyse the structure, dynamics and stability of large composite particles.

Further analysis of bonding in the Meta-Atom AChem is required. For example, the distribution of reaction types between meta-atoms to determine the most frequent reactions is necessary. Furthermore, it needs to be investigated how the structure of a meta-atom affects the likelihood of a reaction pathway occurring. This would assist in the prediction of products from a reaction. To fully understand the dynamics of this new AChem, further work is also required in analysing the difference in strength between higher and lower-level bonds. Understanding this property would also aid in the prediction of products from a reaction.

Future work will involve extending the functionality of this AChem. Two main areas of interest are energetics and spatiality. By adding these features the Meta-Atom AChem may increase its richness.

Adding energetics could lead to new reaction pathways emerging. Energetics would modify the set of all possible reactions with a probability of occurrence at the current temperature. The effects of a reaction may cause a change in temperature. As the temperature varies, the probabilities vary, and so we would get a feedback coupling between reactions that occur and the probabilities of subsequent reactions.

Currently the AChem has no spatial components to bonding. Introducing emergent spatial properties to bonds could lead to interesting products being formed.

Additionally, further work will investigate if this high-level AChem could form the basis of another, even higher-level AChem. The possibility of using the Meta-Atom AChem as the basis of a higher level AChem will also need to be explored once functionality such as energetics are introduced. Emergent behaviours due to these extensions may occur and could be a natural property to form a higher level AChem.

Acknowledgements

This work was funded by a York Cross-disciplinary Centre for Systems Analysis (YCCSA) summer school studentship.

References

- Banzhaf, W. and Yamamoto, L. (2015). *Artificial Chemistries*. MIT Press.
- Belmonte, L. and Mansy, S. (2016). Metal catalysts and the origin of life. *Elements*, 12:413–418.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial Chemistries—A Review. *Artificial Life*, 7(3):225–275.
- Drossel, B. (2008). Random boolean networks. In Schuster, H. G., editor, *Reviews of Nonlinear Dynamics and Complexity*, chapter 3, pages 69–110. Wiley.
- Faulkner, P., Krastev, M., Sebald, A., and Stepney, S. (2018). Sub-symbolic artificial chemistries. In Stepney, S. and Adamatzky, A., editors, *Inspired by Nature*, pages 287–322. Springer.
- Gershenson, C. (2004). Introduction to random boolean networks. In Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., editors, *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173.
- Krastev, M., Sebald, A., and Stepney, S. (2016). Emergent bonding properties in the Spiky-RBN AChem. In *ALife 2016, Cancun, Mexico, July 2016*, pages 600–607. MIT Press.
- Krastev, M., Sebald, A., and Stepney, S. (2017). Functional grouping analysis of varying reactor types in the Spiky-RBN AChem. In *ECAL 2017, Lyon, France, September 2017*, pages 247–254. MIT Press.
- Russell, M. J. (2018). Green rust: The simple organizing ‘seed’ of all life? *Life*, 8(3):35.
- Wuensche, A. (1998). Discrete dynamical networks and their attractor basins. In *Complexity International*, pages 3–21.
- Wuensche, A. (2008). Basins of attraction in network dynamics: A conceptual framework for biomolecular networks. In *Reviews of Nonlinear Dynamics and Complexity*, chapter 13, pages 288–311. University Press, Chicago.

Pathways to Good Healthcare Services and Patient Satisfaction: An Evolutionary Game Theoretical Approach

Zainab Alalawi, The Anh Han, Yifeng Zeng and Aiman Elragig

School of Computing and Digital Technologies, Teesside University, United Kingdom, TS1 3BX
z.alalawi@tees.ac.uk, t.han@tees.ac.uk, y.zeng@tees.ac.uk, a.elragig@tees.ac.uk

Abstract

Spending by the UK's National Health Service (NHS) on independent healthcare treatment has been increased in recent years and is predicted to sustain its upward trend with the forecast of population growth. Some have viewed this increase as an attempt not to expand the patients' choices but to privatize public healthcare. This debate poses a social dilemma whether the NHS should stop cooperating with Private providers. This paper contributes to healthcare economic modelling by investigating the evolution of cooperation among three proposed populations: *Public Healthcare Providers*, *Private Healthcare Providers* and *Patients*. The Patient population is included as a main player in the decision-making process by expanding patient's choices of treatment. We develop a generic basic model that measures the cost of healthcare provision based on given parameters, such as NHS and private healthcare providers' cost of investments in both sectors, cost of treatments and gained benefits. A patient's costly punishment is introduced as a mechanism to enhance cooperation among the three populations. Our findings show that cooperation can be improved with the introduction of punishment (patient's punishment) against defecting providers. Although punishment increases cooperation, it is very costly considering the small improvement in cooperation in comparison to the basic model.

Introduction

The NHS is a free healthcare service at the point of delivery in the UK funded through taxpayers' contributions (Slawson, 2018). Many public healthcare services are currently allocating amounts of their budgets to sourcing services from Independent Healthcare Services Provider (ISP)/private sector. Motivated by the NHS's *Five Year Forward View* (FYFV), we choose the patient as the core focus of healthcare planning who is to be included in the decision-making process (England, 2014; Ham, 2017) for better health, patient care and financial sustainability. Several systematic studies looking into the improvement of clinical decision-making find that most patients expect to be informed about their situation and the treatment required, and play an important role in their clinical decision-making. However, little attention has been given to understanding the patients' role quantitatively as part of dynamic system modelling.

To investigate the dynamic system interactions among Public and Private health sectors and Patients, we resort to an Evolutionary Game Theory (EGT)-based solution. Researchers have applied EGT in a wide range of disciplines running the gamut from economics, politics and security, to ecology, mathematical biology, and computer science (Adami et al., 2016). EGT allows us to understand and analyze the complex healthcare system, interactions between individuals from various populations in a game, and how strategic behavior might evolve among individuals (Nowak, 2006a; Morgenstern and Von Neumann, 1953).

The main challenges of using EGT lie in formulating a valid payoff matrix and defining the ties between parameters within the proposed payoff matrix of each population. Some research on the evolution of cooperation has focused on the human willingness to engage in behavior that would involve paying a cost in return for imposing punishment on defectors or perceived wrongdoers (Sigmund et al., 2010; Dreber et al., 2008). Punishment is one mechanism that can enhance cooperation between individuals caught in social dilemmas (Hauert et al., 2007). Punishment can be seen applied in human society, such as by punishing free-riders (Sigmund et al., 2010), and in governance or institutional systems that impose rewards and punishments on agents (participants) (Andreoni et al., 2003). Different types of punishment can be implemented based on the structure of the played game: peer punishments, pool punishments and institutional punishments (Sigmund et al., 2010).

In this article, we consider the patient's role when developing an EGT decision-based model of healthcare services through a tripartite, one-shot EGT game (Gigerenzer and Gaissmaier, 2011; Elwyn et al., 2012). In our proposed model, we measure patients satisfaction and healthcare providers' reputation which impact the quality of healthcare services (Ham, 2017; Robertson et al., 2017). The model is aimed at identifying dynamic interactions that could enhance cooperation between the three populations. It will allow us to ascertain how selecting certain decisions within a dynamic system would influence patients' satisfaction with the provided service and their

willingness to cooperate. This could have important implications for addressing the significant, alarming drop in public satisfaction with NHS-provided services in recent years (Robertson et al., 2017). The social norm behavior of individuals is analyzed by applying EGT using one-shot game.

The dynamic of the game is computed through stochastic selection of strategies based on patients' satisfaction with received treatments and providers' reputations. The main question in this article is about how patients influence the dynamics of their cooperation in an EGT-based framework of three populations. The main contribution of this article is to build a basic model for the interactions of the three populations that closely capture the costs and benefits of every strategy combining decisions by agents in finite populations on either *cooperation* or *defection*. We further introduce peer punishment into the model: the patients' punishment which takes the form of complaints for clinical negligence (Bryden and Storey, 2011; Cooper et al., 2011).

The main contributions are summarized below.

- We develop a simple, yet expressive, basic mathematical model to formulate interactions among the three populations: Patients, Private and Public health sectors. This is the first dynamical model that captures the interactions of these populations in healthcare economic modelling research;
- We develop a mechanism to study the behavior of individuals in each population and extend the basic model by introducing punishments in their interactions;
- We analyze the models and examine how the relevant factors would influence cooperation among individuals within the populations;
- We conduct a comprehensive simulation analysis to determine various types of behavior most frequently adopted by individuals based on certain factors.

Related Work

The rapid development in research on the learning of social behavior has significantly increased our understanding of the dynamic interaction among individuals from different populations (Nowak, 2006a; Sigmund et al., 2010). Cooperation is one of the fundamental indicators to measure the strength and dynamism of a population (Smith, 1974; Kurokawa and Ihara, 2009; Encarnação et al., 2016). It can be studied by applying EGT using different types of mechanisms, such as reciprocal behaviors, mutual reciprocity among populations, replication, kin selection and costly punishment (Nowak, 2006b; Hofbauer and Sigmund, 1998). Researchers seeking to understand the behavior of different agent representations within the healthcare system use AI (Anh et al., 2013), game

theory (Brekke and Sjørgard, 2007), multi-agent systems (De and Gelfand, 2017) and big data (Murdoch and Detsky, 2013) to predict and understand behaviors within the system.

(Brekke and Sjørgard, 2007) argued that having a blurred line between the private and public healthcare providers within the NHS might lead to imbalances in the costs of provided health services and a drift towards privatisation. While (Wu et al., 2016) developed their proposed set of various non-cooperative and cooperative games for the Emergency Department response based on different types of patients. Another research investigates different dilemmas based on a three-population EGT framework involving the cost for prescribed antibiotics via healthcare providers (Bettinger, 2016). The main limitations are related to the actual cost paid for prescriptions and efficiency in quantifying incentives of patients for selecting the most satisfying or preferred provider. Another research by Encarnação (Encarnação et al., 2016) shows that the advent of the civil sector adds another layer of complexity to a scene that used to be dominated by two sectors: private and public.

Cooperation level is analysed by frequency-dependent selection of strategies within the populations (Kandori et al., 1993; Taylor et al., 2004). In this context, we seek to show how a drift towards a cooperate strategy in interactions between the three populations can be promoted by the adoption of the most dominant strategy (social behavior) in our proposed model. Based on the stochastic factors and processes associated with the healthcare model, this article intends to investigate evolving societal behavior between patients and different sectors in the healthcare dynamic system. The selection of the patient population in our model was made for the following reasons: finding the best behavior and strategy for decision-makers among the three populations; determining the impact of implemented peer punishments by a patient and how this social mechanism could influence the decision-making process for better cooperation; and, finding the best strategy for involved populations by computing the interaction between private and public sectors (Cooper et al., 2011; Brekke and Sjørgard, 2007).

A stochastic multi-objective auto-optimisation model was introduced by Bastian et al. to effectively manage resource allocation for the military health system with an eye to achieving a more efficient funding and staffing distribution between the Army, Air Force and Navy (Bastian et al., 2017). Bastian et al.'s research suffers from serious limitations; primarily, that the model introduced was not generic and depended on fixed inputs.

The Patient population in our model plays a major role (as discussed in the results) in influencing decision-making. Our analysis significantly improves our understanding of the model structure and the fac-

Parameters' description	Symbol
Reputation benefit for the Public and Private healthcare providers	b_R
Patient's benefit	b_P
Cost of investment spent by the Public/Private healthcare provider	c_I
Cost of treatment acquired by the healthcare provider	c_T
Cost of healthcare management	c_M
Extra Patient's benefit when both providers cooperate	ε

Strategies			Payoffs		
P1	P2	P3	Public	Private	Patient
C	C	C	$b_R - c_I - c_T$	$b_R - c_I - c_M$	$b_P + \varepsilon b_P$
C	C	D	$-c_I$	$-c_I$	0
C	D	C	$b_R - c_I - c_T$	0	b_P
C	D	D	$-c_I$	0	0
D	C	C	0	$b_R - c_I - c_M$	$b_P - c_T$
D	C	D	0	$-c_I$	0
D	D	C	0	0	$-c_T$
D	D	D	0	0	0

TABLE 1: The healthcare model (Public healthcare providers $P1$, Private healthcare providers $P2$ and Patient $P3$).

tors that lead to cooperation. It helps us answer important questions such as: What factors might influence the patient's rating of the healthcare services? What factors would induce healthcare providers to seek better reputations when the patient derives no or little benefit from the treatment (e.g. the patient files a complaint for receiving bad healthcare services).

EGT-Based Solutions

Basic Model and Extended Model

Model I - Basic model In this model we consider three populations: Public providers, Private providers and consumers/Patients. While Public represents the NHS or the Public healthcare providers, Private independent healthcare providers sell healthcare services, and Patients represents a person seeking treatment(s). An individual from each population (Public, Private and Patient) can choose from two strategies: provide/accept sustainable treatment(s) identified as cooperating, otherwise can't provide/refuse treatment(s) leading the patient to seek alternative treatment(s) from other providers. An agent's payoff is acquired based on the strategy played by each individual from the three populations, as explained in Table 1.

Every individual or agent in each of the three populations experiences one of the following scenarios based on two strategies namely, *cooperate* (C) and *defect* (D). This allows us to understand how cooperation evolves in altruistic interactions among individuals in a game. The following are the strategies an individual within each population can select (refer to Table 1):

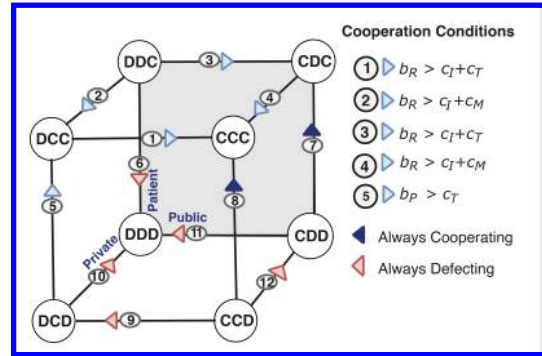


FIGURE 1: Evolution dynamics of the simplex's edges represents three-player actions (Basic model).

Public healthcare providers: ① (Cooperate, C) offers treatment paid for from taxpayers' money, in return gets a reputation benefit. (In case of cooperation, the public commits to invest (c_I) from allocated budget). ② (Defect, D) does not want to pay for the treatment.

Private healthcare providers: ① (Cooperate, C) offers treatment either paid by Public (when Public cooperates) or self-paid by Patient (so the main cost involved is represented by management cost (c_M)), and obtains a reputation benefit (b_R). In case of cooperation with the Patient, Private commits to invest (c_I) from its revenue. ② (Defect, D) does not want to offer the treatment.

Patient: ① (Cooperate, C) accepts the treatment and pays for the treatment (c_T) in the Private instance; Patient obtains health benefit (b_P) if one provider *cooperates* and an extra health benefit (εb_P) when both providers *cooperate*. ② (Defect, D) rejects the treatment and looks for alternative treatment mostly overseas.

The main issue we investigate in this article is the spending and cost effectiveness in the healthcare system with an eye to elucidating the social dilemma as mapped in Table 1. The dilemma questions the probability of cooperation (C) among the two sectors (Public and Private) and consumers (Patient) resulting in sustainable spending and cost effectiveness of the provision of treatment funded from the taxpayers' money¹.

When applying the evolutionary rules as explained in Fig. 1 on the matrix given in Table 1, the payoff is a simple representation of the healthcare cost-effectiveness, healthcare providers' reputation benefit and patient's benefit. The eight possible strategic scenarios are described below:

- Individuals from all three populations choose to *cooperate*, (CCC). In this case, the Public opts to pay

¹O'Connell in this <https://bit.ly/2VBY4im>, article talks about finding a structural change to reduce NHS spending and enhance the tendency towards saving.

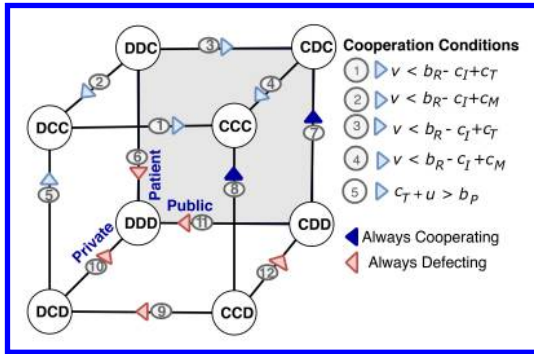


FIGURE 2: Evolution dynamics of the simplex's edges represents three-player actions (Extended model with patient's punishment).

for the treatment provided by the Private sector and the Patient accepts the provided treatment in pursuit of her/his own benefit of wellbeing and better health. The Public bears the costs of investment and pays for the Patient's treatment ($c_I + c_T$) from its allocated budget. It gains reputation benefits (b_R) derived from the treatment provided by the Private healthcare provider. Reputation benefits are derived from the Patient's satisfaction with the provided service. On the other hand, the Private sector will provide the required treatment to the Patient and receive the payment covering the costs from the Public. The cost of investment (c_I) of the Private healthcare provider is to invest in staffing and healthcare facilities, while (c_M) includes administrative and operational costs. Consequently, both the Private and the Public obtain a reputation benefit (b_R). Patient obtains extra health benefit (εb_P) as both healthcare providers are cooperating, where (ε) captures a fraction of the Patient benefit.

- Both the Private and Public healthcare providers want to pay and provide treatment to the patient, but the patient rejects the service (CCD). The payoff indicates that the Public healthcare provider will still invest (c_I) back in the Public as it is set to cooperate. Similarly, the Private sector only invests (c_I) back into its own resources, which include staffing, equipment and research, and gets nothing in return. The Patient's payoff is 0 as no treatment cost was involved; nor did she/he get health benefits from the services of the healthcare providers involved.

- The Patient accepts the treatment to be provided by the Private sector and paid for by the Public but the Private refuses to provide the treatment (CDC). The Public's payoff consists of the cost of investment and treatment ($c_I + c_T$), which are covered from its allocated budget, and a reputation benefit (b_R) is accumulated based on patients' satisfaction. The cost of accepting the treatment for the Patient is 0 and s/he obtains health benefit (b_P).

- The Public is the only party willing to provide treatment, but the Patient rejects the treatment offered (CDD). So, there is neither treatment cost nor

Parameters' description	Symbol
Patient's punishment cost	v
Fee paid by punished agent	u

Strategies			Payoffs		
P1	P2	P3	Public	Private	Patient
C	C	C	$b_R - c_I - c_T$	$b_R - c_I - c_M$	$b_P + \varepsilon b_P$
C	C	D	$-c_I$	$-c_I$	0
C	D	C	$b_R - c_I - c_T$	$-v$	$b_P - u$
C	D	D	$-c_I$	0	0
D	C	C	$-v$	$b_R - c_I - c_M$	$b_P - c_T - u$
D	C	D	0	$-c_I$	0
D	D	C	$-v$	$-v$	$-c_T - 2u$
D	D	D	0	0	0

TABLE 2: The healthcare model with Patient's Punishment (Public healthcare providers $P1$, Private healthcare providers $P2$ and Patient $P3$).

benefit returned. Hence, the payoff for both the Private healthcare provider and the Patient is 0, while the Public still bears a cost of investment (c_I) in new treatments (Sapiña et al., 2018).

- The Public healthcare provider opts out of providing treatment and refuses to pay for it; therefore, the payoff for the Public is 0 (DCC). As the Public does not want to provide and pay for the Patient's treatment, the Patient would look for treatment provided by a Private sector in a competitive market. The Private sector's payoff is derived from the gains it makes in its own reputation benefit (b_R); even when it is not cooperating with the Public. The payment received goes towards defraying the costs of investment and management ($c_I + c_M$). The Patient, on the other hand, gets health benefits and pays the cost of the treatment ($b_P - c_T$).

- Only the Private healthcare provider offers to provide treatment with a specified price tag while neither the Public wants to pay for the treatment nor the Patient is accepting the treatment (DCD). The payoff for the latter two agents is 0, while the transaction involves an investment cost (c_I) to be taken out of the Private provider's budget.

- None of the healthcare providers is willing to provide treatments (DDC). This situation leads the Patient to look for alternative treatments, both domestically and possibly overseas, and certainly the Patient has to pay for a treatment cost (c_T).

- The situation depicts all three agents choosing not to interact with one another (DDD). Hence, there will be no winner or loser and all agents get a 0 payoff.

Model II- Model with Patient's Punishment This model extends the basic version by introducing peer punishment. Patient has the choice to punish defected healthcare providers after a one-shot game has been played. Costly punishment means the patient pays a cost u to force the defected healthcare provider to pay a cost of clinical negligence v . The model payoff matrix is provided in Table 2. (Here we assume that, $u <$

v) (Fowler, 2005).

When an individual from healthcare providers fails to fulfill their commitment to provide acquired healthcare services up to standards, a patient's punishment would be meted out to the defecting individual(s) from healthcare providers v ; simultaneously a cost u is accrued from the patient in this process. Furthermore, if a co-player refuses to commit, the payoff for both is 0. The transition probability of a mutant playing strategy D to invade a population of C players can be measured following the method explained below.

Method: Evolutionary Dynamics for Three Populations

EGT method is adopted to study the evolutionary dynamics and interactions among individuals from three distinct finite populations: Public healthcare providers $P1$, Private healthcare providers $P2$ and Patient $P3$. Here, we assumed that the populations are of a fixed size N . Every individual in each population will be involved in one of the eight strategic scenarios as mentioned earlier. Individuals have the choice to *cooperate*, (C) or to *defect*, (D) in a paradigm shift fashion. In our proposed model, there are eight possible paradigms corresponding to the eight possible combinations of the basic strategies within the three populations, namely, CCC , CCD , CDC , CDD , DCC , DCD , DDC and DDD . Denoting the numbers of cooperators in $P1$, $P2$ and $P3$ by x , y , and z , respectively, the payoff of each strategy can be written as follows:

$$\left. \begin{aligned} P_s^{Public}(x, y, z) &= P_{syz} \\ P_s^{Private}(x, y, z) &= P_{xsz} \\ P_s^{Patient}(x, y, z) &= P_{xys} \end{aligned} \right\} \quad (1)$$

where P_{xyz} and $x, y, z \in \{0, 1\}$, is the payoff for the strategy selected by individuals from one of the stated populations, and (x, y, z) represents the selected strategies C or D . For instance, individuals from $P1$ have the options to play C or D . The selected strategy will replace the s at x vertex, while y and z vertices remain unchanged for every selected strategy for the Public population in this instance. The payoff of randomly selected individuals A and B in the population depends on the proportion of both players in the population. In each time step an individual B with fitness π_B imitates a randomly selected individual A with π_A fitness adopting pairwise comparison rule. The probability ρ that A adopts B 's strategy is given by the Fermi's function (Sigmund et al., 2010; Traulsen et al., 2007)

$$\rho = [1 + e^{-\beta[\pi_A - \pi_B]}]^{-1} \quad (2)$$

where the parameter $\beta \geq 0$ represents the 'intensity of selection' or 'imitation strength' ($\beta = 0$ represents neutral drift where imitation decision is random, while for large $\beta \rightarrow \infty$ the imitation decision is increasingly deterministic). In order to construct a symmetric matrix, the fitness of an individual

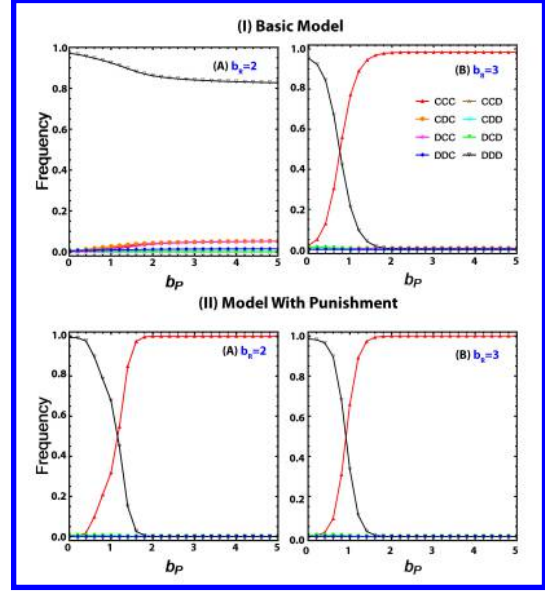


FIGURE 3: The plots represent the frequency of all strategies adopted by the populations. Panel (I) represents the frequency of all strategies adopted by the populations based on the basic model's matrix. Panel (II) examines the frequency of the model's with punishment strategies for varying (b_P and b_R) as stated, where: $u = 0.5$ and $v = 1.5$. Other parameters: $c_I, c_T, c_M = 1$, $b_R = 2$ and 3 , $\varepsilon = 0.2$, $N = 100$ and $\beta = 0.1$.

adopting a strategy s within a population is derived from the average obtained from the tripartite one-shot game described in Table 1. The social dynamics of the three finite populations interacting in eight strategies as a combination of C s and D s is represented by a death-birth process using pairwise comparison (Traulsen et al., 2006; Nowak, 2006a). Individuals with the highest payoff reproduce and their social behavior is adopted by weak opponents, i.e. the invading agent. For instance an agent B from the Public population imitates the successful strategic behaviour, C , by another regional healthcare provider as agent A . Subsequently, the transition matrix is evaluated by values given to the associated parameters as depicted in Table 1. At each time step, there is a probability of stochastic selection of an individual from a population whereby an individual B from one of the populations playing D with payoff f_D may imitate another randomly selected individual A with payoff f_C from the same population. The probability of the occurrence of this action (Nowak, 2006a; Kandori et al., 1993) is stated in the equation above. The transition probability drifts for agent A playing C from k to $k \pm 1$ is given by:

$$T^\pm(k) = \frac{k}{N} \frac{(N-k)}{N} [1 + e^{\mp\beta(\Pi_C(k) - \Pi_D(k))}]^{-1} \quad (3)$$

As mentioned earlier, the process has two absorbing states $k = 0$ and $k = N$. In mixed populations,

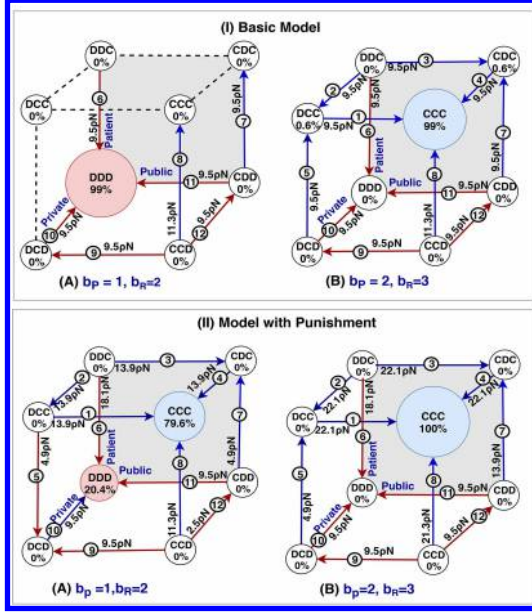


FIGURE 4: Stationary distribution and fixation probabilities. It illustrates the parameters values as stated for each simplex, whereas: red arrow represents transition towards defection, blue arrow transition towards cooperation and dashed line refers to neutral state. A high percentages 80% of individuals would adopt cooperation with lower b_P in (II-A) compared to (I-A), whilst an adequate 1% increase in cooperation with higher b_R in (II-B) compared to (I-B) is registered. The transition probability and frequency dependency are normalised ($1/N$), where $N = 100$. Other parameters: $c_I, c_T, c_M = 1, \epsilon = 0.2, N = 100$, and $\beta = 0.1$.

one of the absorbing states at the end would be a population with either all- C or all- D . Determining the different fixation probabilities $\rho_{D,C}$ is given by:

$$\rho_{D,C} = \left(1 + \sum_{i=1}^{N-1} \prod_{j=1}^i \frac{T^-(j)}{T^+(j)} \right)^{-1} \quad (4)$$

The transition matrix Λ with a set of $\{1, \dots, s\}$ strategy (Encarnação et al., 2016; Nowak, 2006a; Anh et al., 2013) is:

$$\Lambda_{ij, j \neq i} = \frac{\rho_{ji}}{3} \quad \text{and} \quad \Lambda_{ii} = 1 - \sum_{j=1, j \neq i}^s \Lambda_{ij} \quad (5)$$

The various fixation probability acquired from ρ_{ij} is that a population at a single state i transits to another state j when a mutant from one of the populations adopts an s different strategy. In games with large N , an invader can emerge as the stronger if the condition below is correct (Sigmund et al., 2010):

$$\sum_{i=1}^{N-1} \Pi_C(k) > \sum_{i=0}^N \Pi_D(k) \quad (6)$$

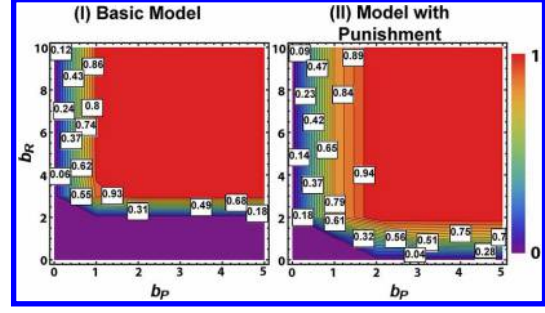


FIGURE 5: Frequency of strategy CCC for varying the main parameters b_R and b_P : (I) for the basic model, (II) extended model with costly punishment. In (II) $u = 0.5$ and $v = 1.5$. In general, CC performs better when punishment is introduced to the basic model. Additionally, significant cooperation is achieved for sufficiently low b_R and large b_P . Other parameters $c_I, c_T, c_M = 1, \epsilon = 0.2, N = 100$, and $\beta = 0.1$.

Results

Pathways to healthcare cooperation

In conducting a numerical and systematic analysis for the basic model, we focus on the social interactions between players in each population and how their decision influences the level of cooperation to sustain cost-effective services and better patient satisfaction. (Fig. 3 I) represents the computation investigating the frequency of adopting one of the eight strategies in the basic model by analysing the stochastic behavior of mutation in one of the three populations based on the frequency for each of the eight strategies given in (Table 1). That allows us to measure the ultimate behavior of those adopting the same strategy following the rules of social learning (Rendell et al., 2010).

In the basic model, refer to (Fig. 3 I-A), where b_R is small ($b_R = 2$), the defection strategy DDD is pervasive. The players of each of the three populations spend most of their time adopting defecting strategies rather than cooperating strategy. By simulating the matrix implementing (Eqs. 4 and 5) with a selected range of examined parameters (b_P and b_R), as stated in (Fig. 4), the analysis shows that the DDD strategy dominates the populations dynamic by 99%. In other words, healthcare providers have to invest more efforts in order to satisfy patients as better cooperation CCC is achieved when (b_R) is sufficiently high.

As has been observed in the basic model where punishment is absent, players of each population spend most of their time at defecting strategies (see Fig. 4 I-A&B). We started by pairwise computation of the interaction strategies in the payoff matrix (Table 2) based on different values of the parameters (b_R and b_P) to measure the stationary distribution and the frequency of the eight strategies. Recalling that in our model the patient has the option to mete out a costly punishment v to the defecting healthcare provider(s) at u cost (i.e. legal fees).

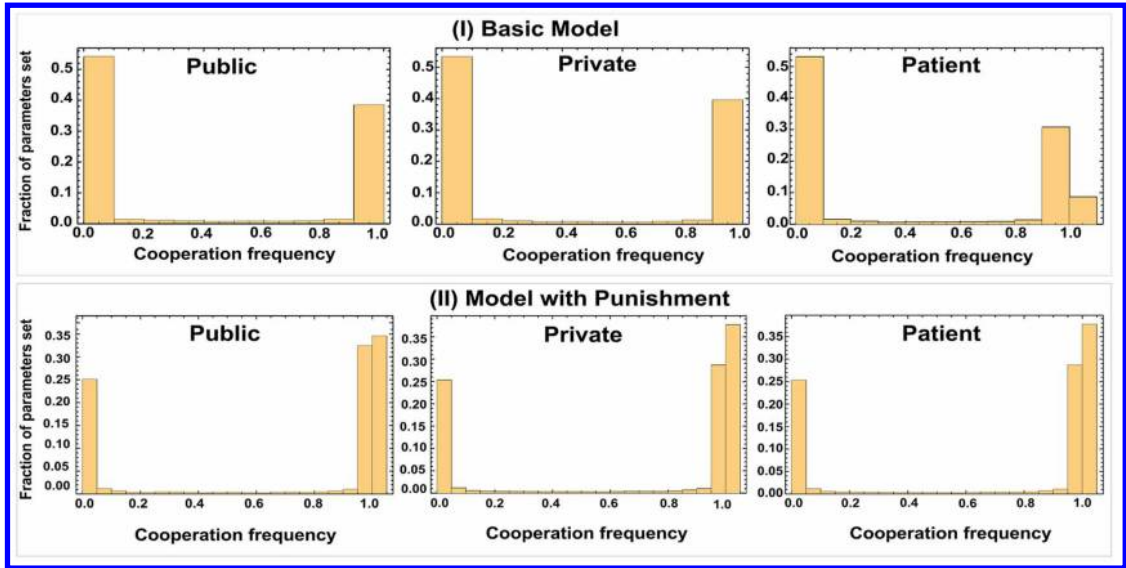


FIGURE 6: Robustness of the results across game configurations and parameters of the model with or without punishment (using 10000 samples). For the basic model: (I) Parameter values: $0.5 \leq b_P \leq 5$, $0.5 \leq b_R \leq 5$. (II) the model with punishment, where parameter values: $0.5 \leq b_P \leq 5$, $0.5 \leq b_R \leq 5$, $0 \leq u \leq 3$ and $u \leq v \leq 10u$ randomly sampling from uniform distributions on the intervals. Other parameters: $c_M, c_I, c_T = 1$ and $\varepsilon = 0.2$, $N = 100$, $\beta = 0.1$.

Most finite populations drift towards cooperation (i.e. *CCC*) as the most dominant strategy. In the presence of patient's punishment (see Fig. 4 II-A & B), cooperators invade defectors when there is a large enough patient's benefit (namely, $b_P > 1$), where the populations spend 100% of their time in cooperation. Our simulation suggests that, regardless of the reputation's benefit, cooperation is highly achievable (see Fig. 4 II).

Robustness of Parameters

A more compelling analysis has been carried out by computing 10000 samples with the set of parameters stated in (Fig. 6) to obtain the stationary distribution of cooperation for both the basic model and the extended model. Considering the collected results in the plot for the basic model in (Fig. 4), it is noticeable that the mean outcomes in (Fig. 6 I&II) are approximate.

Discussion and Conclusions

In summary, this paper investigates the behavior of individuals within the three finite populations to achieve better cooperation. By analyzing the performance of each population and its preferred path towards evolving and adopting a new strategy, we explored how to move towards cooperation while taking into consideration the cost of effectiveness and patient satisfaction with the provided services. Basically, the cooperation rules of each of the model's dynamics explained in (Fig. 1 & 2) clearly show that there are some paths moving toward full cooperation (see the panels in Fig. 1).

In contrast, avoiding the enveloped state (where some critics alluded to great fear of collaboration with private healthcare providers as 'the beast of the (P-Privatisation)') is represented by the *DDD* node; the public sector represents the main agent that can take the lead in changing the

rules and influence the behavior of the private sector and patient by introducing new policies. For different values of b_R and b_P , the frequency of strategy *CCC* has improved in the extended model (as apposed to the basic model), (see Fig. 5 A & B).

Moreover, the initial result yielded by the basic model is generally moving towards defection, apart from cases where a considerably high reputation value is imposed on the computation process, in which case there is a drift towards defection. In comparison, the results obtained when patient's costly punishment is introduced for the populations show a heightened tendency towards cooperation. This requires not only high patient's satisfaction but also a high reputation gained by healthcare providers; however, a full cooperation can be achieved within the basic framework on considerably low patient's satisfaction. In this case, the imposed peer punishment enhances cooperation with low patient's satisfaction, but it is proved too costly to be adopted in the proposed healthcare model. Realistic scenarios could be derived from some elective treatments such as hip replacements (Moscelli et al., 2016). In fact, data collected by the NHS and Care Quality Commission (CQC) (CQC, 2019), and The Patient Reports Outcome Measures (PROMs) (NHS Digital, 2019) are well represented in our proposed model in addition to some NHS statistics that substitutes for the cost of treatment and expenditures. We do recognise that our proposed models have some limitations, such as that our models are not validated against real healthcare data (it will be considered in future work) and the lack of comparison with other techniques. In this work, we only focus on analysing the behavior between agents using basic factors.

Future work will involve investigating the implementation of institutional punishment, expanding the model (i.e., by introducing new factors) to study the relation between the new elements and the dynamic behaviour of individuals. Finally, our proposed model directs attention to how the patient's decision would impact the process of collaboration between healthcare providers, and to the effectiveness of

management decisions made by the private sector in influencing the patient's choice of cooperation.

Acknowledgements

This work was supported by EPSRC project No. (EP/S011609/1) and The Anh Han is supported by Future of Life Institute (grant RFP2-154).

References

- Adami, C., Schossau, J., and Hintze, A. (2016). Evolutionary game theory using agent-based methods. *Physics of life reviews*, 19:1–26.
- Andreoni, J., Harbaugh, W., and Vesterlund, L. (2003). The carrot or the stick: Rewards, punishments, and cooperation. *American Economic Review*, 93(3):893–902.
- Anh, H. T., Pereira, L. M., Santos, F. C., and Lenaerts, T. (2013). Why is it so hard to say sorry? evolution of apology with commitments in the iterated prisoner's dilemma. In *IJCAI*, pages 177–183. AAAI Press.
- Bastian, N. D., Ekin, T., Kang, H., Griffin, P. M., Fulton, L. V., and Grannan, B. C. (2017). Stochastic multi-objective auto-optimization for resource allocation decision-making in fixed-input health systems. *Health care management science*, 20(2):246–264.
- Bettinger, B. M. (2016). *Game theory and mechanism design for cooperative competition dilemmas between healthcare providers*. Doctoral dissertation, Northeastern University.
- Brekke, K. R. and Sørsgard, L. (2007). Public versus private health care in a national health service. *Health economics*, 16(6):579–601.
- Bryden, D. and Storey, I. (2011). Duty of care and medical negligence. *Continuing Education in Anaesthesia Critical Care & Pain*, 11(4):124–127.
- Cooper, Z., Gibbons, S., Jones, S., and McGuire, A. (2011). Does hospital competition save lives? evidence from the english nhs patient choice reforms. *The Economic Journal*, 121(554):F228–F260.
- CQC (2019). Care Quality Commission for All Healthcare services in the UK. <https://bit.ly/2VkfKyx>.
- De, Soham; Nau, D. S. and Gelfand, M. J. (2017). Understanding norm change: An evolutionary game-theoretic approach. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1433–1441. International Foundation for Autonomous Agents and Multiagent Systems.
- Dreber, A., Rand, D. G., Fudenberg, D., and Nowak, M. A. (2008). Winners don't punish. *Nature*, 452(7185):348.
- Elwyn, G., Frosch, D., Thomson, R., Joseph-Williams, N., Lloyd, A., Kinnersley, P., Cording, E., Tomson, D., Dodd, C., Rollnick, S., et al. (2012). Shared decision making: a model for clinical practice. *Journal of general internal medicine*, 27(10):1361–1367.
- Encarnação, S., Santos, F. P., Santos, F. C., Blass, V., Pacheco, J. M., and Portugali, J. (2016). Paradigm shifts and the interplay between state, business and civil sectors. *Royal Society open science*, 3(12):160753.
- England, N. (2014). Five year forward view. Technical report, NHS England.
- Fowler, J. H. (2005). Altruistic punishment and the origin of cooperation. *Proceedings of the National Academy of Sciences*, 102(19):7047–7049.
- Gigerenzer, G. and Gaissmaier, W. (2011). Heuristic decision making. *Annual review of psychology*, 62:451–482.
- Ham, C. (2017). Next steps on the NHS five year forward view.
- Hauert, C., Traulsen, A., Brandt, H., Nowak, M. A., and Sigmund, K. (2007). Via freedom to coercion: the emergence of costly punishment. *Science*, 316(5833):1905–1907.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge University Press.
- Kandori, M., Mailath, G. J., and Rob, R. (1993). Learning, mutation, and long run equilibria in games. *Econometrica: Journal of the Econometric Society*, pages 29–56.
- Kurokawa, S. and Ihara, Y. (2009). Emergence of cooperation in public goods games. *Proceedings of the Royal Society of London B: Biological Sciences*, 276(1660):1379–1384.
- Morgenstern, O. and Von Neumann, J. (1953). *Theory of games and economic behavior*. Princeton University Press.
- Moscelli, G., Gravelle, H., Siciliani, L., et al. (2016). Market structure, patient choice and hospital quality for elective patients. Technical report, Centre for Health Economics, University of York.
- Murdoch, T. B. and Detsky, A. S. (2013). The inevitable application of big data to health care. *Jama*, 309(13):1351–1352.
- NHSDigital (2019). Finalised Patient Reported Outcome Measures (proms) in England for Hip and Knee Replacement Procedures (April 2017 to March 2018). <https://bit.ly/2J3Rq2h>.
- Nowak, M. (2006a). *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard University Press.
- Nowak, M. A. (2006b). Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563.
- Rendell, L., Boyd, R., Cownden, D., Enquist, M., Eriksson, K., Feldman, M. W., Fogarty, L., Ghirlanda, S., Lillierap, T., and Land, K. N. (2010). Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213.
- Robertson, R., Appleby, J., and Evans, H. (2017). Public satisfaction with the NHS and social care in 2016. *Results and trends from the British social attitudes survey*. London: King's Fund.
- Sapiña, L., Ligeró, S., and Domínguez, M. (2018). Protons for Jehovah's Witnesses? how press coverage of Ashya King's case brought proton beam therapy to the public sphere? (Padres secuestradores o empoderados? Como la cobertura mediática del caso Ashya King difundió la terapia de protones en la esfera pública). *Estudios sobre el Mensaje Periodístico*, 24(2):1773–1788.
- Sigmund, K., De Silva, H., Traulsen, A., and Hauert, C. (2010). Social learning promotes institutions for governing the commons. *Nature*, 466(7308):861.
- Slawson, S. (2018). NHS compensation payouts 'unsustainable', say health leaders. *The guardian*.
- Smith, J. M. (1974). The theory of games and the evolution of animal conflicts. *Journal of theoretical biology*, 47(1):209–221.
- Taylor, C., Fudenberg, D., Sasaki, A., and Nowak, M. A. (2004). Evolutionary game dynamics in finite populations. *Bulletin of mathematical biology*, 66(6):1621–1644.
- Traulsen, A., Nowak, M. A., and Pacheco, J. M. (2006). Stochastic dynamics of invasion and fixation. *Physical Review E*, 74(1):011909.
- Traulsen, A., Pacheco, J. M., and Nowak, M. A. (2007). Pairwise comparison and selection temperature in evolutionary game dynamics. *Journal of theoretical biology*, 246(3):522–529.
- Wu, C. K., Chen, Y. M., and Wu, D. (2016). A game theory approach for deploying medical resources in emergency department. In *Game Theory and Applications*, pages 18–30. Springer, Singapore.

On the Expected Number and Distribution of Equilibria in Multi-player Evolutionary Games

Manh Hong Duong¹ and The Anh Han²

¹ School of Mathematics, University of Birmingham, B15 2TT, UK. Email: h.duong@bham.ac.uk

² School of Computing, Media and Art, Teeside University, TS1 3BX, UK. Email: T.Han@tees.ac.uk

Evolutionary game theory (EGT) has become a powerful mathematical framework for the modelling and analysis of complex biological/economical systems whenever there is frequency dependent selection – the fitness of an individual does not only depend on its strategy, but also on the composition of the population in relation with (multiple) other strategies (Maynard Smith and Price, 1973; Hofbauer and Sigmund, 1998). The payoff from the games is interpreted as individual fitness, naturally leading to a dynamical approach. Random evolutionary games in which the payoff entries are random variables form an important subclass of EGT. They are necessary to model social and biological systems in which very limited information is available, or where the environment changes so rapidly and frequently that one cannot describe the payoffs of their inhabitants’ interactions (Fudenberg and Harris, 1992; Gross et al., 2009). As in classical game theory with the Nash equilibrium, see e.g. (McLennan, 2005), the analysis of properties of equilibrium points in EGT has been of special interest, see e.g. (Gokhale and Traulsen, 2010). These equilibrium points predict the composition of strategy frequencies where all the strategies have the same average fitness. In random games, due to the randomness of the payoff entries, it is essential to study statistical properties of equilibria. How to determine the distribution of internal equilibria in random evolutionary games is an intensely investigated subject with numerous practical ramifications in ecology, population genetics, social sciences, economics and computer science providing essential understanding of complexity in a dynamical system, such as its behavioural, cultural or biological diversity and the maintenance of polymorphism. Properties of equilibrium points, particularly the probability of observing the maximal number of equilibrium points, the attainability and stability of the patterns of evolutionarily stable strategies have been studied recently (Gokhale and Traulsen, 2010; Han et al., 2012; Gokhale and Traulsen, 2014). However, as these papers used a direct approach that consists of solving a system of polynomial equations, the mathematical analysis was mostly restricted to evolutionary games with a small number of players, due to the impossibility of solving general polynomial

equations of a high degree.

In this extended abstract, we present a summary of our recent works (Duong and Han, 2015, 2016; Duong et al., 2018b,a), in which we analyze random evolutionary games with an arbitrary number of players. The key technique that we develop is to connect the number of equilibria in an evolutionary game to the number of real roots of a system of multi-variate random polynomials (Bharucha-Reid and Sambandham, 1986; Edelman and Kostlan, 1995). Assuming that we consider d -player n -strategy evolutionary games, then the system consists of $n - 1$ polynomial equations of degree $d - 1$:

$$\sum_{\substack{0 \leq k_1, \dots, k_{n-1} \leq d-1, \\ \sum_{i=1}^{n-1} k_i \leq d-1}} \beta_{k_1, \dots, k_{n-1}}^i \binom{d-1}{k_1, \dots, k_n} \prod_{i=1}^{n-1} y_i^{k_i} = 0,$$

for $i = 1, \dots, n - 1$. Here $\beta_{k_1, \dots, k_{n-1}}^i := \alpha_{k_1, \dots, k_n}^i - \alpha_{k_1, \dots, k_n}^{i_0}$ where $\alpha_{k_1, \dots, k_n}^{i_0} := \alpha_{i_1, \dots, i_{d-1}}^{i_0}$ is the payoff of the focal player and $k_i, 1 \leq i \leq n$, with $\sum_{i=1}^n k_i = d - 1$, is the number of players using strategy i in $\{i_1, \dots, i_{d-1}\}$. In (Duong and Han, 2015, 2016), we analyze the mean number $E(n, d)$ and the expected density $f(n, d)$ of internal equilibria in a general d -player n -strategy evolutionary game when the individuals’ payoffs are *independent, normally distributed*. We provide computationally implementable formulas of these quantities for the general case and characterize their asymptotic behaviour for the two-strategy games (i.e. $E(2, d)$ and $f(2, d)$), estimating their lower and upper bounds as d increases. For instance, under certain assumptions on the payoffs, we obtain

- Asymptotic behaviour of $E(2, d)$:

$$\sqrt{d-1} \lesssim E(2, d) \lesssim \sqrt{d-1} \ln(d-1).$$

As a consequence,

$$\lim_{d \rightarrow \infty} \frac{\ln E(2, d)}{\ln(d-1)} = \frac{1}{2}.$$

- Explicit formula of $E(n, 2)$: $E(n, 2) = \frac{1}{2^{n-1}}$.

For a general d -player n -strategy game, as supported by extensive numerical results, we describe a conjecture regarding the asymptotic behaviours of $E(n, d)$ and $f(n, d)$. We also show that the probability of seeing the maximal possible number of equilibria tends to zero when d or n respectively goes to infinity and that the expected number of stable equilibria is bounded within a certain interval.

In (Duong et al., 2018b) we generalize our analysis for random evolutionary games where the payoff matrix entries are *correlated* random variables. In social and biological contexts, correlations may arise in various scenarios particularly when there are environmental randomness and interaction uncertainty such as in games of cyclic dominance, co-evolutionary multi-games or when individual contributions are correlated to the surrounding contexts (e.g. due to limited resource)(Szolnoki and Perc, 2014; Santos et al., 2012). We establish a closed formula for the mean numbers of internal (stable) equilibria and characterize the asymptotic behaviour of this important quantity for large group sizes and study the effect of the correlation. The results show that decreasing the correlation among payoffs (namely, of a strategist for different group compositions) leads to larger mean numbers of (stable) equilibrium points, suggesting that the system or population behavioral diversity can be promoted by increasing independence of the payoff entries.

As a further development, in (Duong et al., 2018a) we derive a closed formula for the distribution of internal equilibria, for both normal and uniform distributions of the game payoff entries. We also provide several universal upper and lower bound estimates, which are independent of the underlying payoff distribution, for the probability of obtaining a certain number of internal equilibria. In addition, the asymptotic behaviour of the probability of having no internal equilibria is then obtained (Can et al., 2018). The distribution of equilibria provides more elaborate information about the level of complexity or the number of different states of biodiversity that will occur in a dynamical system, compared to what obtained with the expected number of internal equilibria.

In short, by connecting EGT to random polynomial theory, we have achieved new results on the expected number and distribution of internal equilibria in multi-player multi-strategy games. Our studies provide new insights into the overall complexity of dynamical systems, including biological, social and Artificial Life ones, as the numbers of players and strategies in an interaction within the systems increase. As the theory of random polynomials is rich, we expect that our novel approach can be extended to obtain results for other more complex models in population dynamics such as the replicator-mutator equation and evolutionary games with environmental feedback (Weitz et al., 2016).

References

Bharucha-Reid, A. T. and Sambandham, M. (1986). *Random polynomials*. Probability and Mathematical Statistics. Academic

Press, Inc., Orlando, FL.

- Can, V. H., Duong, M. H., and Pham, V. H. (2018). Persistence probability of a random polynomial arising from evolutionary game theory. [arXiv:1804.05908](https://arxiv.org/abs/1804.05908).
- Duong, M. H. and Han, T. A. (2015). On the expected number of equilibria in a multi-player multi-strategy evolutionary game. *Dynamic Games and Applications*, pages 1–23.
- Duong, M. H. and Han, T. A. (2016). Analysis of the expected density of internal equilibria in random evolutionary multi-player multi-strategy games. *Journal of Mathematical Biology*, 73(6):1727–1760.
- Duong, M. H., Tran, H. M., and Han, T. A. (2018a). On the distribution of the number of internal equilibria in random evolutionary games. *Journal of Mathematical Biology*.
- Duong, M. H., Tran, H. M., and Han, T. A. (2018b). On the expected number of internal equilibria in random evolutionary games with correlated payoff matrix. *Dynamic Games and Applications*.
- Edelman, A. and Kostlan, E. (1995). How many zeros of a random polynomial are real? *Bull. Amer. Math. Soc. (N.S.)*, 32(1):1–37.
- Fudenberg, D. and Harris, C. (1992). Evolutionary dynamics with aggregate shocks. *J. Econ. Theory*, 57:420–441.
- Gokhale, C. S. and Traulsen, A. (2010). Evolutionary games in the multiverse. *Proc. Natl. Acad. Sci. U.S.A.*, 107(12):5500–5504.
- Gokhale, C. S. and Traulsen, A. (2014). Evolutionary multiplayer games. *Dynamic Games and Applications*, 4(4):468–488.
- Gross, T., Rudolf, L., Levin, S. A., and Dieckmann, U. (2009). Generalized models reveal stabilizing factors in food webs. *Science*, 325(5941):747–750.
- Han, T. A., Traulsen, A., and Gokhale, C. S. (2012). On equilibrium properties of evolutionary multi-player games with random payoff matrices. *Theoretical Population Biology*, 81(4):264 – 272.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge.
- Maynard Smith, J. and Price, G. R. (1973). The logic of animal conflict. *Nature*, 246:15–18.
- McLennan, A. (2005). The expected number of nash equilibria of a normal form game. *Econometrica*, 73(1):141–174.
- Santos, F. C., Pinheiro, F. L., Lenaerts, T., and Pacheco, J. M. (2012). The role of diversity in the evolution of cooperation. *Journal of theoretical biology*, 299:88–96.
- Szolnoki, A. and Perc, M. (2014). Coevolutionary success-driven multigames. *EPL (Europhysics Letters)*, 108(2):28004.
- Weitz, J. S., Eksin, C., Paarporn, K., Brown, S. P., and Ratcliff, W. C. (2016). An oscillating tragedy of the commons in replicator dynamics with game-environment feedback. *Proceedings of the National Academy of Sciences*, 113(47):E7518–E7525.

Entropy-Based Team Self-Organization with Signal Suppression

David W. King¹, Lukas Esterle² and Gilbert L. Peterson¹

¹ Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio, USA

² Aston University, Birmingham, UK and Aarhus University, DK
david.king@afit.edu, lukas.esterle@eng.au.dk, gilbert.peterson@afit.edu

Abstract

Self-organized and distributed control methods are increasingly important as they allow multi-agent systems to scale more readily than centralized control techniques. Furthermore, these methods increase system robustness and flexibility. In the online multi-object k -coverage domain studied here, a collective of autonomous agents must dynamically form sub-teams to accomplish two concurrent tasks: target discovery and coverage. Once a target is discovered, the collective of agents must create a sub-team of k -agents to cover the target. The work presented here introduces a novel, entropy-based task selection technique that incorporates signal suppression behaviors found in bee colonies. We test the technique in the online multi-object k -coverage domain while exploring three team properties: heterogeneity, team size, and sensor ranges, and their impact on multi-task accomplishment. Results show that signal suppression helps avoid over-provisioning of team resources to individual targets, dynamically creating sub-teams that simultaneously accomplish target discovery and coverage tasks.

Introduction

Computational systems composed of several independent subsystems often rely on central coordination to achieve shared goals. However, as the size of the system increases, centralized controllers become overwhelmed by increasing complexity, resulting in the need for decentralized control approaches. This becomes even more challenging when different tasks need to be tackled concurrently to achieve an overall goal. In order to meet this challenge, this work presents a novel self-organizing approach using an entropy-maximization task selection technique that incorporates signal suppression behaviors found in bee colonies. The technique enables autonomous agents to dynamically select and form teams to tackle different but complementary tasks. These teams are formed exploiting local knowledge gained from on-board sensors and information shared among nearby agents. At the same time, the suppression signal limits information sharing to a specific number of agent required to accomplish the task.

The authors test the technique in a version of the online multi-object k -coverage problem (Esterle and Lewis, 2017).

In this domain, all agents can sense targets within their respective sensing range. If an agent has a target within its sensing range, it may elect to follow it, once followed, the target is considered *covered*. Together, the collective of agents aims to maximize the number of targets covered by k agents.

The chosen domain presents several challenges. One, the set of targets is not fixed and may change over time. This means targets can arise or disappear at any moment, creating the need for agent robustness and flexibility. Two, agents do not know the number or locations of targets within the global environment, constraining agent knowledge. Three, both targets and agents are not static, moving freely through the environment, challenging the agent collective to continuously find and cover targets.

This problem gives rise to an interesting agent dilemma: should an agent follow a target in order to maximize the number of agents covering it or continue to search through the space in order to increase the probability of detecting new targets. We aim to overcome this dilemma by introducing multiple teams for different purposes. First, we divide the agents into two teams where the first team's emphasis is on target discovery while the second team focuses on coverage of those targets. Second, the covering agents dynamically team up to ensure each target is provisioned with k agents. The relation between the team sizes and the amount of targets and its effect on the overall team performance is previously unexplored. Therefore, this work also studies the impact of three team and agent properties: heterogeneity, team size, and sensor ranges, on team performance in accomplishing target detection and coverage goals.

Specifically we are interested in the following three research questions:

1. Can an entropy-maximization technique, aided by bee colony based signal suppression, be used to inform a collective autonomous decision making process, where agents decide whether to cover a target or continue searching for other targets? Furthermore, what is the effect of such an approach on dynamic team formation and selection?

2. Is a homogeneous team of tracking agents, employed with target detection and coverage, more effective than heterogeneous teams composed of tracking and observer agents each tasked with different goals, i.e. detection and coverage of targets over time?
3. What is the impact of the communication and sensing ranges of individual robots on the achieved rates of detection and k - n coverage when employing homogeneous and heterogeneous teams?

The remainder of this paper is structured as follows. First we give an overview of related work and the state-of-the-art. Afterwards, we discuss the problem in detail and our methodology. Thereafter, we present and discuss the results and outcomes of our experiments. We finish the paper with a summary and an outlook on future work.

Related Work

This work investigates the potential benefits of team work through the lens of heterogeneous and homogeneous teams engaged in a two-dimensional, k - n coverage domain. The methodology presented here tests three team properties: an entropy-based agent decision function, group size, and sensor ranges, and their impact on goal accomplishment.

The k -coverage problem combines the Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) problem introduced by Parker and Emmons (1997) and the k -coverage problem associated with sensor networks proposed by Huang and Tseng (2005). The problem's main premise is for a population of agents to discover random targets in space and assign k agents to them, where k is a predetermined number of agents. Researchers have used various methods to try and solve the k -coverage problem. For example, Werger and Matarić (2001) assigned weights to each target based on the number of robots and targets in the environment, and agents would broadcast their eligibility to engage the target to allow for team coordination. However, these assignments occurred with *a priori* knowledge which is not realistic in real-world environments.

To overcome this limitation, Jung and Sukhatme (2006) proposed enabling robots to calculate target weights at runtime which allowed robots, who shared this information with the collective, to distribute themselves accordingly through the search space. Kolling and Carpin (2007) allowed agents to request help from others to create longer loiter times for agents over assigned targets. They also added a signaling feature where agents could request help in covering targets as they appeared.

In a similar manner, bee colonies use direct signaling to marshal help from other bees. As foragers return to the colony, they recruit other bees to help collect resources via vigorous waggle dances (Rajasekhar et al., 2017). In engineered systems, Tolba et al. (2016)'s agents used signaling to gather underwater search vessels towards a discov-

ered sunken target, while agents in Beard et al. (2002) used signaling to coordinate flight plans and rendezvous points. Similarly, in a search and rescue domain, robots in Jennings et al.'s (1997) experiments used signaling to ask others for help in accomplishing their tasks. Signaling in both natural and engineered systems serves as a coordination medium over which information, such as target locations and agent capabilities, are shared. In this work, agent signaling is used to inform, i.e. share target locations, as well as to influence other agent behaviors, akin to bees suppressing the waggle dances of other bees (Lakhtakia and Martín-Palma (2013)).

Theraulaz et al. (1998) suggested that individual agents should both be able to autonomously select teams, as well as switch teams when necessary. Lewis et al. (2015) showed that team performance can be increased if individual agents change behaviors based on the current system state. Although Groß et al. (2008) argued against agent specialization, and showed that teams could accomplish goals without it, many natural systems do evolve specialized agents.

Wilson (1979) theorized that environmental pressures created the conditions for the emergence of specialized castes in ant colonies. It is logical to assume that such specialization serves the needs of the ant colony in a manner that is worth the energy required to create different types of ants in the same colony. This work tests the effectiveness of heterogeneous teams, i.e. teams with specialized agents, and homogeneous teams, i.e. teams without specialized agents. Agent behaviors rely on both agent suppression signals as well as an entropy-based decision process.

Using entropy to guide agent decisions is a relatively recent approach in multi-agent systems. Agents in Parunak and Brueckner (2001), Wissner-Gross and Freer (2013), and Mann and Garnett (2017), used entropy calculations to guide immediate agent decisions. Each showed that some measure of system entropy could enable dynamic agent coordination and team building. King and Peterson (2018) used entropy maximization to guide target selection in an Unmanned Aerial Vehicle (UAV) search domain. Similar to this work, simulated UAVs searched a 2D space for moving targets. UAVs would swarm to detected targets within sensor range; however, the UAVs were not held to specific k - n coverage demands. Instead, agents evenly divided themselves among available targets by polling local neighbors and selecting the target that resulted in the highest increase in local entropy. As even task coverage is inadequate for the k - n coverage problem in this work, a suppression signal - a behavior inspired by bee colonies - was incorporated into the agent decision process.

The suppression behavior creates a dynamic decentralized-control mechanism to prevent overprovisioning. Unlike previous works, such as Hefeeda and Bagheri (2007), Fusco and Gupta (2009), and Liu et al. (2013), both agents and targets move through the space. As targets randomly appear, and disappear, *a priori* based

approaches, e.g. Elhoseny et al. (2018), cannot be used, as agent teams must adapt at run-time to detected targets. The environmental dynamics requires teams to self-organize into sub-teams to be able to meet both goals of k - n coverage and target detection. Dynamic team self-organization can be accomplished by specialized agents, or encoded behavior responses to local phenomena.

The methodology presented here extends previous work by incorporating entropy-maximization and agent signaling suppression techniques to create a decentralized-control mechanism. This mechanism allows dynamic team, and sub-team, formation in response to current environmental conditions. Additionally, it explores the potential benefits of specialized teams in a dynamic, k -coverage domain.

Problem Statement

At its core, this is a k -assignment problem coupled with resource constraints. Agents must position themselves in the environment to effectively detect a large number of targets whose appearance, time and position, are unknown *a priori*. Furthermore, once agents detect targets, they must meet agent-target distribution (i.e. k - n coverage) requirements while still positioning themselves to find, and cover, any additional targets that may appear.

Formally, given a set of targets, $O_t = \{o_1, o_2, \dots, o_n\}$, at time t and a set of agents embodied in mobile robots, $R = \{r_1, r_2, \dots, r_m\}$, agents must discover and provide a k - n coverage of targets who appear with a probability of $\rho = 0.60$ and duration of $\lambda = [500, INF)$. The point, $l_i(t) = (x_i, y_i)$, represents an agent's or target's location in 2D space at time t . All agents and targets move through the 2D space at a set velocity, $v_i(t)$, and heading, $\omega_i(t)$ with all agents being assigned a random initial heading at t_0 . Agents have a sensing range σ and may communicate within a set range χ .

Two types of target discovery and tracking agents exist: trackers and observers. Tracker agents can discover and track, i.e. cover, detected targets within sensor range, or those shared by observer agents. Observer agents can only discover targets. Once a target is discovered, Observer agents broadcast the location of the target to all agents within range. Tracker agents add the target to their target lists, and then decide whether to cover the target or not. Tracker agents never broadcast the location of any discovered targets to their neighbors.

We consider that an agent detects targets when they are within the sensing range of the robot, i.e. for a distance d_{ij} between a target i and an agent j and $d_{ij} < \sigma_j$. Further, $det_i = 1$ if $\exists j \in R(d_{ij} < \sigma_j)$ indicates that a target i has been detected by at least one robot. Agents are unaware of the total number of targets in the environment. They are only aware of targets within their sensor range, or those shared by observer agents. The first goal of the set of agents is to maximize the number of detected targets in the environment:

$$det_{max} = \max \sum_{i=1}^n det_i \quad (1)$$

The second goal of our agents is to maximize the number of targets being covered by k agents:

$$kcov_{max} = \max \sum_{i=1}^n kcov_i. \quad (2)$$

We consider a target i covered $cov_{ij} = 1$ by a robot j if the agent decides to follow the target and $kcov_i = 1$ if $\sum_{j=1}^m cov_{ij} = k$ or 0, otherwise.

Entropy-based Team Self-organization

Our proposed approach aims to enable agents to achieve two goals: detect targets as they appear and provide k - n coverage of detected targets. Agents create k - n coverage of targets through entropy-maximization and signaling. When a target is detected, agents poll all agents within their communication range to create an agent-centric distribution of agents to targets. Using Shannon's entropy equation (Equation 3) (Shannon, 1948), agents select the target that creates the highest entropy value for their neighborhood.

$$H = - \sum_i p_i \log_2 p_i \quad (3)$$

Specifically, in Equation 3, p_i is calculated by dividing the number of known agents currently assigned to target i by the total number of agents within sensor range. Each agent adds itself to the number of agents assigned to both the target and population for each known target, resulting in a local entropy score for that target. The agent then selects the target that produces the highest resultant entropy score, i.e. maximizes its local entropy score.

The shortfall with this method is agents will create a uniform distribution across known targets which, for k - n coverage problems, is insufficient. To overcome this, agents already engaged in a target with maximized coverage send suppression signals to other agents within their communication range. When an agent receives a suppression signal, it drops that target from its detected list, allowing it to either engage other known targets, or continue to search for undiscovered ones.

Tracker, observer and target agents possess different update rules that follow Reynold's flocking method (Reynolds (1987)) with respect to separation and goal seeking behaviors. Tracker agents update their position by moving towards any detected targets, $g_i(t)$, while avoiding other tracking and observer agents, $sep_i(t)$ (4). Observer and target agents update their positions similarly, only adding the separation vector, $sep_i(t)$, to their update rule (5). Resultant vectors were capped by a max movement value of 0.688 pixels per time

Table 1: Control and independent variables.

Control:	
Visual Field	360 degrees
Velocity _{max}	0.688 pixels/time-step
Steering Force _{max}	0.50
Targets	10
Prob Target Appearing	0.60
Prob Target Disappearing	0.50
Independent:	
Tracker Team Size	5, 10, 15, 20, 25
Observer Team Size	0, 5, 10, 15
Tracker Sensor Range	100, 150, 200
Observer Sensor Range	200, 250, 300, 350

step. The behavior of an agent a_i at time t_i can be described as:

$$l_i(t+1) = l_i(t) + v_i(t) + sep_i(t) + g_i(t) \quad (4)$$

$$l_i(t+1) = l_i(t) + v_i(t) + sep_i(t) \quad (5)$$

$$\omega_i(t+1) = \omega_i(t) + \alpha_i(t) \quad (6)$$

Each agent autonomously updates their internal heading vector, $\omega_i(t)$, with respect to its desired heading, $\alpha_i(t)$, capped by a max steering force of 0.05 at each time-step.

Experiment

Experiments ran a mixed number of trackers, [5-25], and observers, [0-15], through 1,000 simulations of 5,000 time steps a piece. Teams without observers, i.e. observers = 0, were considered homogeneous. Heterogeneous teams were composed of any number of tracker and observer agents where the number of observers was greater than zero. Both tracker and observer agents were allowed to move freely about a 1,000 by 1,000 pixel grid. Tracker sensor and communication ranges were initialized at 50 pixel intervals [100,200] with observer sensor ranges spanning 50 pixel intervals [200-350]. Up to two targets from a ten target pool were randomly selected and placed into the environment every 500 time steps. Targets freely floated through the grid guided solely by one avoidance rule: keep a 25 pixel distance between it and any other object. All agents and targets were bounded by the grid. Initial positions for all agents and targets were randomly assigned. We uploaded a video, showing a single experimental run¹. See Table 1 for control and independent variables.

¹Video of simulation: <https://vimeo.com/337228864>

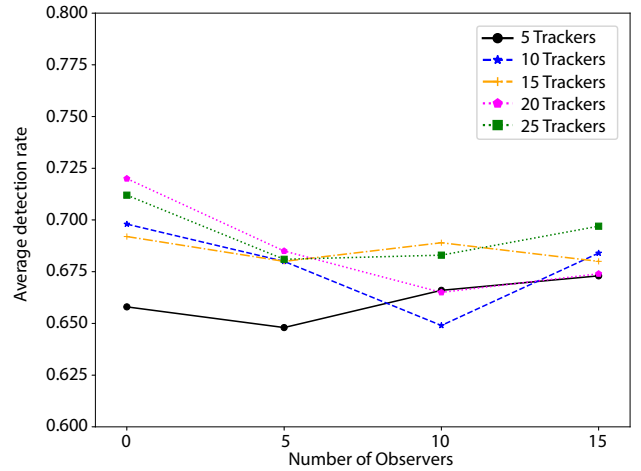


Figure 1: Average detection rates across all sensor ranges with teams of n trackers and m observers (x -axis), $k = 3$.

Results

Figures 1 and 2 show the average detection and coverage rates achieved by n tracker and m observer teams across all sensor ranges. Results indicate that the entropy maximization with signal suppression method creates dynamic teams capable of high target detection and coverage rates. Specifically, the method leads to dynamic, self-organization of sub-teams capable of providing the desired k - n coverage rates (i.e. $k = 3$ in this case). Furthermore, it becomes clear that the suppression signal used in our approach, prevents over-provisioning targets with more than k agents. However, distinct differences in team behaviors appear as the number of observers increases across all sensor ranges.

Results show a positive correlation between heterogeneous teams, i.e. those composed of specialized observer and tracker agents, and k - n coverage rates. However, as the number of observer agents increase in these teams, the detection rate immediately decreases. As more observer agents are added, a slow, positive increase in detection occurs, but never reaches the success of homogeneous teams in the space.

Although this result may be surprising, a closer inspection of heterogeneous team behavior explains the disparity. As observer agents locate targets and propagate target locations to tracker agents within sensor range, tracker agents begin clumping together. Information sharing ends up creating dense pockets of tracker agents pursuing targets in the same vicinity. The tracker agent sensor and communication range limitations cause a delay in suppression behaviors. The delay ends up altering the course of multiple tracker agents, resulting in multi-agent sensor overlap, decreasing the sensor range of the entire population. However, these pockets do result in higher coverage rates as more agents reside in an area to cover discovered targets.

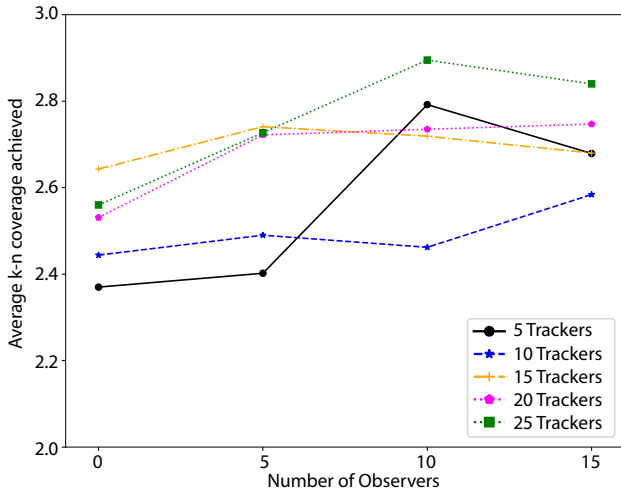


Figure 2: Average saturation rates across all sensor ranges with teams of n trackers and m observers (x-axis), $k = 3$.

Homogeneous teams, not sharing new target information, spread out across the grid, increasing the rate of task detection. Furthermore, signal suppression occurs immediately as sensor and communication ranges are equal for all agents, which avoids agent clumping. Homogeneous teams, however, do have the lowest coverage (i.e. saturation) rates, which makes sense given their population density is lower across the entire domain. The authors conclude that these behaviors create a trade-off between homogeneous and heterogeneous teams in the k - n domain.

From Figure 1, it is clear that heterogeneous teams do not approach the detection rates of homogeneous teams even when outnumbering homogeneous teams by a margin of 5:1. Coverage rates are a little more difficult to generalize as some heterogeneous teams peak with teams that contain 10 observers, while others, require less (e.g. teams with 15 tracking agents), or more (e.g. teams with 10 tracking agents). However, all experience a positive increase in coverage rates as the population size of observers increases. Notably, target coverage rates do approach the desired $k = 3$ rate. Theoretically, a pure 3:1 rate would require 30 tracker agents to cover 10 tasks; however, the results indicate that the desired 3:1 rate can be accomplished with fewer tracker agents. This shows that both the introduction of observer agents and entropy-based, dynamic team formation can reduce the number of tracker agents required to cover certain regions.

Table 2 presents the top three performing teams for detection and coverage tasks across all sensor ranges. Overall, homogeneous teams make up 47% of the highest performing detection teams with a 74.47% average detection rate compared to heterogeneous teams that achieved a 72.16% detection rate. This result again highlights the difficulty het-

Table 2: Top 3 teams and their detection and saturation rates across all sensor ranges. **TR** and **OR** represents sensing range of tracker and observer agents, respectively. **High 3 D** and **High 3 S** the three highest results for detection and saturation (k - n coverage), respectively. **T** and **O** stands for the number of used tracker and observer agents, respectively.

TR	OR	High 3 D	T	O	High 3 S	T	O
100	200	0.65	10	0	2.986	25	10
		0.64	10	5	2.963	15	10
		0.63	15	5	2.937	20	5
100	250	0.9	25	0	3.11	25	15
		0.75	25	15	2.801	25	10
		0.72	20	0	2.775	5	10
100	300	0.68	25	0	3.479	25	10
		0.67	20	10	3.066	5	10
		0.67	10	0	2.981	15	10
100	350	0.78	20	10	3.014	15	10
		0.78	20	0	2.93	25	10
		0.75	10	15	2.92	25	5
150	200	0.68	20	0	2.847	15	15
		0.67	25	10	3	25	10
		0.67	20	0	2.835	25	15
150	250	0.69	25	5	3.392	20	10
		0.69	25	0	3.073	20	15
		0.68	25	15	3.002	15	5
150	300	0.7	15	0	2.867	25	15
		0.69	25	0	2.815	25	0
		0.69	20	10	2.769	15	5
150	350	0.91	20	0	2.81	25	15
		0.8	10	5	2.793	20	15
		0.77	15	10	2.793	20	5
200	200	0.85	10	0	3.152	25	10
		0.75	25	15	3.131	15	10
		0.73	15	15	3.081	20	5
200	250	0.67	20	15	3.196	15	15
		0.66	25	15	3.007	15	5
		0.66	25	0	2.956	20	10
200	300	0.77	25	0	2.842	25	15
		0.76	15	0	2.792	20	15
		0.75	20	15	3.789	20	10
200	350	0.88	20	0	3	25	5
		0.85	15	10	2.976	25	10
		0.78	20	5	2.828	10	0

erogeneous teams have in balancing detection rates with required coverage costs. For coverage rates, heterogeneous teams comprise 94% of the top performing teams with an average 3.00 saturation rate, with teams of 25 tracking agents, with 10 to 15 observer agents, making up the majority of the top performers. These results further solidify the argument that heterogeneous teams, i.e. teams with specialized agents, can create a benefit to the entire population with respect to task completion.

In our next set of results, we used Boneabau’s Fixed Threshold Model (FTM) (Bonabeau et al., 1999) (Equation 7) to create a correlation coefficient for detection and saturation rates.

$$T_{\theta}(S) = \frac{s_i^n}{s_i^n + \theta_i^n} \quad (7)$$

Normally, agents use the FTM to decide which task to engage. It is used here for its mathematical properties to yield a correlation value along the interval, (0,1]. The stimulus parameter, s_i , represents the team detection rate with parameter, θ_i , representing the normalized team k - n coverage

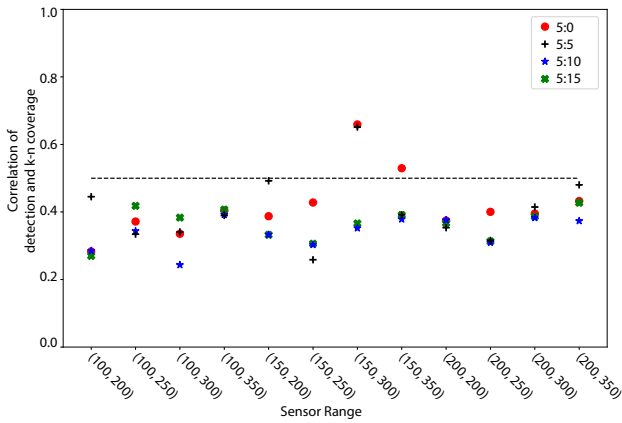


Figure 3: Correlation between detection and k-n coverage for mixed teams of 5 trackers and n observer agents across tested sensor ranges, with $k = 3$.

value. If detection dominates, then the FTM produces values close to 1, while if k - n coverage dominates, then the FTM produces values close to 0. If s_i and θ_i are close to equal, then the FTM produces values near 0.5. Once graphed, these values yield an intuitive understanding of how the team performed with respect to both detection and coverage tasks.

Figures 3-7 present correlation plots using Bonabeau's FTM function for all possible teams types across all sensor range combinations. These plots provide a measure of how well certain team combinations work towards both goals simultaneously, i.e. target detection and coverage. For example, in Figure 3, a heterogeneous team of 5 tracking agents and 5 observer agents with sensor ranges set to 150 and 200 pixels respectively provides an almost perfect balance (correlation = 0.500) of detection and coverage. These plots also provide insight on teams that excel at either target detection or coverage. Again from Figure 3, a team of 5 tracking agents and 20 observer agents with sensor ranges of 100 and 300 provides the best coverage rate, while a homogeneous team of 5 tracking agents with a sensor range of 150 provides the best detection rate.

A closer inspection of these correlation plots yields some interesting behaviors with respect to the difference in sensor ranges in heterogeneous teams. As far as detection rates, there is a general increase in detection rates as the difference between tracker and observer sensor ranges grows. Although this increased detection rate never exceeded the detection rates of the homogeneous teams, one could hypothesize that the detection rates of heterogeneous teams would eventually match, and possibly exceed, the homogeneous teams. Coverage rate results were more difficult to decipher.

In the 5 and 10 tracker team scenarios (Figure 3 and 4), correlation measures do not show any sensor range patterns other than a chaotic, oscillatory nature. However, starting at 15 tracker teams (Figure 5), a couple of heterogeneous

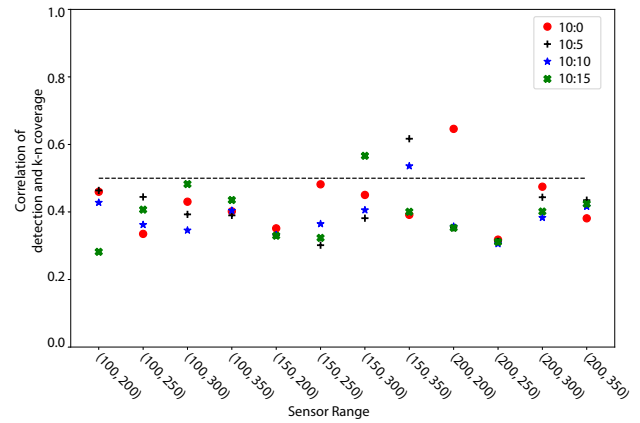


Figure 4: Correlation between detection and k-n coverage for mixed teams of 10 trackers and n observer agents across tested sensor ranges, with $k = 3$.

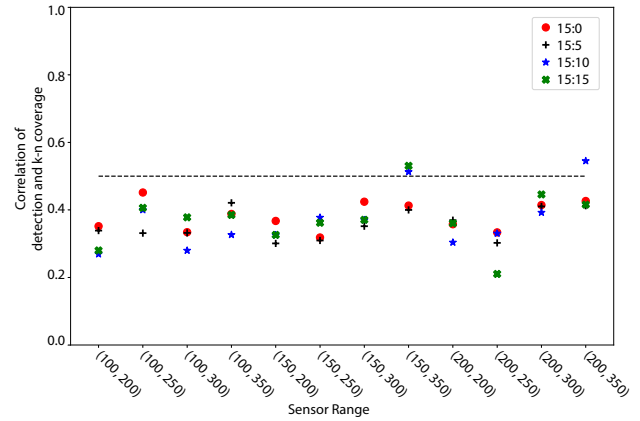


Figure 5: Correlation between detection and k-n coverage for mixed teams of 15 trackers and n observer agents across tested sensor ranges, with $k = 3$.

teams begin showing more stable behavior. In the 15 tracker team scenario, both the homogeneous 15 tracker team and the heterogeneous 15 tracker, 5 observer agent team exhibit small oscillatory behaviors but the highs and lows remain numerically close, i.e. fairly stable. Similarly, in the the 20 and 25 tracker team scenarios (Figures 6 and 7) both 20:5, 25:5, 20:15, and 25:15 heterogeneous teams exhibit the same, mostly stable, behavior across all sensor ranges. It is possible that larger teams are less affected by the differences in sensor ranges between tracker and observation agents, yielding more stable dynamics and behaviors, lending further validity to the conclusion that specialized teams are better than non-specialized ones in multi-task domains.

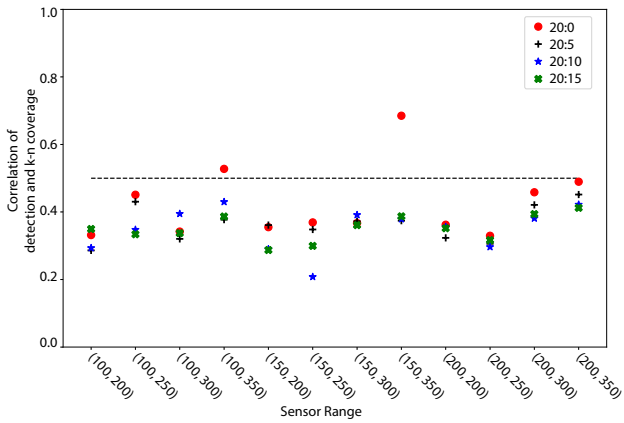


Figure 6: Correlation between detection and k - n coverage for mixed teams of 20 trackers and n observer agents across tested sensor ranges, with $k = 3$.

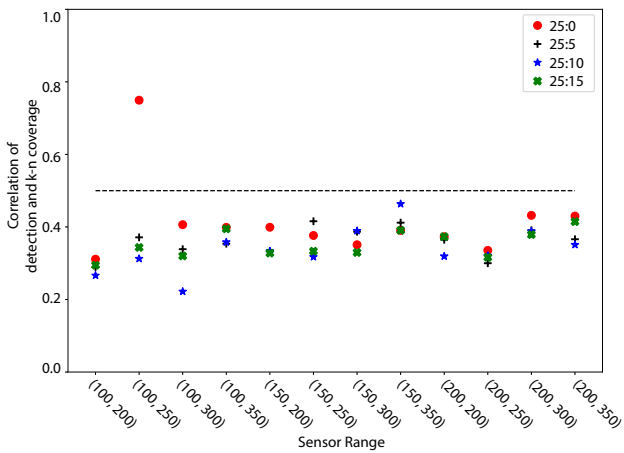


Figure 7: Correlation between detection and k - n coverage for mixed teams of 25 trackers and n observer agents across tested sensor ranges, with $k = 3$.

Discussion

The experimental results showed that homogeneous and heterogeneous teams excel in different tasks. Homogeneous teams were able to discover more targets on average than heterogeneous teams. In fact, almost all homogeneous teams experienced a decline in detection rates when observer agents were added to the team. Heterogeneous teams dominated the coverage task. The majority of top performing teams with the coverage task goal ($k = 3$) were heterogeneous with a mix of 25 tracking agents and 10 to 15 observer agents. These results indicate a trade-off between target discovery and coverage for heterogeneous teams.

As discussed earlier, heterogeneous teams create dense pockets of tracking agents, reducing the effective sensor range of the team in 2D space. However, these pockets lead

to better saturation rates over time as more tracking agents are able to move towards a newly discovered target. Additionally, as the agents are in closer proximity to one another, suppression signals keep coverage teams to the desired k - n ratios, in turn minimizing the wast of resources per target. As no clear team compositions dominated any others, we cannot conclusively state which teams would fare better in certain sensor range combinations, although we can conclude that heterogeneous robot teams outperform homogeneous teams in target coverage.

Results clearly show that agents using a local entropy maximization technique, coupled with a signal suppression strategy, produces robotic teams capable of high target detection and k - n coverage rates. Entropy-maximization based techniques lead to even agent distributions, showing that the signal suppression strategy is pivotal for the technique to reach the desired k - n coverage ratios. Other techniques, such as agent auctioning, would have to be tested against the entropy-maximization technique before one can draw any clear conclusions on effectiveness. Additionally, these future tests would need to track and compare the number of messages passed between agents to establish whether the technique is more efficient.

The impact of communication and sensing ranges of individual robots on the achieved detection and saturation rates appears mixed. All of the heterogeneous teams exhibited oscillatory behaviors with no apparent correlation to sensor range differences between tracker and observation agents. Although, at larger team populations, some heterogeneous teams showed less variability in their correlation values, meaning the amplitude of their oscillations were smaller. The authors can only conclude that sensor ranges increases are positively correlated to the accomplishment of both target detection and coverage. However, no correlations were found in the differences between tracker and observer agent sensor ranges.

Conclusion

The methodology presented here showed that agent specialization provided benefits only in the accomplishment of desired k - n coverage tasks. Homogeneous teams were able to outperform heterogeneous teams in detecting more targets. However, as the number of agents in the heterogeneous team grew, they closed the detection rate gap. These results indicate that a possible ratio of tracker to observer agents could exist in which heterogeneous teams are able to exceed homogeneous team accomplishment scores in both tasks.

Furthermore, experiments showed the validity of entropy-based agent decision functions. Entropy-maximization coupled with suppression signals created the conditions for the emergence of tracker agent sub-teams capable of both detecting targets and providing k - n coverage. Interestingly, teams using the entropy technique were able to meet the desired k - n coverage percentage (i.e. $k = 3$) without requiring

the theoretical population to achieve the 3:1, agent to task ratio, showing a possible resource efficiency.

Future work could start by comparing different self-organization algorithms and techniques with respect to not only task detection and coverage, but also the number of communication messages required for coordination. Other work could compare and contrast centralized versus decentralized techniques in a similar manner. Finally, alternative autonomous agent decision functions could be created and tested.

Acknowledgements

This work was supported by the SOLOMON project (Self-Organisation and Learning Online in Mobile Observation Networks) funded by the European Union H2020 Programme under grant agreement no 705020. Distribution Statement A: cleared for public release.

References

- Beard, R., McLain, T., Goodrich, M., and Anderson, E. (2002). Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. on Robotics and Automation*, 18(6).
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Elhoseny, M., Tharwat, A., Yuan, X., and Hassanien, A. E. (2018). Optimizing k-coverage of mobile wsns. *Expert Systems with Applications*, 92:142 – 153.
- Esterle, L. and Lewis, P. R. (2017). Online multi-object k-coverage with mobile smart cameras. In *Proc. of the Int. Conf. on Distributed Smart Cameras*, pages 1–6. ACM.
- Fusco, G. and Gupta, H. (2009). Selection and orientation of directional sensors for coverage maximization. In *Proc. of the Int. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 1–9.
- Groß, R., Nouyan, S., Bonani, M., Mondada, F., and Dorigo, M. (2008). Division of labour in self-organised groups. In Asada, M., Hallam, J. C. T., Meyer, J.-A., and Tani, J., editors, *From Animals to Animats 10*, pages 426–436. Springer.
- Hefeeda, M. and Bagheri, M. (2007). Randomized k-coverage algorithms for dense sensor networks. In *Proc. of the Int. Conf. on Computer Communications*, pages 2376–2380.
- Huang, C.-F. and Tseng, Y.-C. (2005). The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528.
- Jennings, J. S., Whelan, G., and Evans, W. F. (1997). Cooperative search and rescue with a team of mobile robots. In *Proc. of the Int. Conf. on Advanced Robotics*, pages 193–200.
- Jung, B. and Sukhatme, G. S. (2006). Cooperative multi-robot target tracking. In *Distributed Autonomous Robotic Systems 7*, pages 81–90. Springer.
- King, D. W. and Peterson, G. (2018). A macro-level order metric for self-organizing adaptive systems. In *Proc. of the Int. Conf. on Self-Adaptive and Self-Organizing Systems*, pages 60–69.
- Kolling, A. and Carpin, S. (2007). Cooperative observation of multiple moving targets: an algorithm and its formalization. *The Int. Journal of Robotics Research*, 26(9):935–953.
- Lakhtakia, A. and Martín-Palma, R. J., editors (2013). *Engineered Biomimicry*. Elsevier.
- Lewis, P. R., Esterle, L., Chandra, A., Rinner, B., Torresen, J., and Yao, X. (2015). Static, dynamic, and adaptive heterogeneity in distributed smart camera networks. *ACM Trans. Autonomous and Adaptive Systems*, 10(2):8:1–8:30.
- Liu, B., Dousse, O., Nain, P., and Towsley, D. (2013). Dynamic coverage of mobile sensor networks. *IEEE Trans. on Parallel and Distributed Systems*, 24(2):301–311.
- Mann, R. P. and Garnett, R. (2017). The entropic basis of collective behavior. *Journal of the Royal Society Interface*, 12.
- Parker, L. E. and Emmons, B. A. (1997). Cooperative multi-robot observation of multiple moving targets. In *Proc. of the Int. Conf. on Robotics and Automation*, volume 3, pages 2082–2089 vol.3.
- Parunak, H. V. D. and Brueckner, S. (2001). Entropy and self-organization in multi-agent systems. In *Proc. of the Int. Conf. on Autonomous Agents*, pages 124–130.
- Rajasekhar, A., Lynn, N., Das, S., and Suganthan, P. (2017). Computing with the collective intelligence of honey bees – a survey. *Swarm and Evolutionary Computation*, 32:25 – 48.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed model. *Computer Graphics*, 21(4):25–34.
- Shannon, C. (1948). A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423, 623–656.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). Response threshold reinforcements and division of labour in insect societies. *Proc. of the Royal Society B: Biological Sciences*, 265(1393):327–332.
- Tolba, S., Ammar, R., and Rajasekaran, S. (2016). Taking swarms to the field: Constrained spiral flocking for underwater search. In *Proc. of the Int. Symp. on Computers and Communications*.
- Werger, B. B. and Matarić, M. J. (2001). From insect to internet: Situated control for networked robot teams. *Annals of Mathematics and Artificial Intelligence*, 31(1):173–197.
- Wilson, E. O. (1979). The evolution of caste systems in social insects. *Proc. of the American Philosophical Society*, 123(4):204–210.
- Wissner-Gross, A. and Freer, C. (2013). Causal entropic forces. *Physical Review Letters*, 110(168702).

All in Good Team: Optimising Team Personalities for Different Dynamic Problems and Task Types

Soo Ling Lim^{1,2} and Peter J. Bentley^{1,2}

¹Braintree Research Lab, Braintree Limited, United Kingdom

²Department of Computer Science, University College London, United Kingdom
s.lim@cs.ucl.ac.uk

Abstract

Change is inevitable in this fast-moving world. As the environment and people's needs continuously change, so must the project. In our previous work, we developed an agent-based model of human collaboration that incorporates individual personalities. In this work, we applied a genetic algorithm to select the optimal personality combinations of a team in order to cope with different types of project change. We studied change in the context of three types of tasks: disjunctive (team performance is the performance achieved by the best performing individual), conjunctive (team performance is the performance achieved by the worst performing individual), and additive (team performance is the total performance of the group). Results reveal that different compositions of team personalities are suitable for different dynamic problems and task types. In particular, optimal personalities found for static problems differ from optimal personalities found for dynamic problems.

Introduction

Dynamically changing problems are a fact of life. Teams of people face change every day. Software developers frequently find themselves chasing a moving target as clients change their minds about features to be implemented (Lim & Finkelstein, 2011). Architects and builders must change their solutions as regulators decide there have been infringements of rules. Doctors must modify their treatments as illnesses follow unexpected courses. Engineers must change their processes as new technologies emerge. Companies must alter their products as markets change. People are remarkable in their adaptability and ability to cope with such change. Yet not all people are suited for all types of change. In every team, some may be better suited to cope with certain forms of change compared to others, and this may be related to their personalities.

In a pioneering study of group processes and productivity, Steiner (1972) found that tasks can be classified based on how individual contributions of members of a group are combined. For example, in *disjunctive* tasks, team performance is the performance achieved by the best performing individual (e.g., mathematicians proving a theorem), in *conjunctive* tasks, team performance is the performance achieved by the worst performing individual (e.g., a factory assembly line), and in *additive* tasks, team performance is the total performance of the group (e.g., a relay race or tug of war).

In our previous work, we developed an agent-based model of human collaboration and studied the effectiveness of different personalities at solving problems with different levels of noise (Lim & Bentley, 2018). We also used the model to investigate the effects of diversity in background and personality on team performance (Lim & Bentley, 2019).

In this work, we hypothesise that different combinations of personalities are needed for different dynamic problems and task types. We modify the model in order to simulate dynamic problems and types of task, and add a genetic algorithm in order to optimise the best team personalities to solve each task type and dynamic problem.

The rest of the paper is organised as follows. The next section describes the background, the section after that describes the agent-based model of human collaboration. Then we describe the modifications we made to the model for this work, followed by the experiments, results and conclusions.

Background

Agent-based models have been used to study the effects of human personalities in collaborative work, such as termites gathering food (Salvit & Sklar, 2012), ant colony (Ahrendt et al., 2015) and crowd movement (Durupinar et al., 2011). In these models, each agent is provided with a human personality, which determines how it behaves and interacts with other agents.

Agent-based models have been optimised in previous work using genetic algorithms (GAs). Heppenstall et al. (2007) used a GA to optimise an agent-based model of a retail market. They model petrol stations as agents and integrated additional system behaviour through the use of spatial interaction model. A GA is then used to optimise the model, producing results that match those derived by expert analysis through rational exploration.

Wang et al. (2009) used a GA to optimise partner selection for virtual enterprises that reduces their collaboration time and cost. They consider three types of collaborations (logistics, information transmission and capital flow) and two task allocation scenarios (allocating all tasks to one partner and allocation each task to different partners).

Knoester et al. (2013) used the AVIDA platform (Lenski et al., 2003) to study the evolution of consensus, a cooperative behaviour in which members agree on information sensed in

their environment. They used a form of evolutionary computation where a population of digital organisms is subject to instruction-level mutations and natural selection, and placed them into groups with fitness determined by their ability to perform consensus. Their experiments found that while genetic heterogeneity within groups increases the difficulty of the consensus task, a surprising number of groups were able to evolve this cooperative behaviour.

Lim and Bentley (2019) used a GA to investigate the effects of differing initial knowledge within team members as they collaborate on a shared task. The GA is used to evolve the optimal starting positions of each team member (representing their initial idea about the solution) and diversity is measured as the distance between their starting positions. The experiments found that diversity in team members' initial knowledge improves team performance, although teams with diverse personalities are more resilient to effects of diversity.

In this work we model teams collaborating to solve dynamically changing problems. Most research on dynamic optimisation has so far concentrated on tracking the moving optima as closely as possible (Jin, 2011). In practice, this is costly, if not impossible. To address dynamic optimisation problems more practically, Jin et al. (2013) introduced an optimisation algorithm that aims to find an acceptable (optimal or sub-optimal) solution that changes slowly over time, rather than the moving global optimum. A local approximation model is constructed using its neighbouring historical data in the database to estimate a solution's past performance.

In summary, despite related work touching on many aspects of this topic, there do not exist any agent-based models that attempt to understand which combinations of personalities are optimal for dynamic problems or for different kinds of task. Given that evolutionary algorithms have been shown to be effective at optimising agent-based models, we use this approach to optimise the agent-based model in this work.

Agent-based Model of Collaboration

The model used in this work, proposed in (Lim & Bentley, 2018), is designed to model human behaviour as team members collaborate to solve a shared task. The agent-based model uses a unique variant of particle swarm optimisation (PSO) to simulate the differing behaviours of people according to their personalities. The model has the following key abstractions:

- **Problem.** The shared goal of all agents is abstracted as the shared task to optimise a function (i.e., find the values of \mathbf{x} such that $f(\mathbf{x})$ is maximised).
- **Agent psychology.** The current "mental state" of each agent is modelled by giving it a position in the solution space (denoting the solution its mind has found so far), a velocity vector (denoting the direction and speed of its thought process), and acceleration vectors (representing the force of ideas and influences that modify the direction and speed of thought), the latter determined by its personality (behaviour algorithm).
- **Agent communication.** The distribution of information between agents is modelled as they each try to solve the

same problem. The exact type of information perceived by each agent and its use is determined by its personality.

Figure 1 shows the algorithm of the model, and the following sections describe each component in detail.

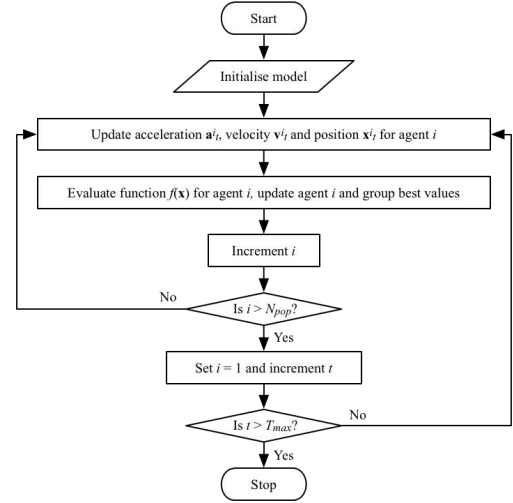


Figure 1: Algorithm of the model (Lim & Bentley, 2018).

Initialise

The model is initialised with:

- a problem space $\mathbf{D} \in \mathbb{R}^n$
- an objective function $f(\mathbf{x})$
- the number of timesteps T_{max} to run the model
- a population of agents N_{pop} , each agent $i \in \{1, \dots, N_{pop}\}$ is initialised with:
 - a personality type \mathbf{P}^i (one of the MBTI personality types, defined by a corresponding genotype (see later))
 - a random position $\mathbf{x}_0^i \in \mathbf{D}$: $\mathbf{x}_{min} \leq \mathbf{x}_0^i \leq \mathbf{x}_{max}$
 - a random velocity $\mathbf{v}_0^i \in \mathbb{R}^n$: $-\mathbf{v}_{init} \leq \mathbf{v}_0^i \leq \mathbf{v}_{init}$
 - personal best $f_{best}^i = f(\mathbf{x}_0^i)$
- Group best for timestep 0, $f_{best_0}^g$ (see later)

Update

For each timestep $t \in \{1, \dots, T_{max}\}$, each agent i 's position \mathbf{x}_t^i is updated using Equation 1:

$$\mathbf{x}_t^i = \mathbf{x}_{t-1}^i + \mathbf{v}_t^i \quad (1)$$

with the velocity \mathbf{v}_t^i calculated using Equation 2:

$$\mathbf{v}_t^i = \mathbf{v}_{t-1}^i + \mathbf{a}_t^i \quad (2)$$

If $|\mathbf{v}_t^i| > \mathbf{v}_{max}$, it is scaled to equal \mathbf{v}_{max} , in order to prevent excessive speed (an individual with high velocity would literally become too "set in their ways" and would find it impossible to change its direction of thought into a useful direction).

Acceleration \mathbf{a}_t^i is used to change the direction and speed of thought, as determined by the agent's personality – one of the

16 MBTI personality types (Myers, 1962). The interpretation provided here is designed to enable each personality to have an equally good chance of finding the solution. Interpretations were created to represent MBTI personality types appropriately and were not tuned in order to achieve any specific result in later experiments.

The MBTI consists of 16 personality types based on a person’s preferences on four opposing dichotomies: Extraversion (E) – Introversion (I), Sensing (S) – Intuition (N), Thinking (T) – Feeling (F), and Judging (J) – Perceiving (P) (Myers, 1962). J – P defines the person’s preferred manner (either S–N or T–F) of dealing with the outer world. Each personality type has a dominant Jungian function (more developed) supported by an auxiliary Jungian function (less developed) as shown in Table 1.

Table 2 defines the Jungian attitude, and perception and judgment functions. Each attitude (extraversion and introversion) is used as a source of information for each function (Thinking, Feeling, Sensing, Intuition), resulting in Jung’s eight psychological types: extraverted Thinking (Te), introverted Thinking (Ti), extraverted Feeling (Fe), introverted Feeling (Fi), extraverted Sensing (Se), introverted Sensing (Si), extraverted iNtuition (Ne), introverted iNtuition (Ni).

The Jungian intuitive functions (Ne and Ni) includes the notion of intuiting solutions, i.e., from sparse data they interpolate missing information, sometimes resulting in remarkable predictions (and sometimes not). This is modelled through a Gaussian process regression function which builds, from the data available to the agent, an internal imaginary view of the solution space for that agent. The agent then samples its imaginary space and is attracted to the area that it “believes” is a maximum. The Gaussian Progress Regressor from the Scikit-learn Python library (implemented based on Algorithm 2.1 in Rasmussen and Williams (1996)) is used with default options.

Acceleration \mathbf{a}^i_t is calculated using Equation 3:

$$\mathbf{a}^i_t = \mathbf{a}_j^i_t + \mathbf{a}_p^i_t \quad (3)$$

where $\mathbf{a}_j^i_t$ is the judging acceleration is calculated using Table 3 and $\mathbf{a}_p^i_t$ is the perceiving acceleration calculated using Equation 4:

$$\mathbf{a}_p^i_t = \sum_{j=1}^3 r_j (\mathbf{c}_j - \mathbf{x}^i_{t-1}) \quad (4)$$

where $r_1 = 0.5$, $r_2 = 0.3$, and $r_3 = 0.2$, and \mathbf{c}_1 , \mathbf{c}_2 and \mathbf{c}_3 are the top 3 candidates derived using Table 4 with $f(\mathbf{c}_1) \geq f(\mathbf{c}_2) \geq f(\mathbf{c}_3)$. In both tables, agent i ’s neighbours are defined as the five nearest agents to agent i measured by Euclidean distance, i.e., the peer group of each agent comprises those who share similar ideas to the agent. To ensure that the auxiliary component plays a lesser role compared to the dominant component, $\mathbf{a}_j^i_t$ is scaled down such that $|\mathbf{a}_j^i_t|^2 = \frac{|\mathbf{a}_p^i_t|^2}{2}$ if $|\mathbf{a}_j^i_t|^2 > \frac{|\mathbf{a}_p^i_t|^2}{2}$ (if \mathbf{P}^i has dominant perception and auxiliary judgment, otherwise vice versa).

Type	ISTJ	ISFJ	INFJ	INTJ
Dominant	Si	Si	Ni	Ni
Auxiliary	Te	Fe	Fe	Te
Type	ISTP	ISFP	INFP	INTP
Dominant	Ti	Fi	Fi	Ti
Auxiliary	Se	Se	Ne	Ne
Type	ESTP	ESFP	ENFP	ENTP
Dominant	Se	Se	Ne	Ne
Auxiliary	Ti	Fi	Fi	Ti
Type	ESTJ	ESFJ	ENFJ	ENTJ
Dominant	Te	Fe	Fe	Te
Auxiliary	Si	Si	Ni	Ni

Table 1: Myers-Briggs Type Table Showing the 16 Personality Types, with Dominant and Auxiliary Functions (Myers, 1962).

Attitude	Extraversion: Directs perception and judgment on outer world of people and things.
	Introversion: Directs perception and judgment on outer world of people and things.
Perception	Sensing: Concrete perception, finds interest in actualities (made aware directly through the senses), prefers to rely on objective, concrete facts.
	Intuition: Abstract perception, finds interest in connecting concepts and drawing parallels (made aware indirectly by way of the unconscious).
Judgment	Thinking: Analyses and determines the truth or falseness of information in an impersonal fashion, comes to conclusions based on a logical process, aimed at an impersonal finding (facts and ideas).
	Feeling: Person-centred assessment, comes to conclusions based on a process of appreciation, giving things a personal, subjective value.

Table 2: Jungian Attitudes, Perceptions, and Judgments (Jung, 1923).

Evaluate

Agent i ’s fitness at timestep t is evaluated as $f^i_t = f(\mathbf{x}^i_t)$. The agent’s personal best (f^i_{best}) and group best at timestep t ($f^g_{best_t}$) are evaluated as described in the next section.

Updates to the Model

In this work, we update the existing model by (1) modelling task types, (2) modelling dynamic problems, and (3) using a genetic algorithm to optimise team personalities.

Task Types

We model different task types by calculating group best $f^g_{best_t}$ at timestep t based on each task type as follows:

- **Disjunctive.** Group performance is the performance of its best member, $f^g_{best_t}$ is the best f^i_{best} , $i \in \{1, \dots, N_{pop}\}$. This is the scenario most commonly used in optimisation algorithms.

Function	Implementation
Te: The agent is influenced by its neighbours' best personal best. It accelerates towards its neighbours' best personal best from the previous timestep.	$\mathbf{a}_{Te}^i = \mathbf{x}_{n_{best_{t-1}}}^i - \mathbf{x}_{t-1}^i$ (5) where $\mathbf{x}_{n_{best_{t-1}}}^i$ is agent i 's neighbours' personal best position in the previous timestep that results in the highest $f(\mathbf{x})$, and \mathbf{x}_{t-1}^i is the agent's position in the previous timestep.
Ti: The agent focusses on its own personal best (the outcome of its own thoughts). It accelerates towards its own personal best, with randomness added to enable exploration.	$\mathbf{a}_{Ti}^i = (\mathbf{x}_{best_{t-1}}^i - \mathbf{x}_{t-1}^i) + \varphi$ (6) where $\mathbf{x}_{best_{t-1}}^i$ is agent i 's personal best position in the previous timestep, \mathbf{x}_{t-1}^i is the agent's position in the previous timestep, and φ is a random float in the interval $[-2.0, 2.0]$.
Fe: The agent "identifies with other agent's feelings" and "seeks harmony" by matching its neighbours' average velocity (direction of thought) from the previous timestep and to a lesser extent accelerates towards its neighbours' best personal best from the previous timestep.	$\mathbf{a}_{Fe}^i = \omega_1 \cdot \bar{\mathbf{v}}_{N_{t-1}}^i + \omega_2 \cdot \mathbf{a}_{Te}^i$ (7) where weights $\omega_1 = 0.8$, $\omega_2 = 0.2$, $\bar{\mathbf{v}}_{N_{t-1}}^i$ is agent i 's neighbours' average velocity in the previous timestep, and \mathbf{a}_{Te}^i is calculated using equation (5).
Fi: The agent "empathises with" its neighbours' ideas by accelerating towards its neighbours' average position from the previous timestep. It also cares about its own personal thoughts, so accelerates towards its own best position.	$\mathbf{a}_{Fi}^i = \omega_1 \cdot (\mathbf{C}_{n_{t-1}}^i - \mathbf{x}_{t-1}^i) + \omega_2 \cdot (\mathbf{x}_{best_{t-1}}^i - \mathbf{x}_{t-1}^i)$ (8) where weights $\omega_1 = 0.8$, $\omega_2 = 0.2$, $\mathbf{C}_{n_{t-1}}^i$ is the centroid (arithmetic mean position) of agent i 's neighbours' positions in the previous timestep.

Table 3: Jungian Judging Functions and Their Use in Calculating Judging Acceleration, \mathbf{a}_j^i (Lim & Bentley, 2018).

Function	Implementation
Se: The agent sees its neighbours' positions and their quality. Candidates are the positions of the agent's nearest neighbours in the previous timestep.	$\mathbb{C}_{Se}^i = \{\mathbf{x}_{n1_{t-1}}^i, \dots, \mathbf{x}_{n5_{t-1}}^i\}$ (9) where $\mathbf{x}_{n1_{t-1}}^i$ is agent i 's first neighbour's position in the previous timestep, and $\mathbf{x}_{n5_{t-1}}^i$ is agent i 's fifth neighbour's position in the previous timestep. The candidates for current and previous timestep \mathbb{C}_{Se}^i and \mathbb{C}_{Se}^i are then sorted in the order of decreasing $f(\mathbf{x})$.
Si: The agent remembers all its own previous positions and a few nearby points and their quality. Candidates are the agent's previous path and new points near to their position.	$\mathbb{C}_{Si}^i = \{\mathbf{x}_{t-1}^i, \dots, \mathbf{x}_0^i\} \cup \mathbf{P}$ (10) where \mathbf{P} is the set of points near to \mathbf{x}_{t-1}^i . Given $\mathbf{x}_{t-1}^i = (x_1, x_2, \dots, x_n)$, $\mathbf{P} = \{(x_1 + \delta, x_2, \dots, x_n), (x_1 - \delta, x_2, \dots, x_n), (x_1, x_2 + \delta, \dots, x_n), (x_1, x_2 - \delta, \dots, x_n), \dots, (x_1, x_2, \dots, x_n + \delta), (x_1, x_2, \dots, x_n - \delta)\}$ where δ is a random number from a normal distribution $N(\mu, \sigma)$ with $\mu = 1$ and $\sigma = 0.01$. The quality of old solutions is reduced as follows: every solution in the agent's previous path $\{\mathbf{x}_0^i, \dots, \mathbf{x}_{t-1}^i\}$ that are more than 10 timesteps old are reduced in quality by decrementing the fitness by 0.001 each timestep. The candidates for current and previous timestep \mathbb{C}_{Si}^i and \mathbb{C}_{Si}^i are then sorted in the order of decreasing $f(\mathbf{x})$.
Ne: The agent sees its neighbours' positions and uses them to create an "imaginary solution space". Candidates produced from Se (data from the environment) are used as input to train the Gaussian process regression function. Candidates are then the best quality solutions resulting from sampling this imaginary space.	$f^* = \mathcal{GP}: \text{train}(\mathbb{C}_{Se}, f(\mathbb{C}_{Se})); \text{predict}(\mathbb{C}_{Ne}^i)$ (11) where \mathcal{GP} is the Gaussian process regression function (Williams & Rasmussen, 1996), training on \mathbb{C}_{Se} and $f(\mathbb{C}_{Se})$, and \mathbb{C}_{Ne}^i is a vector of points in \mathbf{D} , sampled every 10 points. The candidates for current and previous timestep \mathbb{C}_{Ne}^i and \mathbb{C}_{Ne}^i are then sorted in the order of decreasing f^* .
Ni: The agent sees its own previous positions and a few nearby points and uses them to create an "imaginary solution space". Candidates produced from Si (internal data) are used as input to train the Gaussian process regression function. Candidates are then the best quality solutions resulting from sampling this imaginary space.	$f^* = \mathcal{GP}: \text{train}(\mathbb{C}_{Si}, f(\mathbb{C}_{Si})); \text{predict}(\mathbb{C}_{Ni}^i)$ (12) where \mathcal{GP} is the Gaussian process regression function, training on \mathbb{C}_{Si} and $f(\mathbb{C}_{Si})$, and \mathbb{C}_{Ni}^i is a vector of points in \mathbf{D} , sampled every 10 points. The candidates for current and previous timestep \mathbb{C}_{Ni}^i and \mathbb{C}_{Ni}^i are then sorted in the order of decreasing f^* .

Table 4: Jungian Perceiving Functions and Their Use in Getting Candidates (Top Three Candidates Returned as c_1, c_2 and c_3) (Lim & Bentley, 2018).

- **Conjunctive.** Group performance is the performance of its weakest member, $f_{best_t}^g$ is the worst $f_{best}^i, i \in \{1, \dots, N_{pop}\}$.
- **Additive.** Group performance is a sum of all individual performances, $f_{best_t}^g$ is the sum of all $f_{best}^i, i \in \{1, \dots, N_{pop}\}$.

Each agent’s personal best, f_{best}^i , is calculated as the agent’s best fitness in the last 10 timesteps, i.e., $\max(f_t^i, f_{t-1}^i, \dots, f_{t-8}^i, f_{t-9}^i)$. Retaining the memory of all best fitnesses for the entire run is not appropriate for dynamic problems where the optimal solution changes over time.

This work evaluates the performance of groups of agents as they collaborate to solve dynamic problems over time, thus fitness scores are obtained throughout the run. The final group performance f_{best}^{fg} (used as the fitness function for the GA) is calculated using Equation 13:

$$f_{best}^{fg} = \frac{\sum_{t=T_{poll}[0]}^{T_{poll}[4]} f_{best_t}^g}{5} \quad (13)$$

where T_{poll} is a list of timestep values and $f_{best_t}^g$ is the group best at timestep t .

Dynamic Problems

To model dynamic problems, we use a simple two-dimensional problem with a clear gradient as in (Lim & Bentley, 2018). The objective function $f(x, y)$ is described in Equation 14:

$$f(x, y) = -\sqrt{(x - a)^2 + (y - b)^2} \quad (14)$$

where agent i ’s position at time t , $\mathbf{x}_t^i = (x, y)$, the values of a and b are varied over time and $f(x, y)$ is normalised such that $f(x, y) \in [0, 1] : \forall x \in [x_{min}, x_{max}], \forall y \in [y_{min}, y_{max}]$. Figure 2 shows the heatmap and surface plot when $a = 0$ and $b = 0$.

We investigate the following types of change:

- **Static:** This is the baseline scenario with no change, where $a = 0$ and $b = 0$ for the entire duration.
- **Linear:** The position of the maximum moves from left to right on the x -axis. At $t = 0$, $a = -25$ and $b = 0$. The maximum starts at $(-25, 0)$ and a increments by 1 at every timestep, so at $t = T_{max}$, the maximum is at $(25, 0)$, see Figure 3. This models a simple “moving target” problem, for example a design specification that changes over time as the team tries to find the solution.
- **Oscillating:** The position of the maximum moves from left to right on the x -axis and returns to where it started. At $t = 0$, $a = -25$ and $b = 0$, and a increments by 2 at every timestep until $a = 25$ and $b = 0$, then a decreases by 2 at every timestep so at $t = T_{max}$, $a = -25$ and $b = 0$, see Figure 4. This models a problem where best solutions oscillate and repeat, e.g., in trading, sometimes it is good to buy, sometimes it is good to sell.
- **Rotary:** The position of the maximum moves 90° clockwise every 10 timesteps. At $t = 0$, $a = 0$ and $b = 50$, at $t = 11$, $a = 50$ and $b = 0$, at $t = 21$, $a = 0$ and $b = -50$, and so on, see Figure 5. This models a cyclic problem, e.g., designing gifts for different seasons throughout a year.

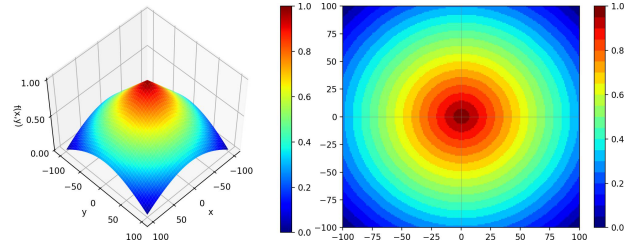


Figure 2: Surface plot (left) and heatmap (right) for normalised Equation 14 with a maximum in $(0, 0)$. Colour ranges from blue (minimum) to red (maximum).

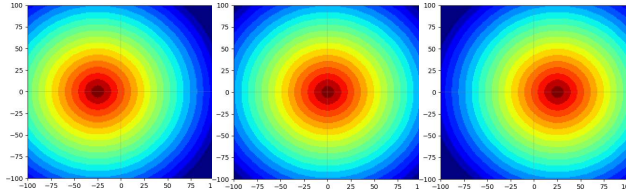


Figure 3: Linear at $t=0, 25$ and 50 .

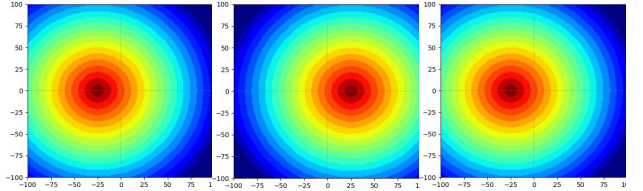


Figure 4: Oscillating at $t=0, 25$ and 50 .

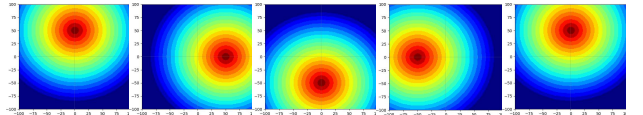


Figure 5: Rotary at $t=0-10, 11-20, 21-30, 31-40,$ and $41-50$.

Genetic Algorithm for Optimising Team Personalities

We used a genetic algorithm to evolve an optimal combination of agent personalities for each dynamic problem. For a team of $n=4$ members as used in the experiments, their personalities are specified by a 16-digit binary genotype. For example, the 16-digit genotype 1100000110001010 corresponds to the phenotype INTJ, ESTP, ISTJ, ISFJ.

A standard canonical GA is used with population size of 20, each individual solution representing the personalities of a team. Each team has four members, for each member of the population, fitness is calculated by decoding the genotype to produce four personalities for the team members. The agent-based model is run 50 times for the team to produce average team performance. The group best at the end of each run is recorded and team performance is measured by their average group best, which is the total group best for all runs divided by total number of runs. Based on fitness, 8 individuals are chosen as parents. 20 child teams are created using single-point crossover from the parents. A single bit flip mutation occurs with a probability of 0.2 per chromosome.

The GA is run for 20 generations. Values were found following preliminary experiments to determine fastest and most effective settings.

Experiments

Our experiments investigate each task type (Disjunctive, Conjunctive, Additive) with each dynamic problem (Static, Linear, Oscillating, Rotary) as illustrated in Table 5.

Static Disjunctive	Static Conjunctive	Static Additive
Linear Disjunctive	Linear Conjunctive	Linear Additive
Oscillating Disjunctive	Oscillating Conjunctive	Oscillating Additive
Rotary Disjunctive	Rotary Conjunctive	Rotary Additive

Table 5: Task types with dynamic problems.

For each experiment, we ran the GA ten times and analysed personalities selected by the GA. The model was initialised with constant settings in Table 6. For each task type and dynamic problem, we measured the number of times each personality is used by the GA to assemble teams. We also measured the average fitness over generations and average fitness over time for each task type and dynamic problem. Finally, we counted the opposing MBTI dichotomies used by the GA to assemble teams, i.e., Extraverts vs. Introverts (E vs. I), Sensors vs. Intuitives (S vs. N), Thinkers vs. Feelers (T vs. F), and Judgers vs. Perceivers (J vs. P).

Constants	Values
T_{max}	50
N_{pop}	4
v_{max}	5.0
x_{min}, y_{min}	-100
x_{max}, y_{max}	100
v_{init}	(1.0, 1.0)
T_{poll}	[10, 20, 30, 40, 50]

Table 6: Constants Settings for The Model

Results

Different personality types are selected at different frequencies as the best team compositions for different tasks and dynamic problems, as illustrated by the heatmap in Figure 6. The GA never chooses teams made from a single personality type, rather, it selects mixtures of different personalities to work in combination. The GA is able to optimise all types of task and dynamic problem, although disjunctive tasks are the easiest to optimise, with the highest average fitness over generations for all types of change, followed by additive tasks and conjunctive tasks (Figure 7). Figure 8 illustrates the average team performance over 50 timesteps for all the ten best teams selected by the GA. Good combinations of personalities are found, resulting in improvement over time by the agents as they solve each type of problem, even the difficult rotary problem, where performance drops when change occurs, but the teams are still able to gradually improve the performance over time (Figure 8).

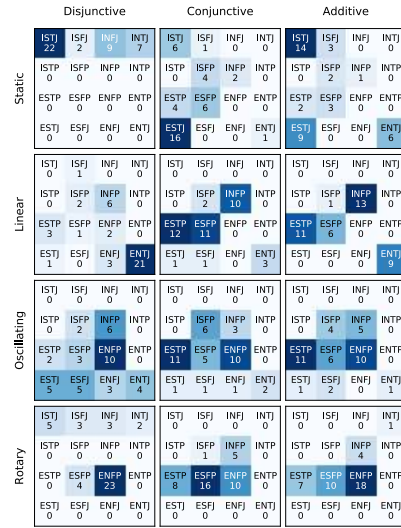


Figure 6: Heatmap of personality type in teams for dynamic problems and task types.

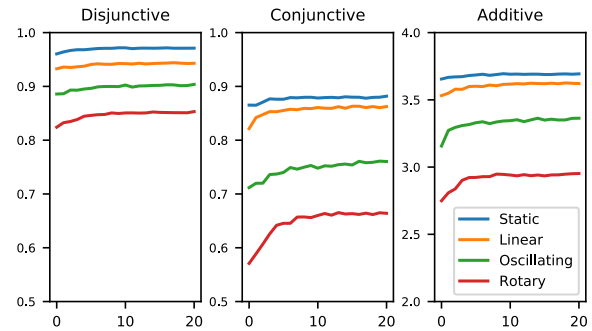


Figure 7: Average fitness over generations for dynamic problems and task types. x-axis is generations, y-axis is ff_{best}^g .

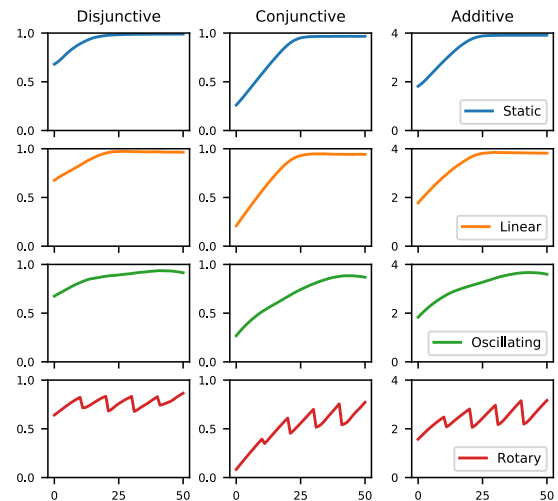


Figure 8: Average team performance over 50 timesteps for all 10 best teams selected by the GA for types of task and change. x-axis is timesteps, y-axis is ff_{best}^g .

Personalities best suited for static problems are different from those suited for dynamic problems (Figure 9). For all static problems, the GA evolved more Introverts compared to Extraverts, with Static Disjunctive exclusively composed of Introverts. For all dynamic problems, the GA evolved more Extraverts compared to Introverts. This is because Extraverted team members communicate more, which is essential when facing a moving target. The finding is supported by literature, where extraverted personalities has been found to be positively correlated to adaptability (Teixeira et al., 2012).

Similarly, for all static problems, the GA evolved teams with more Judgers compared to Perceivers, with Static Disjunctive exclusively composed of Judgers (Figure 9). For all dynamic problems, it evolved more Perceivers compared to Judgers, with Rotary Conjunctive being exclusively composed of Perceivers. Perceivers put more weight on the current state of their environment when making decisions, enabling them to detect and react to change, and management literature has found that Judging individuals prefer to regulate and control, while Perceivers prefer to understand and adapt (Nutt, 1993).

For all static problems, the GA evolved teams with more Thinkers compared to Feelers, but evolved teams with more Feelers compared to Thinkers for all dynamic problems except for Linear Disjunctive (more Thinkers than Feelers) and Linear Additive (equal numbers of Feelers and Thinkers) (Figure 9). This is because Feelers are more influenced by the behaviour of their companions compared to Thinkers. Such personalities have been found in existing research to be more consultative and adaptive to change (Nutt, 1993).

Comparison between Sensors and Intuitives are less clear cut. For all static problems, the GA evolved more Sensors compared to Intuitives, while for dynamic problems, Intuitives are used more frequently than Sensors for all disjunctive tasks and linear and rotary additive tasks (Figure 9). When fitness is determined by using the best solution from any team member, the GA chooses team members that “intuit” the solution space and anticipate where to move. Figure 10 illustrates the solution space as perceived by a team in one run, sampled over time. At $t=50$, team member INTJ (last column row 2) has correctly mapped the solution space, finding four distinct optimal regions corresponding to the locations where the optimal rotates to every 10 timesteps. A study by Allinson et al. (2000) of more than 150 founders of high growth companies found that these founders exhibit higher intuition compared to general population of managers and the intuition has helped them to be quick at identifying and exploiting opportunities. In evolutionary computation literature, gaussian process modelling have been used to improve speed and quality of optimisation (e.g., Büche et al. (2005), Zhang et al. (2010), and Han et al. (2017)). For conjunctive tasks, when fitness of a team depends on the fitness of the worst team member, the GA optimises teams with more Sensors compared to Intuitives.

Finally, in all disjunctive tasks, we found an unusual behaviour by this PSO-based model: instead of team members

converging onto one solution, the team spreads themselves more widely so that members become more likely to catch the moving target as it passes by. This is evident in Figure 11, where all disjunctive tasks have a higher average distance between team members compared to other tasks. In particular, for rotary change, the distance is up to four times more than the other changes, as it is beneficial for team members to be stationed at (0, 50), (-50, 0), (50, 0) and (0, -50).

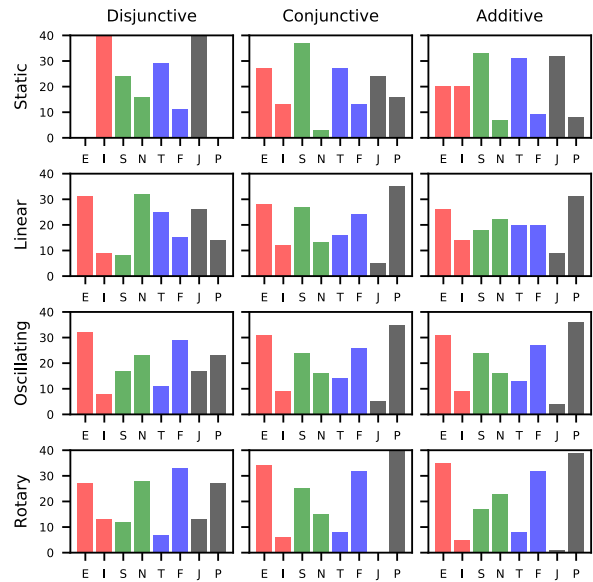


Figure 9: Personality count for task types and dynamic problems in terms of opposing MBTI dichotomy: Extraverts vs. Introverts (E vs. I), Sensors vs. Intuitives (S vs. N), Thinkers vs. Feelers (T vs. F), and Judgers vs. Perceivers (J vs. P). x-axis is the MBTI dichotomies, and y-axis is the personality count.

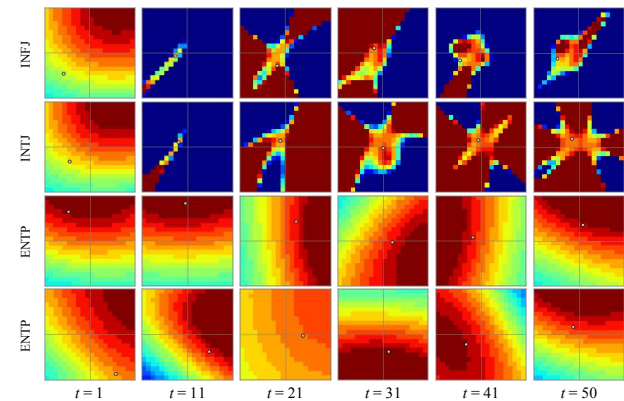


Figure 10: Solution space as perceived by each team member in one of the teams selected by the GA for Rotary Disjunctive, sampled at $t=1, 11, 21, 31, 41, 50$. The circle \circ in each image denotes the agent's position for that timestep.

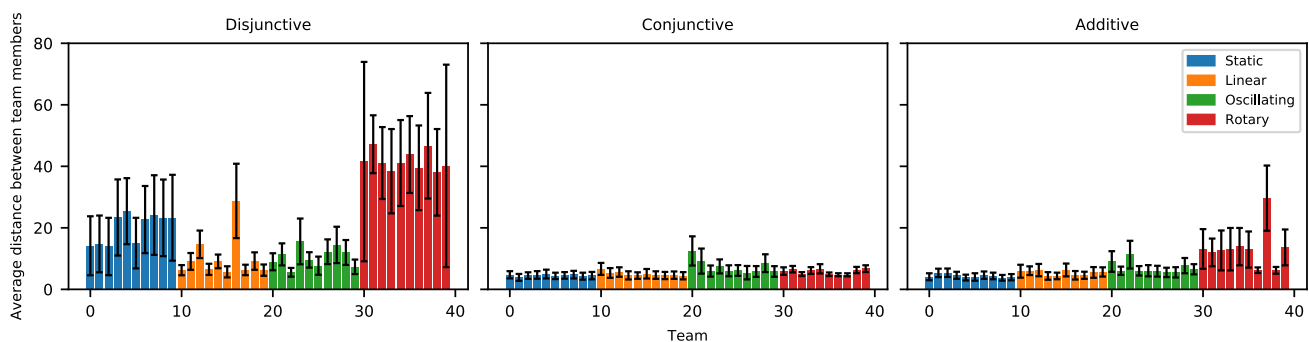


Figure 11: Average distance between team members at $t=50$ for all teams selected by the GA. Error bars represent standard deviation.

Conclusions

Dynamic optimisation is a commonly found class of problem in the real world, and teams of people handle such problems regularly in their working lives. This research used a genetic algorithm to optimise the constituent members of teams as they tackled dynamic problems and task types. The teams were represented by an agent-based model of personality, each corresponding to a different PSO-based behaviour and communication strategy.

We found that different combinations of personalities are selected for different dynamic problems. The GA evolved teams comprising heterogenous personalities, with different combinations of personalities for each type of problem. Introvers, Sensors, Thinkers and Judgers are used frequently by the GA for static problems, while Extraverts, Intuitives, Feelers and Perceivers are used more frequently for dynamic problems. Analysing the movement of the agents over time and their internal representations of the problem, it is clear that Intuitive types learned to “anticipate” repeating solutions, working as a team to spread themselves across the likely good solution areas, rather than behaving in the more typical manner of converging to a single point in the solution space.

References

- Ahrndt, S., Fährndrich, J., & Albayrak, S. (2015). Modelling of personality in agents: from psychology to implementation. *Proceedings of the Workshop on Human-Agent Interaction Design and Models*, pages 1-16.
- Allinson, C. W., Chell, E., & Hayes, J. (2000). Intuition and entrepreneurial behaviour. *European Journal of Work and Organizational Psychology*, 9(1): 31-43.
- Buche, D., Schraudolph, N. N., & Koumoutsakos, P. (2005). Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2): 183-194.
- Durupinar, F., Pelechano, N., Allbeck, J., Gudukbay, U., & Badler, N. I. (2011). How the ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications*, 31(3): 22-31.
- Han, Z.-H., Zhang, Y., Song, C.-X., & Zhang, K.-S. (2017). Weighted gradient-enhanced kriging for high-dimensional surrogate modeling and design optimization. *AIAA Journal*, 55(12): 4330-4346.
- Heppenstall, A. J., Evans, A. J., & Birkin, M. H. (2007). Genetic algorithm optimisation of an agent-based model for simulating a retail market. *Environment and Planning B: Planning and Design*, 34(6): 1051-1070.
- Jin, Y. (2011). Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2): 61-70.
- Jin, Y., Tang, K., Yu, X., Sendhoff, B., & Yao, X. (2013). A framework for finding robust optimal solutions over time. *Memetic Computing*, 5(1): 3-18.
- Jung, C. (1923). *Psychological Types*. Harcourt, Brace, Oxford, England.
- Knoester, D. B., Goldsby, H. J., & McKinley, P. K. (2013). Genetic variation and the evolution of consensus in digital organisms. *IEEE Transactions on Evolutionary Computation*, 17(3): 403-417.
- Lenski, R. E., Ofria, C., Pennock, R. T., & Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936): 139.
- Lim, S. L., & Bentley, P. J. (2018). Coping with Uncertainty: Modelling Personality when Collaborating on Noisy Problems. *Proceedings of the 2018 Conference on Artificial Life*, pages 566-573.
- Lim, S. L., & Bentley, P. J. (2019). Diversity improves teamwork: optimising teams using a genetic algorithm. *Proceedings of the IEEE Congress on Evolutionary Computation*, in press.
- Lim, S. L., & Finkelstein, A. (2011). Anticipating change in requirements engineering. *Relating Software Requirements and Architectures* (pp. 17-34): Springer.
- Myers, I. B. (1962). *The Myers-Briggs Type Indicator: Manual (1962)*. Consulting Psychologists Press, Palo Alto, California.
- Nutt, P. C. (1993). Flexible decision styles and the choices of top executives. *Journal of Management Studies*, 30(5): 695-721.
- Salvit, J., & Sklar, E. (2012). Modulating agent behavior using human personality type. *Proceedings of the Workshop on Human-Agent Interaction Design and Models*, pages 145-160.
- Steiner, I. D. (1972). *Group Processes and Productivity*. Academic Press, New York.
- Teixeira, M. A. P., Bardagi, M. P., Lassance, M. C. P., de Oliveira Magalhães, M., & Duarte, M. E. (2012). Career adapt-abilities scale—Brazilian form: psychometric properties and relationships to personality. *Journal of Vocational Behavior*, 80(3): 680-685.
- Wang, Z.-J., Xu, X.-F., & Zhan, D.-C. (2009). Genetic algorithm for collaboration cost optimization-oriented partner selection in virtual enterprises. *International Journal of Production Research*, 47(4): 859-881.
- Williams, C. K., & Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, pages 514-520.
- Zhang, Q., Liu, W., Tsang, E., & Virginas, B. (2010). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3): 456-474.

Only Two Types of Strategies Enforce Linear Payoff Relationships Under Observation Errors in Repeated Prisoner's Dilemma Games

Azumi Mamiya¹ and Genki Ichinose¹

¹Department of Mathematical and Systems Engineering, Shizuoka University,
3-5-1 Johoku, Naka-ku, Hamamatsu, 432-8561, Japan
ichinose.genki@shizuoka.ac.jp

Abstract

The repeated prisoner's dilemma (RPD) game has revealed how cooperation and competition arise among competitive players in long-run relationships. In the RPD game with no errors, zero-determinant (ZD) strategies allow a player to unilaterally set a linear relationship between the player's own payoff and the opponent's payoff regardless of the strategy of the opponent. On the other hand, unconditional strategies such as ALLD and ALLC also unilaterally set a linear relationship. However, little is known about the existence of such strategies in the RPD game with errors. Here, we analytically search for the strategies that enforce a linear payoff relationship under observation error in the RPD game. As a result, we found that, even in the case with observation errors, the only strategy sets that enforce a linear payoff relationship are either ZD strategies or unconditional strategies.

The two-player repeated prisoner's dilemma (RPD) game is a model for exploring the players' long-run relationships, which mathematically reveals how cooperation and competition arise among competitive players. In the long history of the RPD game, strategies that can unilaterally control opponent's payoff have been unknown. However, in 2012, Press and Dyson found a novel class of strategies which contain such ultimate strategies, called zero-determinant (ZD) strategies (Press and Dyson, 2012). ZD strategies impose a linear relationship between the payoffs for a focal player and his opponent regardless of the strategy that the opponent implements. The discovery of ZD strategies inspired various relevant studies, which promote an understanding of the nature of human cooperation. In contrast, unconditional strategies such as ALLC and ALLD can also unilaterally set a linear payoff relationship to the opponent. A previous study revealed that those two types of strategies are the only sets which enforce a linear payoff relationship (Ichinose and Masuda, 2018).

These two strategies were found in the case of no errors. Errors are unavoidable in biological organisms and it may lead to the collapse of cooperation due to negative effects. Thus, the effect of errors has been considered in the literature of the RPD game. However, except for Hao et al. (2015), the effect of errors has not been considered for strategies that enforce a linear relationship. In this study, we incorporate observation errors within our model. In human interactions, observation errors refer to the misunderstand-

ing of the opponent's action, which often happens in a real society. The purpose of this study is how those errors affect the strategies that enforce a linear payoff relationship. Specifically, we aim to investigate whether those strategies can exist even in the case where observation errors are considered.

Here, we consider the symmetric two-person prisoner's dilemma game. Each player $i \in \{X, Y\}$ chooses an action $a_i \in \{C, D\}$. Each player cannot directly see what action the opponent chose. Instead, they can only observe a signal $\omega_i \in \{g, b\}$, where g and b denote good and bad signals, respectively. The signal cannot be observed by the other player, meaning that the signal is private information. Each player's signal ω_i basically depends on the opponent's action but is also affected by noise from the environment, which is a stochastic variable. In other words, a player observes g (or b) when the other player chooses an action C (or D). However, when an error occurs, a player observes b (or g) although the other player chooses an action C (or D). Let ϵ be the probability that an error happens to one particular player but not to the other and ξ be the probability that an error happens to both players. Then, the probability that an error occurs to neither player is $1 - 2\epsilon - \xi$. The realized payoff for each player depends only on the action he chose and the signal he received. Then the payoff matrix is given by

$$\begin{array}{cc} & \begin{array}{c} g \\ b \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} R & S \\ T & P \end{pmatrix}. \end{array} \quad (1)$$

The entries represent the payoffs that the focal player gains in a single round of a repeated game. Each row and column represents the action that the focal player chose and the signal he observed, respectively. The expected payoffs under different action profiles (C, C), (C, D), (D, C) and (D, D) are denoted as R_E , S_E , T_E and P_E , where $R_E = R(1 - \epsilon - \xi) + S(\epsilon + \xi)$, $S_E = S(1 - \epsilon - \xi) + R(\epsilon + \xi)$, $T_E = T(1 - \epsilon - \xi) + P(\epsilon + \xi)$, $P_E = P(1 - \epsilon - \xi) + T(\epsilon + \xi)$, respectively. We assume that $T_E > R_E > P_E > S_E$ and $2R_E > T_E + S_E$, which dictates the repeated prisoner's dilemma game.

Consider two players X and Y that adopt memory-one strategies, with which they use only the outcomes of the last round to decide the action to be submitted in the current

round. A memory-one strategy is specified by a 4-tuple; X 's strategy is given by a combination of $\mathbf{p} = (p_1, p_2, p_3, p_4)$, where $0 \leq p_1, p_2, p_3, p_4 \leq 1$. The subscripts 1, 2, 3 and 4 of \mathbf{p} mean previous outcome Cg, Cb, Dg and Db , respectively. p_1 is the conditional probability that X cooperates when X cooperated and observed signal g in the last round, p_2 is the conditional probability that X cooperates when X cooperated and observed signal b in the last round, and so forth. Similarly, Y 's strategy is specified by a combination of $\mathbf{q} = (q_1, q_2, q_3, q_4)$, where $0 \leq q_1, q_2, q_3, q_4 \leq 1$. Because both players adopt a memory-one strategy, the stochastic state of the two players in round t is described by $\mathbf{v}(t) = (v_1(t), v_2(t), v_3(t), v_4(t))$, where the subscripts 1, 2, 3 and 4 of \mathbf{v} mean the stochastic state (C,C), (C,D), (D,C) and (D,D), respectively. $v_1(t)$ is the probability that both players cooperate in round t , $v_2(t)$ is the probability that X cooperates and Y defects in round t , and so forth.

The state transition matrix for the game can be described by M . Then, the stochastic state of the two players in round $t + 1$ is calculated by $\mathbf{v}(t + 1) = \mathbf{v}(t)M$. The stationary distribution for M is a vector \mathbf{v} such that $\mathbf{v} = \mathbf{v}M$. The dot product of an arbitrary vector $\mathbf{f} = (f_1, f_2, f_3, f_4)$ with the stationary distribution vector \mathbf{v} , $\mathbf{v} \cdot \mathbf{f} \equiv D(\mathbf{p}, \mathbf{q}, \mathbf{f})$, can be described by the form of determinant as follows:

$$\begin{vmatrix} \tau p_1 q_1 + \epsilon p_1 q_2 + \epsilon p_2 q_1 + \xi p_2 q_2 - 1 & \mu p_1 + \eta p_2 - 1 & \mu q_1 + \eta q_2 - 1 & f_1 \\ \epsilon p_1 q_3 + \xi p_1 q_4 + \tau p_2 q_3 + \epsilon p_2 q_4 & \eta p_1 + \mu p_2 - 1 & \mu q_3 + \eta q_4 & f_2 \\ \epsilon p_3 q_1 + \tau p_3 q_2 + \xi p_4 q_1 + \epsilon p_4 q_2 & \mu p_3 + \eta p_4 & \eta q_1 + \mu q_2 - 1 & f_3 \\ \xi p_3 q_3 + \epsilon p_3 q_4 + \epsilon p_4 q_3 + \tau p_4 q_4 & \eta p_3 + \mu p_4 & \eta q_3 + \mu q_4 & f_4 \end{vmatrix} \quad (2)$$

where $\mu = 1 - \epsilon - \xi$, $\eta = \epsilon + \xi$. Then player X 's expected payoff in the stationary state is represented by

$$s_X = \frac{\mathbf{v} \cdot \mathbf{S}_X}{\mathbf{v} \cdot \mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_X)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}, \quad (3)$$

where $\mathbf{S}_X = (R_E, S_E, T_E, P_E)$ is X 's payoff vector. Similarly, player Y 's expected payoff is represented by

$$s_Y = \frac{\mathbf{v} \cdot \mathbf{S}_Y}{\mathbf{v} \cdot \mathbf{1}} = \frac{D(\mathbf{p}, \mathbf{q}, \mathbf{S}_Y)}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}, \quad (4)$$

where $\mathbf{S}_Y = (R_E, T_E, S_E, P_E)$ is Y 's payoff vector.

We search for player X 's strategies which impose a linear payoff relationship between the two players, i.e.,

$$\alpha s_X + \beta s_Y + \gamma = 0. \quad (5)$$

Here, the linear combination of s_X and s_Y is given by

$$\alpha s_X + \beta s_Y + \gamma = \frac{D(\mathbf{p}, \mathbf{q}, \alpha \mathbf{S}_X + \beta \mathbf{S}_Y + \gamma \mathbf{1})}{D(\mathbf{p}, \mathbf{q}, \mathbf{1})}. \quad (6)$$

If the numerator of the right side of Eq. (6) is zero, that is, $D(\mathbf{p}, \mathbf{q}, \alpha \mathbf{S}_X + \beta \mathbf{S}_Y + \gamma \mathbf{1}) = 0$, the right side of Eq.(6) becomes zero and it holds Eq.(5). We use the following determinant theorem for the analysis: For an $n \times n$ matrix A , the following holds: $\det(A) = 0 \Leftrightarrow$ The columns of the matrix A are dependent vectors. Under this condition, we searched for X 's strategies which impose a linear relationship between the two players' payoffs. The result showed

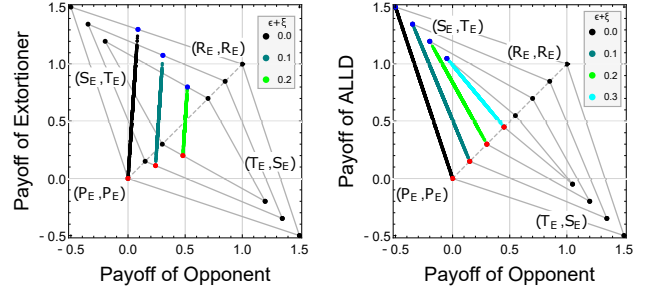


Figure 1: Left: Extortioneer (Player X) vs 1,000 randomly generated strategies (Player Y). Right: ALLD (Player X) vs 1,000 randomly generated strategies (Player Y). In both cases, player X always enforces a linear payoff relationship to player Y . $(T, R, P, S) = (1.5, 1, 0, -0.5)$.

that the only strategies that impose a linear relationship between the two players' payoffs are either

$$\begin{aligned} \mu p_1 + \eta p_2 - 1 &= \alpha R_E + \beta R_E + \gamma \\ \eta p_1 + \mu p_2 - 1 &= \alpha S_E + \beta T_E + \gamma \\ \mu p_3 + \eta p_4 &= \alpha T_E + \beta S_E + \gamma \\ \eta p_3 + \mu p_4 &= \alpha P_E + \beta P_E + \gamma, \end{aligned} \quad (7)$$

or

$$p_1 = p_2 = p_3 = p_4. \quad (8)$$

The former corresponds to the ZD strategies and the latter corresponds to the unconditional strategies, respectively.

Figure 1 shows the numerical realizations of Extortioneer (example of ZD; left panel) and ALLD (example of unconditional strategies; right panel). In each panel, player X (vertical axis) adopts Extortioneer (left) or ALLD (right), respectively. We randomly generated 1,000 strategies for player Y (horizontal axis) for each panel. We numerically confirmed that Extortioneer and ALLD can enforce a linear payoff relationship to player Y even in the case that errors are considered ($\epsilon + \xi = 0.1, 0.2, 0.3$).

In conclusion, we analytically found that, in the RPD game with observation errors, the only strategy sets that enforce a linear payoff relationship are either the ZD strategies or the unconditional strategies, which are consistent with the case of no errors. This result suggests that, in any case, those two sets are only types of strategies that enforce a linear payoff relationship between two players.

References

- Hao, D., Rong, Z., and Zhou, T. (2015). Extortion under uncertainty: Zero-determinant strategies in noisy games. *Phys. Rev. E*, 91:052803.
- Ichinose, G. and Masuda, N. (2018). Zero-determinant strategies in finitely repeated games. *J. Theor. Biol.*, 438:61–77.
- Press, W. H. and Dyson, F. J. (2012). Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent. *Proc. Natl. Acad. Sci. USA*, 109:10409–10413.

Emergence of Coordination with Asymmetric Benefits via Prior Commitment

Ogbo Ndidi Bianca¹, The Anh Han¹

¹School of Computing and Digital Technologies, Teesside University
Emails: ogbondidibianca@gmail.com, t.han@tees.ac.uk

Abstract

When starting a new collective venture, it is important to understand partners' motives and how strongly they commit to common goals. Arranging prior commitments or agreements on how to behave has been shown to be an evolutionary viable strategy in the context of cooperation dilemmas, ensuring high levels of mutual cooperation among self-interested individuals. However, in many situations, commitments can be used to achieve other types of collective behaviours such as coordination. Coordination is arguably more complex to achieve since there might be multiple desirable collective outcomes in a coordination problem (compared to mutual cooperation, the only desirable outcome in cooperation dilemmas), especially when these outcomes entail asymmetric benefits or payoffs for those involved. Using methods from Evolutionary Game Theory (EGT), herein we study how prior commitments can be adopted as a tool for enhancing coordination when its outcomes exhibit an asymmetric payoff structure. Our results, both by numerical simulations and analytically, show that whether prior commitment would be a viable evolutionary mechanism for enhancing coordination strongly depends on the collective benefit of coordination, and more importantly, how asymmetric benefits are resolved in a commitment deal.

Introduction

Achieving collective behaviours among individuals with their own personal interest is an important social challenge in various societies (Ostrom, 1990; Pitt et al., 2012). From coordinating individuals in the workplace to maintaining cooperative relationship between nations, it is often jeopardised by individual self-interest. The study of mechanisms that support the evolution of such collective behaviours has been of great interest in many disciplines, ranging from Evolutionary Biology, Economics, Physics and Computer Science (Nowak, 2006; Sigmund, 2010; West et al., 2007; Han, 2013). Several mechanisms that can promote the emergence and stability of collective behaviours among such individuals, have been proposed. They include kin and group selection, direct and indirect reciprocities, spatial networks, reward and punishment (Nowak, 2006).

Recently, the capacity to create, and commit to, prior agreements (Nesse, 2001; Frank, 1988; Han et al., 2017,

2015a) has been proposed as an evolutionarily viable strategy inducing cooperative behaviour in the context of cooperation dilemmas; namely, the Prisoner's Dilemma (PD) (Han et al., 2013) and the Public Goods Game (PGG) (Han et al., 2017). It provides an alternative to different forms of punishment against inappropriate behaviour and of rewards to stimulate the appropriate one (Martinez-Vaquero et al., 2015; Sasaki et al., 2015; Powers et al., 2012). These works have solely focused on the roles of commitments for enhancing mutual cooperation among self-interested individuals. However, commitments can be adopted as a tool for enhancing other types of collective behaviour such as coordination (Nesse, 2001; Ostrom, 1990; Barrett, 2016). In the context of cooperation dilemma games such as the PD and PGG (Nowak, 2006), mutual cooperation is the only desirable collective outcome to which all parties are required to commit if an agreement is to be arranged. In contrast, in a coordination problem, there might be multiple optimal or desirable collective outcomes and players might have distinct, incompatible preferences regarding which outcome a mutual agreement should aim to achieve (e.g. due to asymmetric payoffs). Consider for instance a situation where two investment firms competing within a same product market who need to make strategic decision on which technology to adopt (Zhu and Weyant, 2003), a low-benefit (L) or a high-benefit (H) technology. Individually, adopting H would lead to a larger benefit. However, if both firms invest on H they would end up competing with each other leading to a smaller accumulated benefit than if they could coordinate with each other to choose different technologies. However, given the asymmetry in the benefits in such an outcome, clearly no firm would want to commit to the outcome where its option is L, unless some form of compensation from the one selecting H can be ensured.

In this paper, we explore how arranging a prior agreement or commitment can be used as a mechanism for enhancing coordination and social welfare in this type of coordination problems. Before individuals embark on a joint venture, a pre-agreement makes the motives and intentions of all parties involved more transparent, thereby enabling an easier

coordination of personal interests (Nesse, 2001; Cohen and Levesque, 1990; Han et al., 2015b). Although our approach is applicable for a wide range of coordination problems (e.g. single market product investments as described above), we will use technology investment strategic decision making as a case study, allowing us to describe the problem clearly. Namely, we will describe a technology adoption game capturing the competitive market and decision-making process among firms adopting new technologies (Zhu and Weyant, 2003), with a key parameter α representing how competitive the market is (thus describing how important coordination is). Similar to previous commitment models, we will rely on theoretical analysis and numerical simulations based on dynamical methods from Evolutionary Game Theory (EGT) (Hofbauer and Sigmund, 1998).

Our results show that whether pre-commitment would be a viable mechanism for enhancing coordination and social welfare strongly depends on α . Moreover, we demonstrate that agreements for coordination (with asymmetric benefits) exhibit more complex decision points than in previous models on cooperation dilemmas, leading to a larger behavioural space and a larger set of strategies. For instance, the firms will need to decide which desirable coordination outcome they want to propose to others. Moreover, since for such a coordination outcome, players receive different benefits, a commitment proposer needs to decide how much compensation for the one who agrees to commit to select L is required, depending on the cost of arranging commitments as well as the costs and benefits of adopting available technologies. We show that the outcome of evolutionary dynamics significantly depends on the compensation amounts.

The next section discusses related work, which is followed by a description of our models and details of the EGT methods for analysing them. Results of the analysis and a final discussion will then follow.

Related Work

The problem of explaining the emergence and stability of collective behaviours has been actively addressed in different disciplines (Nowak, 2006; Sigmund, 2010). Among other mechanisms, such as reciprocity and costly punishment, closely related to our present model is the study of cooperative behaviours and pre-commitment in cooperation dilemmas, for both two-player and multiplayer games (Han et al., 2013, 2017; Sasaki et al., 2015). It has shown that to enhance cooperation commitments need to be sufficiently enforced and the cost of setting up the commitments is justified with respect to the benefit derived from the interactions—both by means of theoretical analysis and of behavioural experiments (Ostrom, 1990; Cherry and McEvoy, 2013). Our results show that this same observation is seen for coordination problems. However, arranging commitments for enhancing coordination is more complex, exhibiting a larger behavioural space, and furthermore,

their outcomes strongly depend on new factors only appearing in coordination problems; namely, a successful commitment deal needs to take into account the fact that multiple desirable collective outcomes exist for which players have incompatible preferences; and thus how benefits can be shared through compensations in order to resolve the issues of asymmetric benefits, is crucially important.

There have been several works studying the evolution of coordination, using the so-called Stag Hunt game, see e.g. (Skyrms, 2003; Pacheco et al., 2009; Santos et al., 2006; Sigmund, 2010). However, to the best of our knowledge there has been no work studying how prior commitments can be modelled and used for enhancing the outcome of the evolution of coordination. As our results below show, significant enhancement of coordination and population welfare can be achieved via the arrangement of suitable commitment deals.

Furthermore, it is noteworthy that commitments have been studied extensively in Artificial Intelligence and Multi-agent systems literature, see e.g. (Castelfranchi and Falcone, 2010; Chopra and Singh, 2009; Rzdca et al., 2015). Differently from our work, these studies utilise commitments for the purpose of regulating individual and collective behaviours, formalising different aspects of commitments (such as norms and conventions) in multi-agent systems. However, our results and approach provide important new insights into the design of such systems as these require commitments to ensure high levels of efficient collaboration and coordination within a group or team of agents. For example, by providing suitable agreement deals agents can improve the chance that a desirable collective outcome (which is best for the systems as a whole) is reached even when benefits provided by the outcome are different for the parties involved.

Models and Methods

We first describe a two-player technology adoption game then extend it with the option of arranging prior commitments before the game. We then describe our method based on Evolutionary Game Theory for finite populations, for analysing the games.

Technology Adoption (TD) game

We consider the scenario that two firms (players) compete for the same product market, and they need to make a (strategic) decision on which technology to invest on, a low-benefit (L) or a high-benefit (H) technology. The outcome of the interaction can be described in terms of costs and benefits of investments by the following payoff matrix (for row player):

$$\begin{array}{c} \begin{array}{cc} H & L \end{array} \\ \begin{array}{cc} H & L \\ \left(\begin{array}{cc} \alpha b_H - c_H & b_H - c_H \\ b_L - c_L & \alpha b_L - c_L \end{array} \right) = \frac{H}{L} \left(\begin{array}{cc} a & b \\ c & d \end{array} \right), \end{array} \end{array} \quad (1)$$

where c_L , c_H and b_L , b_H ($b_L \leq b_H$) represent the costs and benefits of investing on L and H, respectively; $\alpha \in (0, 1)$

indicates the competitive level of the product market: the firms receive a partial benefit if they both choose to invest on the same technology. Collectively, the smaller α is (i.e. the higher the market competitiveness), the more important that the firms coordinate to choose different technologies. The entries of the payoff matrix are denoted by a, b, c, d , as above. We have $b > a$ and $c > d$. Without loss of generality, we assume that H would generate a greater net benefit, i.e. $c = b_L - c_L < b_H - c_H = b$.

Note that although we describe our model in terms of technology adoption decision making, it is generally applicable to many other coordination problems for instance wherever there are strategic investment decisions to make (in competitive markets of any products).

Commitments

We now describe our extended model where players can arrange a prior commitment before a TD interaction. A commitment proposal is to ask the co-player to adopt a different technology. That is, a strategist intending to use H (resp., L) would ask the co-player to adopt L (resp., H). We denote these commitment proposing strategies as HP and LP, respectively. Similarly to previous models of commitments (for PD and PGG) (Han et al., 2013, 2015a), to make the commitment deal reliable, a proposer pays an arrangement cost ϵ . If the co-player agrees with the deal, then the proposer assumes that the opponent will adopt the agreed choice, yet there is no guarantee that this will actually be the case. Thus whenever a co-player refuses to commit, HP and LP would play H in the game. When the co-player accepts the commitment though later does not honour it, she has to compensate the honouring co-player at a personal cost δ .

Differently from previous models on PD and PGG where an agreed outcome leads to the same payoff for all parties in the agreement (mutual cooperation benefit), in the current model such an outcome would lead to different payoffs for those involved. Therefore, as part of the agreement, HP would compensate after the game an amount θ_1 to accepted player that honours the agreement; while LP would request a compensation θ_2 from such an accepted co-player.

Besides HP and LP, we consider a minimal model with the following (basic) strategies in this commitment version:

- Non-proposing acceptors, HC and LC, who always commit when being proposed a commitment deal wherein they are willing to adopt any technology proposed (even when it is different from their intended choice), honour the adopted agreement, but do not propose a commitment themselves. They play their intended choice, i.e. H and L, respectively, when there is no agreement in place;
- Non-acceptors, HN and LN, who do not accept commitment, play their intended choice during the game, and do not propose commitments;

- Fake committers, HF and LF, who accept a commitment proposal yet play the choice opposite to what has been agreed whenever the game takes place. These players assume that they can exploit the commitment proposing players without suffering the consequences;

Note that similar to the commitment models for the PD game (Han et al., 2013), some possible strategies have been excluded from the analysis since they are dominated by at least one of the strategies in any configuration of the game: they can be omitted without changing the outcome of the analysis. For example, those who propose a commitment (i.e. paying a cost ϵ) but then do not honour (thus have to pay the compensation when facing a honouring acceptors) would be dominated by the corresponding non-proposers.

Together the model consists of eight strategies that define the following payoff matrix, capturing the average payoffs that each strategy will receive upon interaction with one of the other seven strategies (where we denote $\lambda = \theta_1 + \theta_2$, $\lambda_1 = b - \epsilon - \theta_1$, $\lambda_2 = c - \epsilon + \theta_2$, $\lambda_3 = a - \epsilon + \delta$ and $\lambda_4 = d - \epsilon + \delta$, just for the sake of clear representation)

$$\begin{array}{c} \begin{array}{cccccccc} & \text{HP} & \text{LP} & \text{HN} & \text{LN} & \text{HC} & \text{LC} & \text{HF} & \text{LF} \\ \text{HP} & \left(\frac{b+c-\epsilon}{2} \right. & \left. \frac{2b-\epsilon-\lambda}{2} \right. & a & b & \lambda_1 & \lambda_1 & \lambda_3 & \lambda_3 \\ \text{LP} & \left. \frac{2c-\epsilon+\lambda}{2} \right. & \left. \frac{b+c-\epsilon}{2} \right. & a & b & \lambda_2 & \lambda_2 & \lambda_4 & \lambda_4 \\ \text{HN} & a & a & a & b & a & b & a & b \\ \text{LN} & c & c & c & d & c & d & c & d \\ \text{HC} & c + \theta_1 & b - \theta_2 & a & b & a & b & a & b \\ \text{LC} & c + \theta_1 & b - \theta_2 & c & d & c & d & c & d \\ \text{HF} & a - \delta & d - \delta & a & b & a & b & a & b \\ \text{LF} & a - \delta & d - \delta & c & d & c & d & c & d \end{array} \end{array} \quad (2)$$

Note that when two commitment proposers interact only one of them will need to pay the cost of setting up the commitment. Yet, as either one of them can take this action they pay this cost only half of the time (on average). In addition, the average payoff of HP when interacting with LP is given by $\frac{1}{2}(b - \epsilon - \theta_1 + b - \theta_2) = \frac{1}{2}(2b - \epsilon - \theta_1 - \theta_2)$. When two HP players interact, each receives $\frac{1}{2}(b - \epsilon - \theta_1 + c + \theta_1) = \frac{1}{2}(b + c - \epsilon)$.

We say that *an agreement is fair* if both parties obtain the same benefit when they honour it (after having taken into account the cost of setting up the agreement). For that, we can show that θ_1 and θ_2 must satisfy $\theta_1 = \frac{b-c-\epsilon}{2}$ and $\theta_2 = \frac{b-c+\epsilon}{2}$ ¹, and thus, both parties obtain $\frac{b+c-\epsilon}{2}$. With these conditions it also ensures that the payoffs of HP and LP when interacting with each other are equal.

Our analysis below will first focus on whether and when the fair agreements can lead to improvement in terms of coordination and the overall social welfare (i.e. average population payoff). We will then study how different kinds of agreements (varying θ_1 and θ_2) affect the outcome.

¹These can be obtained for instance by comparing the payoffs of HP and HC when they interact, i.e. $b - c - \theta_1 = c + \theta_1$. Solving this equation we would obtain $\theta_1 = \frac{b-c-\epsilon}{2}$. Similarly for θ_2 .

Evolutionary Dynamics

In this work, we will perform theoretical analysis and numerical simulations (see next section) using EGT methods for finite populations (Nowak et al., 2004; Imhof et al., 2005). In such a setting, individuals' payoff represents their *fitness* or social *success*, and evolutionary dynamics is shaped by social learning (Hofbauer and Sigmund, 1998; Sigmund, 2010), whereby the most successful individuals will tend to be imitated more often by the other individuals. In the current work, social learning is modelled using the so-called pairwise comparison rule (Traulsen et al., 2006), a standard approach in EGT, assuming that an individual A with fitness f_A adopts the strategy of another individual B with fitness f_B with probability p given by the Fermi function, $p_{A,B} = (1 + e^{-\beta(f_B - f_A)})^{-1}$. The parameter β represents the 'imitation strength' or 'intensity of selection', i.e., how strongly the individuals base their decision to imitate on fitness difference between themselves and the opponents. For $\beta = 0$, we obtain the limit of neutral drift – the imitation decision is random. For large β , imitation becomes increasingly deterministic.

In the absence of mutations or exploration, the end states of evolution are inevitably monomorphic: once such a state is reached, it cannot be escaped through imitation. We thus further assume that, with a certain mutation probability, an individual switches randomly to a different strategy without imitating another individual. In the limit of small mutation rates, the dynamics will proceed with, at most, two strategies in the population, such that the behavioural dynamics can be conveniently described by a Markov Chain, where each state represents a monomorphic population, whereas the transition probabilities are given by the fixation probability of a single mutant (Imhof et al., 2005; Nowak et al., 2004). The resulting Markov Chain has a stationary distribution, which characterises the average time the population spends in each of these monomorphic end states.

Let N be the size of the population. Denote $\pi_{X,Y}$ the payoff a strategist X obtains in a pairwise interaction with strategist Y (defined in the payoff matrices). Suppose there are at most two strategies in the population, say, k individuals using strategy A ($0 \leq k \leq N$) and $(N - k)$ individuals using strategies B. Thus, the (average) payoff of the individual that uses A and B can be written as follows, respectively,

$$\begin{aligned}\Pi_A(k) &= \frac{(k-1)\pi_{A,A} + (N-k)\pi_{A,B}}{N-1}, \\ \Pi_B(k) &= \frac{k\pi_{B,A} + (N-k-1)\pi_{B,B}}{N-1}.\end{aligned}\quad (3)$$

Now, the probability to change the number k of individuals using strategy A by \pm one in each time step can be written as (Traulsen et al., 2006)

$$T^\pm(k) = \frac{N-k}{N} \frac{k}{N} \left[1 + e^{\mp\beta[\Pi_A(k) - \Pi_B(k)]} \right]^{-1}. \quad (4)$$

The fixation probability of a single mutant with a strategy A in a population of $(N - 1)$ individuals using B is given by (Traulsen et al., 2006; Nowak et al., 2004)

$$\rho_{B,A} = \left(1 + \sum_{i=1}^{N-1} \prod_{j=1}^i \frac{T^-(j)}{T^+(j)} \right)^{-1}. \quad (5)$$

Considering a set $\{1, \dots, q\}$ of different strategies, these fixation probabilities determine a transition matrix $M = \{T_{ij}\}_{i,j=1}^q$, with $T_{ij,j \neq i} = \rho_{ji}/(q-1)$ and $T_{ii} = 1 - \sum_{j=1, j \neq i}^q T_{ij}$, of a Markov Chain. The normalised eigenvector associated with the eigenvalue 1 of the transposed of M provides the stationary distribution described above (Imhof et al., 2005), describing the relative time the population spends adopting each of the strategies.

Risk-dominance An important measure to compare the two strategies A and B is which direction the transition is stronger or more probable, an A mutant fixating in a population of individuals using B, $\rho_{B,A}$, or a B mutant fixating in the population of individuals using A, $\rho_{A,B}$. It can be shown that the former is stronger, in the limit of large N , if (Nowak et al., 2004; Sigmund, 2010)

$$\pi_{A,A} + \pi_{A,B} > \pi_{B,A} + \pi_{B,B}. \quad (6)$$

Results

Conditions for the viability of commitments

First of all, using pair-wise analysis (using Equation 6) it can be shown that if $\theta_1 + \theta_2 < b - c$ then HP is preferred (i.e. risk-dominant, see Methods) to LP. Otherwise, LP is risk-dominant against HP.

We now derive the conditions regarding the commitment parameters for which HP and LP are viable strategies, i.e. when they are risk-dominant against all other non-proposing strategies. Namely, using Equation 6 we can derive that HP and LP are risk-dominant against all other six non-proposing strategies, respectively, if and only if

$$\begin{aligned}\epsilon &< \min\left\{b + c - 2a, 3b - c - 2d, \frac{3b - c - 2a - 4\theta_1}{3}, \right. \\ &\quad \left. \frac{3b - c - 2d - 4\theta_1}{3}, \frac{b + c - 2a + 4\delta}{3}, \frac{b + c - 2d + 4\delta}{3}\right\}, \\ \epsilon &< \min\left\{b + c - 2a, 3b - c - 2d, \frac{3c - b - 2a + 4\theta_2}{3}, \right. \\ &\quad \left. \frac{3c - b - 2d + 4\theta_2}{3}, \frac{b + c - 2a + 4\delta}{3}, \frac{b + c - 2d + 4\delta}{3}\right\}.\end{aligned}\quad (7)$$

Note that each element in the *min* expressions above corresponds to the condition for one of the six non-proposing strategies HN, LN, HC, LC, HF, LF, respectively.

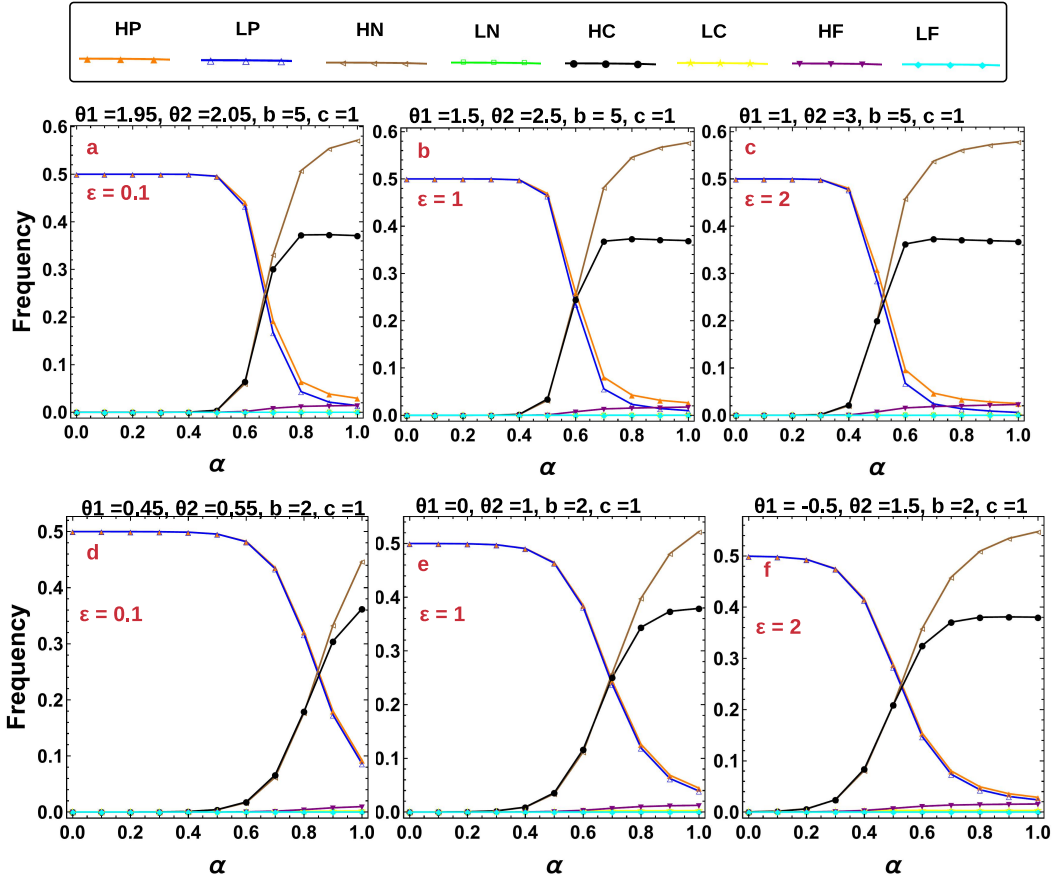


Figure 1: **Frequency of the eight strategies, HP, LP, HN, LN, HC, LC, HF and LF, as a function of α , for different values of ϵ and game configurations.** In general, the commitment proposing strategies HP and LP dominate the population when α is small while HN and HC dominate when α is sufficiently large. The thresholds of α for which HP and LP dominate, in all cases, are in accordance with the analytical condition described in Equation 11. Parameters: in all panels $c_H = 1, c_L = 1, b_L = 2$ (i.e. $c = 1$); Top row) $b_H = 6$ (i.e. $b = 5$) and bottom row) $b_H = 3$ (i.e. $b = 2$); Other parameters: $\delta = 6; \beta = 0.1$; population size $N = 100$; Fair agreements are used, where θ_1 and θ_2 are given by $\theta_1 = (b - c - \epsilon)/2$ and $\theta_2 = (b - c + \epsilon)/2$.

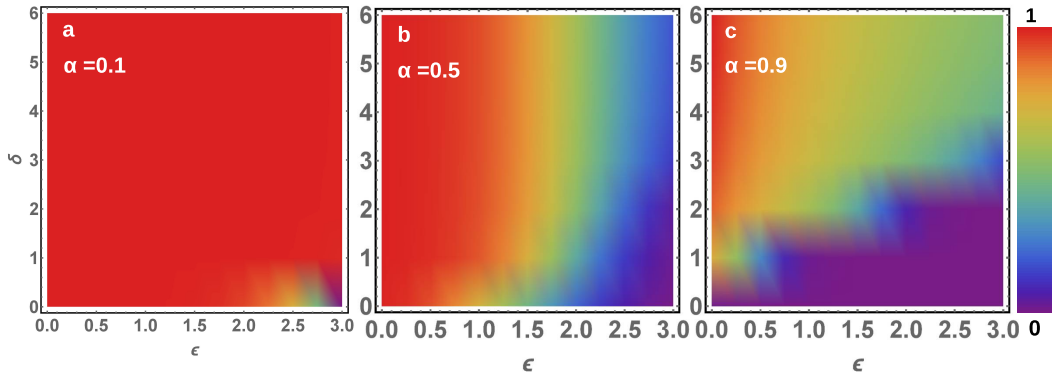


Figure 2: **Total frequency of commitment strategies (i.e. sum of the frequencies of HP and LP), as a function of ϵ and δ , for different values of α .** In general, the commitment proposing strategies dominate the population whenever ϵ is sufficiently small and δ is sufficiently large. Furthermore, the smaller α , these commitment strategies dominate for a wider range of ϵ and δ , especially when α is smaller. Parameters: in all panels $c_H = 1, c_L = 1, b_L = 2$ (i.e. $c = 1$), and $b_H = 6$ (i.e. $b = 5$). Other parameters: $\beta = 0.1$; population size $N = 100$; Fair agreements are used, where θ_1 and θ_2 are given by $\theta_1 = (b - c - \epsilon)/2$ and $\theta_2 = (b - c + \epsilon)/2$.

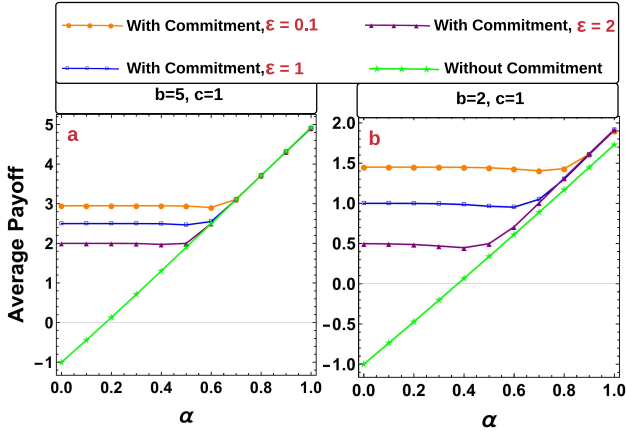


Figure 3: Average population payoff as a function of α , when commitment is absent and when it is present, for different values of ϵ . In general, we observe that when α is small, significant improvement in terms of the average population payoff can be achieved through prior commitments, while when α is sufficiently large, little improvement can be achieved, especially when b_H/b_L is small (compare panels a and b). Parameters: in all panels $c_H = 1$, $c_L = 1$, $b_L = 2$ (i.e. $c = 1$); in panel a) $b_H = 6$ (i.e. $b = 5$) and in panel b) $b_H = 3$ (i.e. $b = 2$); Other parameters: $\delta = 6$; $\beta = 0.1$; population size $N = 100$; Fair agreements are used, where θ_1 and θ_2 are given by $\theta_1 = (b - c - \epsilon)/2$ and $\theta_2 = (b - c + \epsilon)/2$.

Thus, we can derive the conditions for θ_1 , θ_2 and δ :

$$\begin{aligned} \theta_1 &< \frac{1}{4}(3b - c - 3\epsilon - 2 \max\{a, d\}), \\ \theta_2 &> \frac{1}{4}(b - 3c + 3\epsilon + 2 \max\{a, d\}), \\ \delta &> \frac{1}{4}(3\epsilon - b - c + 2 \max\{a, d\}). \end{aligned} \quad (8)$$

In particular, for fair agreements, i.e. $\theta_1 = (b - c - \epsilon)/2$ and $\theta_2 = (b - c + \epsilon)/2$, we obtain

$$\begin{aligned} \epsilon &< \min\{3b - c - 2d, b + c - 2 \max\{a, d\}\}, \\ \delta &> \frac{1}{4}(3\epsilon - b - c + 2 \max\{a, d\}). \end{aligned} \quad (9)$$

Since the first inequality can be simplified further, we obtain

$$\begin{aligned} \epsilon &< b + c - 2 \max\{a, d\}, \\ \delta &> \frac{1}{4}(3\epsilon - b - c + 2 \max\{a, d\}). \end{aligned} \quad (10)$$

(since $3b - c - 2d > b + c - 2 \max\{a, d\}$, which is due to $b > c$ and $\max\{a, d\} \geq d$)

In general, these conditions indicate that for commitments to be a viable option for improving coordination, the cost of arrangement ϵ must be sufficiently small while the compensation associated with the contract needs to be sufficiently large (see already Figure 2 for numerical validation). Furthermore, for the first condition to hold, it is necessary that $b + c > 2 \max\{a, d\}$. It means that the total payoff of two

players when playing the TD game is always greater when they can coordinate to choose different technologies, than when they both choose the same technology.

Moreover, the conditions in Equation 10 can be expressed in terms of α and the costs and benefits of investment, as follows (see again the payoff matrices in Equation 1)

$$\begin{aligned} \alpha &< 1 + \min\left\{\frac{c_H + b_L - c_L - \epsilon}{2b_H}, \frac{c_L + b_H - c_H - \epsilon}{2b_L}\right\}, \\ \alpha &< 1 + \min\left\{\frac{c_H + b_L - c_L - 3\epsilon + 4\delta}{2b_H}, \frac{c_L + b_H - c_H - 3\epsilon + 4\delta}{2b_L}\right\}, \end{aligned}$$

which can be rewritten as

$$\alpha < 1 + \min\left\{\frac{c_H + b_L - c_L - \gamma}{2b_H}, \frac{c_L + b_H - c_H - \gamma}{2b_L}\right\}, \quad (11)$$

where $\gamma = \max\{\epsilon, 3\epsilon - 4\delta\}$.

This condition indicates under what condition of the market competitiveness and the costs and benefits of investing in available technologies, commitments can be an evolutionarily viable mechanism. Intuitively, for given costs and benefits of investment (i.e. fixing c_L , c_H , b_L , b_H), a larger cost of arranging a (reliable) agreement, ϵ , leads to a smaller threshold of α where commitment is viable. Moreover, given a commitment system (i.e. fixing ϵ and δ), assuming similar costs of investment for the two technologies, then a larger ratio of the benefits obtained from the two technologies, b_H/b_L , leads to a smaller upper bound for α for which commitment is viable.

Remarkably, our numerical analysis below (see already Figure 1) shows that the condition in Equation 11 accurately predicts the threshold of α where commitment proposing strategies (i.e. HP and LP) are highly abundant in the population, leading to improvement in terms of the average population payoff compared to when commitment is absent (Figure 3).

Numerical Results

We calculate the stationary distribution in a population of eight strategies, HP, LP, HN, LN, HC, LC, HF and LF, using methods described above. In Figure 1, we show the frequency of these strategies as a function of α , for different values of ϵ and game configurations. In general, the commitment proposing strategies HP and LP dominate the population when α is small while HN and HC dominate when α is sufficiently large. That is, commitment proposing strategies are viable and successful whenever the market competitiveness is high, leading to the need of efficient coordination among the competing players/firms to ensure high benefits. Notably, we observe that the thresholds of α below which HP and LP are dominant, closely corroborate the analytical condition described in Equation 11, in all cases. Namely, for the parameter values in the first and second rows of Figure 1, $\alpha \approx 0.66$, 0.58 , 0.5 and $\alpha \approx 0.81$, 0.67 , 0.5 , for $\alpha = 0.1, 1, 2$, respectively.

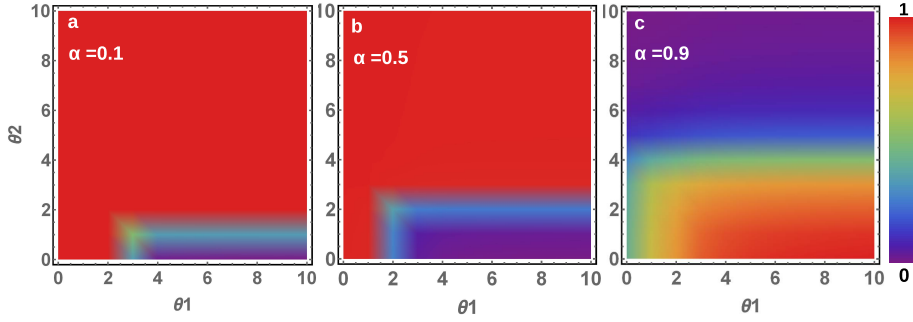


Figure 4: **Average population payoff as a function of θ_1 and θ_2** , for different values of α . When α is small (panels a and b), the highest average payoff is achieved when θ_1 is sufficiently small and θ_2 is sufficiently large, while for large α (panel c), it is the case when θ_1 is sufficiently large and θ_2 is sufficiently small. Parameters: in all panels $c_H = 1$, $c_L = 1$, $b_L = 2$ (i.e. $c = 1$), and $b_H = 6$ (i.e. $b = 5$). Other parameters: $\delta = 4$, $\epsilon = 1$; $\beta = 0.1$; population size $N = 100$.

This observation is robust for varying commitment parameters, i.e. the cost of arranging commitment, ϵ , and the compensation cost associated with commitment, δ , see Figure 2. Namely, we show the total frequency of commitment strategies (i.e. sum of the frequencies of HP and LP) for varying these parameters and for different values of α . It can be seen that, in general, the commitment strategies dominate the population whenever ϵ is sufficiently small and δ is sufficiently large. This observation is in accordance with previous commitment modelling works for the cooperation dilemma games (Han et al., 2013, 2015a, 2017). Furthermore, we observe that in the current coordination problem, that the smaller α is, these commitment strategies dominate the population for wider range of ϵ and δ . Our additional results show that these observations are robust with respect to other game configurations.

Now, in order to determine whether and when commitments can actually lead to meaningful improvement, in Figure 3, we compare the average population payoff or social welfare when a commitment is present and when it is absent. In general, it can be seen that when α is sufficiently small (below a threshold), the smaller it is, the greater improvement of social welfare is achieved through the presence of a commitment deal. Moreover, the smaller the cost of arranging commitments, ϵ , the greater improvement is obtained. On the other hand, when α is sufficiently large, little improvement can be achieved, especially when b_H/b_L is large (which is in accordance with the analytical results above). We can observe that the thresholds for which a notable improvement can be achieved is the same as the one for the viability of HP and LP (i.e. as described in Equation 11).

We now consider what would happen if HP and LP can customise the commitment deal they want to propose, i.e. any θ_1 and θ_2 can be proposed (instead of always being fair). Namely, Figure 4 shows the average population payoff varying these parameters, for different values of α . We observe that when α is small, the highest average payoff is achieved when θ_1 is sufficiently small and θ_2 is sufficiently large, while for large α , it is reverse for the two parameters. That

is, in a highly competitive market (i.e. small α), commitment proposers should be strict (HP keeps sufficient benefit while LP requests sufficient payment, from their commitment partners), while when the market is less competitive (i.e. large α), commitment proposers should be more generous (HP proposes to give a larger benefit while LP requests a smaller payment, from their commitment partners). Our additional results confirm that this observation is robust for different values of ϵ , δ and game configurations.

Conclusions and Future Work

The present paper describes a novel model showing how prior commitments can be adopted as a tool for enhancing coordination when desirable coordination outcomes exhibit an asymmetric payoff structure. For that, we described a technology adoption game where technology investment firms would achieve the best collective outcome if they can coordinate with each other to adopt different technologies, with a parameter α capturing the competitiveness level of the product market and how beneficial it is to achieve coordination. In such a context, there are multiple desirable outcomes and players have distinct preferences in terms of which outcome should be agreed upon, thus leading to a larger behavioural space than in the context of cooperation dilemmas (Han et al., 2013, 2017). We have shown that whether commitment is a viable mechanism for promoting the evolution of coordination, strongly depends on α : when α is sufficiently small, prior commitment is highly abundant leading to significant improvement in terms of social welfare, compared to when commitment is absent. Importantly, we have derived the analytical condition for the threshold of α below which the success of commitments is guaranteed. Furthermore, whenever commitment proposers are allowed to freely choose which deal to propose to their co-players, our results show that, in a highly competitive market (i.e. small α), commitment proposers should be strict (i.e. sharing less benefits), while when the market is less competitive, commitment proposers should be more generous.

In short, our analysis has demonstrated that commitment

is a viable tool for promoting the evolution of diverse collective behaviours among self-interested individuals, beyond the context of cooperation dilemmas where there is only one desirable collective outcome (Nesse, 2001). In future work, we will consider how commitments can solve more complex collective problems, e.g. in a technological innovation race (Han et al., 2019), where there might be a large number of desirable outcomes or equilibriums, especially when the number of players in an interaction increases (Duong and Han, 2015). Also, we aim to compare data on technology adoption from developed and undeveloped countries with our model predictions.

Acknowledgements

The Anh Han is supported by Future of Life Institute (grant RFP2-154).

References

- Barrett, S. (2016). Coordination vs. voluntarism and enforcement in sustaining international environmental cooperation. *Proceedings of the National Academy of Sciences*, 113(51):14515–14522.
- Castelfranchi, C. and Falcone, R. (2010). *Trust Theory: A Socio-Cognitive and Computational Model (Wiley Series in Agent Technology)*. Wiley.
- Cherry, T. L. and McEvoy, D. M. (2013). Enforcing compliance with environmental agreements in the absence of strong institutions: An experimental analysis. *Environmental and Resource Economics*, 54(1):63–77.
- Chopra, A. K. and Singh, M. P. (2009). Multiagent commitment alignment. In *AAMAS’2009*, pages 937–944.
- Cohen, P. R. and Levesque, H. J. (1990). Intention is Choice with Commitment. *Artificial Intelligence*, 42(2-3):213–261.
- Duong, M. H. and Han, T. A. (2015). On the expected number of equilibria in a multi-player multi-strategy evolutionary game. *Dynamic Games and Applications*, pages 1–23.
- Frank, R. H. (1988). *Passions Within Reason: The Strategic Role of the Emotions*. Norton and Company.
- Han, T. A. (2013). *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation: From Artificial Intelligence Techniques to Evolutionary Game Theory Models*, volume 9. Springer SAPERE series.
- Han, T. A., Pereira, L. M., and Lenaerts, T. (2015a). Avoiding or Restricting Defectors in Public Goods Games? *J. Royal Soc Interface*, 12(103):20141203.
- Han, T. A., Pereira, L. M., and Lenaerts, T. (2017). Evolution of commitment and level of participation in public goods games. *Autonomous Agents and Multi-Agent Systems*, 31(3):561–583.
- Han, T. A., Pereira, L. M., and Lenaerts, T. (2019). Modelling and Influencing the AI Bidding War: A Research Agenda. In *AAAI/ACM conference AI, Ethics and Society*.
- Han, T. A., Pereira, L. M., Santos, F. C., and Lenaerts, T. (2013). Good agreements make good friends. *Scientific reports*, 3(2695).
- Han, T. A., Santos, F. C., Lenaerts, T., and Pereira, L. M. (2015b). Synergy between intention recognition and commitments in cooperation dilemmas. *Scientific reports*, 5(9312).
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.
- Imhof, L. A., Fudenberg, D., and Nowak, M. A. (2005). Evolutionary cycles of cooperation and defection. *Proc. Natl. Acad. Sci. U.S.A.*, 102:10797–10800.
- Martinez-Vaquero, L. A., Han, T. A., Pereira, L. M., and Lenaerts, T. (2015). Apology and forgiveness evolve to resolve failures in cooperative agreements. *Scientific reports*, 5(10639).
- Nesse, R. M. (2001). Natural selection and the capacity for subjective commitment. In Nesse, R. M., editor, *Evolution and the capacity for commitment*, pages 1–44.
- Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, 314(5805):1560.
- Nowak, M. A., Sasaki, A., Taylor, C., and Fudenberg, D. (2004). Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428:646–650.
- Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge university press.
- Pacheco, J. M., Santos, F. C., Souza, M. O., and Skyrms, B. (2009). Evolutionary dynamics of collective action in n-person stag hunt dilemmas. *Proc. R. Soc. B*, 276:315–321.
- Pitt, J., Schaumeier, J., and Artikis, A. (2012). Axiomatization of socio-economic principles for self-organizing institutions: Concepts, experiments and challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 7(4):39.
- Powers, S. T., Taylor, D. J., and Bryson, J. J. (2012). Punishment can promote defection in group-structured populations. *Journal of theoretical biology*, 311:107–116.
- Rzadca, K., Datta, A., Kreitz, G., and Buchegger, S. (2015). Game-theoretic mechanisms to increase data availability in decentralized storage systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 10(3):14.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proceedings of the National Academy of Sciences of the United States of America*, 103:3490–3494.
- Sasaki, T., Okada, I., Uchida, S., and Chen, X. (2015). Commitment to cooperation and peer punishment: Its evolution. *Games*, 6(4):574–587.
- Sigmund, K. (2010). *The Calculus of Selfishness*. Princeton University Press.
- Skyrms, B. (2003). *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press.
- Traulsen, A., Nowak, M. A., and Pacheco, J. M. (2006). Stochastic dynamics of invasion and fixation. *Phys. Rev. E*, 74:11909.
- West, S., Griffin, A., and Gardner, A. (2007). Evolutionary explanations for cooperation. *Current Biology*, 17:R661–R672.
- Zhu, K. and Weyant, J. P. (2003). Strategic decisions of new technology adoption under asymmetric information: a game-theoretic model. *Decision sciences*, 34(4):643–675.

Being a leader or being the leader: The evolution of institutionalised hierarchy

Cedric Perret¹, Emma Hart¹ and Simon T. Powers¹

¹Edinburgh Napier University
C.perret@napier.ac.uk

Abstract

Human social hierarchy has the unique characteristic of existing in two forms. Firstly, as an informal hierarchy where leaders and followers are implicitly defined by their personal characteristics, and secondly, as an institutional hierarchy where leaders and followers are explicitly appointed by group decision. Although both forms can reduce the time spent in organising collective tasks, institutional hierarchy imposes additional costs. It is therefore natural to question why it emerges at all. The key difference lies in the fact that institutions can create hierarchy with only a single leader, which is unlikely to occur in unregulated informal hierarchy. To investigate if this difference can affect group decision-making and explain the evolution of institutional hierarchy, we first build an opinion-formation model that simulates group decision making. We show that in comparison to informal hierarchy, a single-leader hierarchy reduces (i) the time a group spends to reach consensus, (ii) the variation in consensus time, and (iii) the rate of increase in consensus time as group size increases. We then use this model to simulate the cost of organising a collective action which produces resources, and integrate this into an evolutionary model where individuals can choose between informal or institutional hierarchy. Our results demonstrate that groups evolve preferences towards institutional hierarchy, despite the cost of creating an institution, as it provides a greater organisational advantage which is less affected by group size and inequality.

Introduction

Why do humans choose their leaders? A meta-analysis of sixty independent studies shows that leadership effectiveness is not always correlated with leadership emergence (Judge et al., 2002). In other words, groups sometimes choose incompetent individuals as leaders. For instance, experiments on leader choice showed that “*evaluations of beauty explain success in real elections better than evaluations of competence, intelligence, likability, or trustworthiness*” (Berggren et al., 2010). Yet, despite these risks, most modern human hierarchies spend time and resources to explicitly choose leaders, even if efficient leaders are already designated by their characteristics and skills.

Social organisation plays an important role in the numerous decisions that groups take to efficiently coordinate

(Calvert, 1992). In social hierarchies, only a minority of individuals (*leaders*) are involved in the decision-making process, while the majority of individuals (*followers*) have limited influence on collective decisions. At the opposite extreme, ancient human hunter-gatherer societies were marked by a relatively equal input from all individuals in group decisions (Boehm, 2001). The transition between these two extremes is believed to have been initiated by the advent of agriculture, which created a surplus of resources and increased group size (Bocquet-Appel, 2011). In return, larger groups produced more resources thanks to division of labour and specialisation (Pindyck and Rubinfeld, 2001). On the flip side, the need for greater numbers of individuals to coordinate their actions is translated into higher costs of organisation (Calvert, 1992). Hierarchy appears as an adaptation to reduce these costs of organisation (Van Vugt et al., 2011), and in particular, to address the increase in cost of organisation as a group grows, i.e. *scalar stress* (Johnson, 1982). In large societies, the benefits created by hierarchy counterbalance the cost of any resulting inequality, eventually leading to its stable emergence.

Human adaptation to hierarchy appeared under two forms (Pielstick, 2000), expressed in (i) human behaviours (Judge et al., 2002), and (ii) human preferences. In an *informal* hierarchy, leaders and followers are defined by their intrinsic characteristics. For instance, leader effectiveness is highly correlated with particular psychological traits such as openness and extroversion (Judge et al., 2002). The second form is *formal* hierarchy where leaders and followers are appointed by group decision. For example, groups confronted by other groups in collective games explicitly elect and identify an individual as a leader (Sherif et al., 1954). We call this form here *institutional* hierarchy to stress that it is supported by institutional rules, which are created by group decision and actively enforced by monitoring and punishment (Ostrom, 1990; Hurwicz, 1996). The emergence of informal or institutional hierarchy can both be explained by the fact that they reduce costs of organisation (Powers and Lehmann, 2014; Perret et al., 2017). However, institutional hierarchies are surprisingly pervasive in modern societies, given that

they carry additional costs in comparison to informal ones. A key to this puzzle lies in the particularity of institutions which allow humans to hand-tune their behaviours, e.g. by designating a single leader, in comparison to informal hierarchies in which leaders emerge through blind evolutionary processes. However, it remains unclear whether this difference could drive the appearance of institutional hierarchies.

Currently, independent explanations for the evolution of informal and institutional hierarchy have been provided (Powers and Lehmann, 2014; Perret et al., 2017), but there is no model that investigates the competition between these two forms of social organisation. To fill this gap, we first investigate if the single-leader model found in institutional hierarchy facilitates group decision-making, i.e. leads to shorter coordination times. Second, we evaluate whether this benefit is sufficient to lead to the evolution of cultural preferences toward institutional hierarchy despite the additional costs of maintaining the institution. To do so, we describe group decision-making using an opinion-formation model (Castellano et al., 2009) that simulates a sequence of discussions between individuals, and has been shown to reflect the organisational advantage brought by leaders (Gavrilets et al., 2016). We define leaders and followers by their capacity to influence others, and analyse the effect of the number of leaders on the time a group spends to reach consensus. We then integrate this model into an evolutionary model where the time spent to reach consensus is translated into the cost of organising collective tasks. The model simulates a population structured around patches where individuals organise and carry out a collective action, which produces additional resources. Individuals can choose between informal social organisation where leaders and followers are defined by individuals' characteristics, or institutional social organisation where leaders and followers are defined by the institution. Our results show that in comparison to informal hierarchy, hierarchy with a single leader reduces (i) the consensus time, (ii) the variation in the consensus time, and (iii) the increase in consensus time as group size increases. We demonstrate that individuals evolve cultural preferences towards institutional hierarchy because it provides a greater organisational advantage than informal hierarchy, and reduces the detrimental effect of group size and inequality on the time spent to organise collective actions.

The effect of the number of leaders on group decision-making

We define social organisation by the proportion of leaders and followers present in a patch. This ranges from a perfect egalitarian organisation described by all individuals being followers or leaders, to the most hierarchical organisation with one leader and the rest of the group as followers. We define political organisation as the process by which leaders and followers are defined. The political organisation of a group can either be *informal*, i.e. leaders and followers

are defined by default by individual characteristics, or *institutional*, i.e. leaders and followers are defined by group decision (Hurwicz, 1996). It is worth noting that we constrain an institutional group to be a hierarchy, but a group can have an informal political organisation with either an egalitarian or hierarchical social organisation.

Model definition

We develop an opinion-formation model to simulate group decision-making based on previous work (Deffuant et al., 2000; Perret et al., 2017). It is an individual-based model which consists of a sequence of discussions between individuals until their opinions are close enough i.e. the group has reached a global consensus. Opinion-formation models are well-known tools to study social dynamics (Castellano et al., 2009), and have been shown to reflect the benefit of leaders on group decision-making (Gavrilets et al., 2016; Perret et al., 2017). Individuals are described by an opinion x , and a value of influence α . These are both continuous values defined on $[0,1]$. The trait α represents the influence of an individual and affects (i) the capacity of one individual to modify the opinion of another individual towards its own opinion, (ii) the reluctance of an individual to change its opinion, and (iii) the probability that an individual talks to other individuals. These three traits, i.e. persuasiveness, stubbornness and talkativeness, are highly correlated in leaders personalities (Judge et al., 2002) and are the key factors in explaining how leaders reduce time to reach consensus (Gavrilets et al., 2016). Individuals can have one of two profiles: a leader $l = 1$ with a high influence value α_l , or a follower $l = 0$ with a low influence value α_f , where $\alpha_l > \alpha_f$.

The opinion x is randomly generated at the beginning of the opinion formation. At each time-step, there is a discussion event where one speaker talks to N_l listeners to bring the followers' opinion closer to its own. The probability P of an individual i to be chosen as a speaker is an increasing function of its α value as follows:

$$P_i(t) = \frac{(\alpha_i(t))^k}{\sum_{n=1}^N (\alpha_n(t))^k}. \quad (1)$$

In the simulations we chose $k = 4$ so that in a group of large size i.e. 1000 individuals, with the most extreme hierarchy (one leader with maximum influence, $N - 1$ followers with minimum influence), the probability that a leader is chosen as a speaker is close to 90%. The speaker talks with N_l listeners randomly sampled within the other individuals in the group. This limit on the number of listeners models time constraints, and cognitive constraints of human brains (Dunbar, 1992). We assume that every individual can be chosen as a listener, i.e. the social network is a complete network, in order to avoid explicitly modelling the network structure and to keep the model tractable. We also consider that individuals interactions are not limited to individuals with close

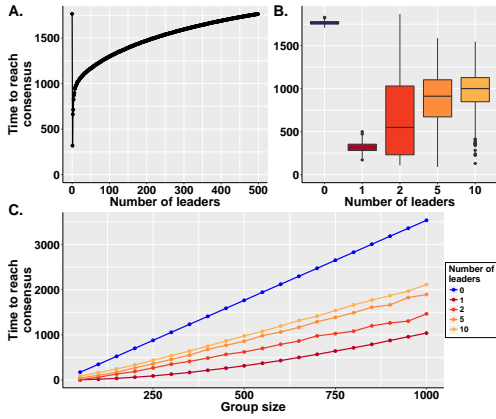


Figure 1: Effect of number of leaders on decision-making. (A,B) Mean consensus time t^* as a function of number of leaders in a group of 500 individuals (C.) Mean consensus time t^* as a function of number of leaders and group size.

opinions i.e. bounded confidence, because this model describes a consensus seeking process where individuals are willing to convince each other. During a discussion event, a listener v updates its preference to a value x'_v following the equation below, where v represents the listener and u the speaker:

$$x'_v = x_v + (\alpha_u - \alpha_v)(x_u - x_v). \quad (2)$$

We assume that the position of speaker gives a slight influential advantage over the listeners. Therefore, the minimum difference of influence $\alpha_u - \alpha_v$ is set to a positive low value, here 0.01. This assumption is necessary to avoid a systematic convergence of the preferences towards the individual with the highest α , a phenomenon not observed in real life. The individuals repeat the previous step until consensus is reached, i.e. the standard deviation of the preferences x is less than a threshold x_θ . The number of discussion events that occurred to reach consensus is called the consensus time, t^* .

Analysis

We use the opinion-formation model to investigate the difference in consensus time between hierarchy with a single leader and multiple leaders. Because of this heterogeneity, we use numerical simulations to analyse the model. The default parameters are for the consensus threshold $f_\theta = 0.05$, the number of listeners $N_l = 50$, the influence of leaders $\alpha_l = 0.75$ and the influence of followers $\alpha_f = 0.25$. The results presented are the mean across 1000 replicates. The error bars represent the standard error from the mean.

Figure 1.A confirms that hierarchy (i.e. a small number of leaders) provides an organisational advantage by reducing the consensus time. Figure 1.B shows that (i) the presence

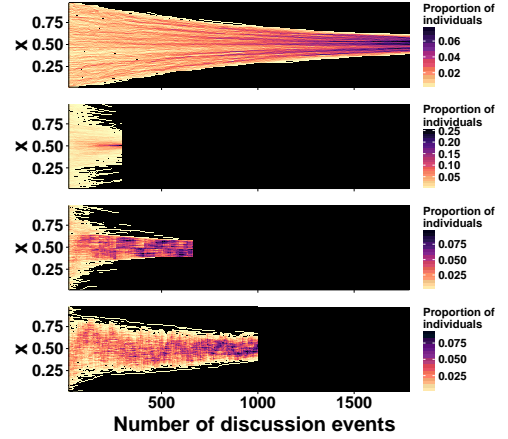


Figure 2: Density distribution of individual opinion as a function of number of discussion events for different number of leaders: from top to bottom 0, 1, 2, 10. For illustration, the difference between the opinions of leaders are set to be maximum and equidistant.

of a single leader reduces the average consensus time compared to multiple leaders, and (ii) the presence of a single leader assures a consistently lower consensus time (shown by the low variance). The presence of two leaders provides a variable advantage, which ranges from the same result as the single leader to the result from a group without a leader. As the number of leaders increases, the time to consensus increases while the variability decreases. Finally, Figure 1.C shows that the rate of increase of consensus time grows more slowly with group size when the number of leaders is smaller. In other words, the benefit of single leader increases with group size.

Figure 2 illustrates the opinion formation and the effect of the number of leaders on group decision-making. First, we see that in the absence of leaders, or with a single leader, individuals' opinions slowly and consistently converge. The presence of a single leader speeds up this process as the leader quickly convinces the majority of the group. The presence of multiple leaders creates a more heterogeneous pattern of convergence. The presence of two leaders results in the majority switching from one leader to another: leaders alternatively convince individuals from the group but neither leader has enough followers to reach consensus. When more than two leaders are present, the majority of opinion fluctuates between the different leaders. In both cases, leaders' stubbornness slows convergence of leaders towards the others, which in turn slows down the whole process. To conclude, hierarchy with a single leader clearly provides a benefit to group organisation which is (i) stronger, (ii) more constant, and (iii) more resistant to group size increase than multiple leaders. Thus, a slight change in the number of leaders can have a drastic effect on group organisation.

An evolutionary model of political organisation

We now develop an evolutionary model to investigate if the benefit of single leader hierarchy is sufficient to lead to the evolution of cultural preferences towards institutional hierarchy. Individuals carry two evolving traits: their social personality s and their preference for political organisation h . The trait s represents the intrinsic personality of an individual in a social interaction (e.g. talkativeness, boldness, charisma) and can be either dominant $s = 1$, or compliant $s = 0$. It defines an individual's influence α in informal organisation, and the probability to be chosen as a leader in institutional organisation. The trait h represents the preference in terms of political organisation of an individual: 0 represents a preference for informal organisation, and 1 a preference for institutional organisation. In addition, individuals are described by a value of influence α as described previously. The influence is either defined by an individual social personality s in an informal hierarchy, or by their assigned individual social position in institutional hierarchy (explained below). The initial values of the social personality of individuals, s , are randomly generated. The initial values of preference for political organisation h are set to 0 to represent the initial absence of institutions. The two traits s, h carried by individuals are transmitted vertically from parent to offspring, e.g. by social learning as is common in hunter-gatherer groups (Hewlett et al., 2011). They mutate following a mutation rate of μ . As these traits are assumed to be at least partly cultural, the mutation rate is higher than for a classical genetic trait. When a mutation occurs, the trait value is flipped.

Life cycle and social traits We consider an island model with a population of individuals that is subdivided into a finite number of patches N_p (Wright, 1931). The life cycle consists of discrete and non-overlapping generations as follows:

1. Individuals decide whether to create an institutional hierarchy and appoint a leader; or defaults to an informal organisation where leaders and followers roles are defined by individuals' personality s . Individuals creating an institutional hierarchy pay a cost c_h .
2. Individuals play a decision-making game on their patch as defined above (equations 1, 2). The time taken to reach consensus is translated into an opportunity cost of organisation (equation 3).
3. After consensus is reached, all individuals on a patch take part in a collective task which produces an amount of extra resource, discounted by the cost of organisation (equation 4).
4. The resource obtained from the collective task is distributed among all individuals on the patch. Leaders get

a surplus of resources modulated by a parameter d which modulates the inequality between leaders and followers (equation 5)

5. Individuals produce a number of offspring drawn from a Poisson distribution, with the mean determined by the resources received (equation 6)
6. All individuals of the previous generation perish.
7. Offspring migrate with a fixed probability m . Migrating individuals enter a patch chosen at random from the population (excluding their natal patch).

Political organisation Each group within a patch is defined by a political organisation h^* . At the beginning of each generation, individuals decide if they want to design an institutional hierarchy and appoint a leader ($h^* = 1$); this occurs if the majority of individuals in the group have a preference toward institutional hierarchy i.e. $\frac{1}{N_j(t)} \sum_i^{N_j(t)} h_{ij}(t) > 0.5$. In the absence of institutions ($h^* = 0$), a group is organised by default as an informal hierarchy.

In an institutional hierarchy, one single leader is randomly selected from the individuals with dominant personality $s = 1$ and its influence is set to α_l . The rest of the individuals within the patch adopt a follower profile and their influences are set to α_f (independently of their social personality). In an informal hierarchy, an individual's influence α is defined by its social personality with α_l for dominant individuals $s = 1$ and α_f for compliant individuals $s = 0$. In order to be sustainable, institutions require resources to monitor individuals and punish transgressors (Ostrom, 1990). Thus, individuals creating an institutional hierarchy pay a cost c_h .

Organisation by decision-making Once individuals have chosen their political organisation, they organise a collective task through group decision-making as described above. The consensus time is translated into a cost of organisation:

$$C_{oj}(t) = t_j^* C_t \quad (3)$$

The cost of organisation comes from the time dedicated to organisation instead of carrying out the actual task – groups that take too long to reach a decision may lose resources or pay other opportunity costs. This cost is modulated by C_t , which is a parameter representing the time constraint on decision making and depends of the limitation of time on the task, for instance, the speed of depletion of resources or the need to build defences before an enemy arrives. We consider here that the final decision reached has no effect on the benefit produced by the collective task – the benefit is only affected by the time taken to reach consensus.

Collective task At each generation, individuals take part in a collective task and produce additional resources $B_j(t)$:

$$B_j(t) = \frac{\beta_b}{1 + e^{-\gamma_b(N_j(t) - b_{mid})}} - C_{oj}(t). \quad (4)$$

The collective task simulates the numerous cooperative tasks realised during the lifetime of an individual. It can encompass many actions such as hunting of large game or construction of an irrigation system. The benefit is calculated from a sigmoid function described by β_b , b_{mid} and γ_b , respectively the maximum, the group size at the sigmoid's midpoint, and the steepness of the increase in the benefit induced by additional participants. We assume economy of scale in which additional participants increase the benefit super-linearly (Pindyck and Rubinfeld, 2001). But as is standard in micro-economic theory, we also make the conservative assumption that the benefit of the collective task eventually has diminishing marginal returns which overcomes the economy of scale because of other limiting factors (Foster, 2004).

Distribution of resources The resources produced by the collective task are distributed between the individuals on a given patch. The share of an individual, $p_{ij}(t)$, is then equal to:

$$p_{ij}(t) = \frac{1 + l_i(t)d}{\sum_{i=1}^{N_j} (1 + l_i(t)d)}. \quad (5)$$

Leaders ($l = 1$) receive a surplus of resources modulated by the level of ecological inequality d . For $d = 0$, the distribution within a patch is egalitarian and the influence of individuals does not affect the share of each individual. Such a scenario is close to that observed in societies of pre-Neolithic hunter-gatherers. For $d = 1$, leaders receive twice the amount a follower receives. It is assumed for simplicity that d is the same for all patches, and is determined for example by the state of technology, e.g. food storage and military technologies.

Reproduction After receiving their share of the additional resources, individuals have a number of offspring sampled from a Poisson distribution centred on the individual fitness, w . The fitness of individual i on patch j at time t is described by the following equation, where $N_j(t)$ is the total number of individual on patch j :

$$w_{ij}(t) = \frac{r_a}{1 + \frac{N_j(t)}{K}} + r_{bij}(t) - c_h h_j^* - c_n s_{ij}. \quad (6)$$

The fitness of an individual is the sum of an intrinsic growth rate r_a limited by the carrying capacity K , and additional growth rate resulting from the extra resources produced by the collective task, $r_{bij}(t)$. The fitness of individuals with institutional organisation is discounted by a cost of institution c_h , which represents the cost to monitor and enforce the institutional rule. The fitness of dominant individuals is discounted by a cost of negotiation c_n which represents the extra time and resources that an individual with dominant personality allocates to persuade others. The additional growth rate $r_{bij}(t)$ is calculated as follows:

$$r_{bij}(t) = \beta_r (1 - e^{-\gamma_r (B_j(t)p_{ij}(t))}). \quad (7)$$

The term $r_{bij}(t)$ is calculated from a logistic function described by γ_r and β_r , respectively the form and the maximum of the increase in growth rate induced by the additional resources. The additional resources are given by the total amount of benefit, $B_j(t)$, multiplied by the share the individual receives, $p_{ij}(t)$. The increase of the growth rate follows a logistic relation because of the inevitable presence of other limiting factors. After reproduction, offspring individuals migrate with a probability equal to a fixed migration rate m . Migrating individuals enter a patch chosen at random from the population (excluding their natal patch).

Analysis

We use this model to answer the following question: *Can the organisational benefit of single leader hierarchy lead to a transition from informal to institutional organisation despite the additional cost of institutions?* Because of the nonlinearities of the model, which result from the interactions of all of the variables, we analyse it using replicated numerical simulations. We focus on the effect of the following parameters: (i) the level of ecological inequality d (ii) the cost of institution C_h and (iii) the time constraint C_t . The default parameters used in the simulations, unless otherwise specified, are $N_p = 50$, $N_j(0) = 20$, $K = 20$, $r_a = 2$, $\beta_b = 10000$, $\gamma_b = 0.005$, $b_{\text{mid}} = 250$, $\beta_r = 3$, $\gamma_r = 0.05$, $\mu_m = 0.01$ and $m = 0.05$. These parameters are chosen in order to allow the transition between tribe size (50 to 100 individuals) to chiefdom size (1000 individuals). The default parameters for the group decision-making are the same as previously. Finally, we want to allow for hierarchy even when the political organisation is informal. To do so, we choose a high cost of negotiation C_N which limits the evolution of too many leaders and allows relatively stable informal hierarchy. The results presented are the mean across 32 replicates when the result is as a function of generations; and across 32 replicates and 5000 generations when the results are as a function of a parameter. Where the result is described as a mean, it is the mean value across patches. The error bars represent the standard error from the mean and are not represented when they are too small to be visible ($< 5\%$ of the maximum value).

Figure 3 demonstrates that for a moderate cost of institution, individual preferences evolve towards institutional hierarchy and thus, most of groups switch from informal to institutional hierarchy. Groups have in average only slightly more than 50% of individuals with preference toward institutional hierarchy because having any proportion above 50% has the same effect on political organisation and therefore the fitness of all individuals within the group. The small proportion of groups with informal hierarchy are explained by the cost of the institution and random mutations in individual's preferences, which can lead some groups to temporarily switch back to informal hierarchy. The prevalence of institutional hierarchy remains stable for long period (5000

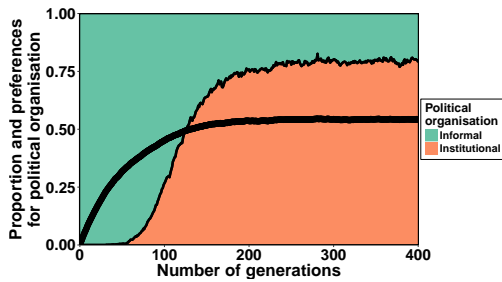


Figure 3: Evolution of the distribution of political organisation h^* (colour) and mean proportion of individuals with preferences towards institutional hierarchy across generations.

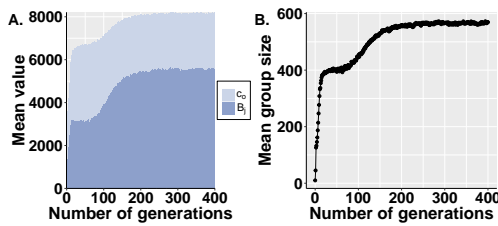


Figure 4: (A) Evolution of mean additional resources B (dark) equal to total resources produced discounted by cost of organisation C_o (light) across generations. (B) Evolution of mean group size across generations.

generations). Figure 4 shows that the total amount of resources produced and thus the group size increases through time. The cost of organisation also increases but remains low enough so that a large group provides more resources than a small group. Figure 4 shows that two increases in production and group size happen. The first corresponds to the emergence of informal hierarchy, and the second to the subsequent emergence of institutional hierarchy. This result and the results presented in Figure 5 demonstrate that institutional hierarchy allows a higher production and a larger group size. This is because a group with institutional hierarchy has (i) a lower cost of organisation and, (ii) a larger production of surplus resources due to the larger size they reach. When both types of organisation are allowed, groups reach an intermediate size and productivity because of the cost of institution and the presence of a minority of small groups with informal hierarchy. To summarise, groups developing institutional hierarchy strongly reduce their cost of organisation. They grow larger, which improves their productivity, while hierarchy limits the increase in the cost of organisation. As a consequence, these groups export a greater number of migrants, who carry their cultural preferences for institutions to other groups, leading to the global spread of institutions.

Figure 6 shows that an increase in the cost of institution

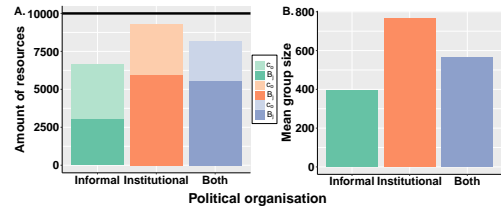


Figure 5: (A) mean additional resources B (dark) equal to total resources produced discounted by cost of organisation C_o (light), and (B) mean group size between simulations where are only allowed either institutional, informal or both organisations.

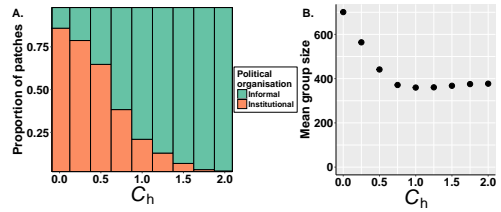


Figure 6: Distribution of (A) political organisation h^* and (B) mean group size as a function of the cost of an institution C_h .

C_h reduces the proportion of institutional hierarchy and the average group size. This result is explained by the high cost of institution overcoming the benefit brought by institutional hierarchy. However, institutional hierarchy still evolves even for a moderate cost of institutions. Indeed, a cost of 1 means that all individuals within a group need a growth rate twice higher and thus, to produce approximately twice as much resources to sustain the same fitness (see equation 6). Moreover, Figure 6 shows that individuals develop institutional hierarchy even if it doesn't significantly modify the average group size e.g. same size between $C_h = 1$ and $C_h = 2$. This is explained by single leader hierarchy providing a more constant organisational benefit than the multiple leaders of informal hierarchy. Figure 7.A shows that a larger proportion of groups develop institutional hierarchy when the time constraint on the decision making C_t is high e.g. a time limited task such as warfare. This is because the shorter consensus time brought by single leader hierarchy has more consequences on the absolute group production.

Figure 7.B shows that a higher proportion of groups develop institutional hierarchy when the level of ecological inequality d is higher. This result is explained by Figure 8 which shows that the benefit provided by institutional hierarchy persists even under high inequality. On the contrary, Figure 8.A shows that in an informal organisation, an increase in the level of inequality leads to an increase in the number of leaders. This results in a collapse of hierarchy,

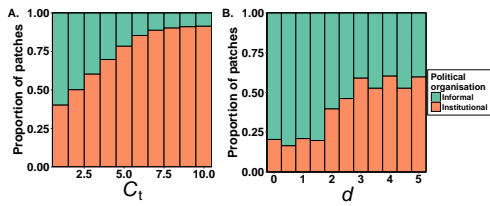


Figure 7: (A) Distribution of political organisation h^* as a function of time constraint C_t . (B) Distribution of political organisation h^* as a function of level of ecological inequality d with $C_h = 1$

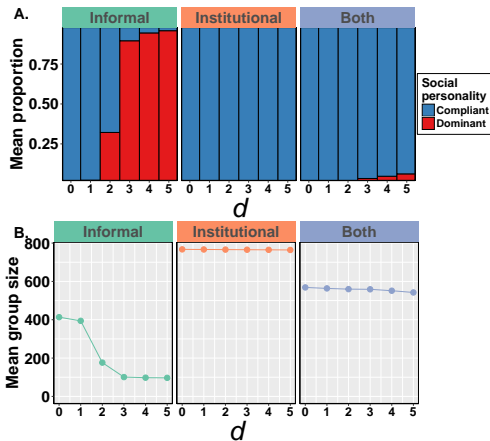


Figure 8: (A) Mean distribution of social personality and (B) mean group size as a function of the level of ecological inequality d .

a high cost of organisation and smaller group size (Figure 8.B). This difference in the effect of inequality is explained by institutional hierarchy having only one expressed leader even if multiple individuals want to be leaders. In addition, only one individual attains the status of leader and hence receives a surplus of resources, which ultimately limits the increase in number of dominant individuals.

Discussion

Human social hierarchy can be formed because individuals act as leaders and followers, i.e. informal hierarchy, or because certain individuals are chosen as leaders and followers, i.e. institutional hierarchy. But why do human groups create costly institutional hierarchies if hierarchy already emerges naturally from individual behaviours? The key difference is that single leaders can appear in institutional hierarchy designed by group decision, but are highly unlikely in informal organisation shaped by blind evolution of personality traits. Thus, in this paper, we have focused on the difference between single and multiple leader hierarchies and

have shown that institutional hierarchy with a single leader reduces more (i) the consensus time, (ii) the variation in the consensus time, and (iii) the increase in consensus time as a group grows. Our evolutionary model demonstrates that this difference results in individuals' preferences evolving towards institutional hierarchy even if this has an additional cost. To conclude, group organisation is facilitated by hierarchy but is highly intolerant to multiple leaders. This particularity provides one possible explanation for the evolution and wide spread of institutional hierarchy. To understand how critical and general is this explanation, further work should (i) explore more widely the model and its parameters and (ii) use data to test the prediction e.g. compare the cost of organisation in informal and institutional hierarchy.

The results of the opinion-formation model confirm previous work which shows that an informal leader with the features defined here speeds up consensus time (Gavrilets et al., 2016). This prior work showed that an increase in the number of leaders slows down the consensus, because it creates more stubborn individuals. Our result adds that multiple leaders also slow down the consensus, because leaders persuade each others' followers, creating conflict of interest between a large proportion of the group. It results in a more detrimental effect of multiple leaders on consensus time, which is amplified by group size. Previous theoretical work have investigated the emergence of either informal or institutional hierarchy, but ignored the competition between the two forms. Powers and Lehmann (2014) developed an evolutionary model in which individuals favour institutional hierarchy over an egalitarian organisation. Other theoretical models have shown that a similar process can drive the evolution of individuals towards leader and follower behaviours, thus creating an informal hierarchy (Johnstone and Manica, 2011; Perret et al., 2017). We confirm and connect these works by showing that institutional hierarchy can be favoured over informal hierarchy because it provides additional benefit to group decision-making, in terms of consensus time.

Our model predicts that institutional hierarchy evolves when (i) group size is high (and so productivity and cost of organisation are high), and (ii) inequality is high. These predictions fit with the environmental and social changes observed following the advent of agriculture. Agriculture created a durable surplus of resources which increased productivity and inequality (Bocquet-Appel, 2011; Mattison et al., 2016). However, our model also predicts that the productivity benefit of institutional hierarchies can be counterbalanced by a high cost of institutions. It is hard to evaluate the costs implied by institutions, but it is worth noting that they result mostly from the resources and time allocated to monitor and punish individuals not complying with the rules, i.e. here individuals trying to become leaders. Our model has shown that institutional hierarchy limits the number of individuals aspiring to become leaders, and thus suggests that

the costs of institutions remain limited even in large groups. It is worth noting that instead of competing, the two forms of political organisation could have interacted and even facilitated the development of each other. First, the development of informal hierarchy also leads to a higher group size and higher inequality. Second, the influence of an individual is in truth defined by both an individual's personality and its social position. Integrating a composite value of influence in this model could provide more insight into the interactions between these two forms of political organisation.

In this model, we have explored only one form of institution and one function of hierarchy. It would be interesting to explore other types of institutions, such as those allowing multiple levels of hierarchy, or restrict the number of people involved in the decision-making, as found in representative democracy. Other functions of hierarchy could also be investigated, e.g. to enforce cooperation (Hooper et al., 2010). However, it is worth noting that extending the model to integrate the possibility of voting for more leaders would carry similar qualitative results with individuals evolving a preference toward one leader. The presence of multiple leaders appears only later in human history, with the rise of complex states composed of multiple layers of hierarchy that constrain the behaviour of different leaders (Johnson and Earle, 2000).

Institutions are believed to be crucial innovations for the emergence of human societies. We have shown here that one of their major benefits is to provide humans with a finer tool to modify their behaviour, which can be crucial for some processes such as shown here with hierarchy. More than a new innovation, the development of institutions marks a transition in the dynamics shaping human behaviours: from long and blind evolutionary process to fast cultural dynamics.

References

- Berggren, N., Jordahl, H., and Poutvaara, P. (2010). The looks of a winner: Beauty, gender and electoral success. *Journal of public economics*, 94(1-2).
- Bocquet-Appel, J.-P. (2011). When the world's population took off: the springboard of the neolithic demographic transition. *Science*, 333(6042):560–561.
- Boehm, C. (2001). *Hierarchy in the forest: The evolution of egalitarian behavior*. Harvard University Press, Cambridge.
- Calvert, R. (1992). Leadership and its basis in problems of social coordination. *International Political Science Review*, 13(1):7–24.
- Castellano, C., Fortunato, S., and Loreto, V. (2009). Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646.
- Deffuant, G., Neau, D., Amblard, F., and Weisbuch, G. (2000). Mixing beliefs among interacting agents. Technical report.
- Dunbar, R. (1992). Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493.
- Foster, K. R. (2004). Diminishing returns in social evolution: The not-so-tragic commons. *Journal of Evolutionary Biology*, 17(5):1058–1072.
- Gavrilets, S., Auerbach, J., and Van Vugt, M. (2016). Convergence to consensus in heterogeneous groups and the emergence of informal leadership. *Scientific Reports*, 6(January).
- Hewlett, B. S., Fouts, H. N., Boyette, A. H., and Hewlett, B. L. (2011). Social learning among Congo Basin hunter-gatherers. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1168–1178.
- Hooper, P. L., Kaplan, H. S., and Boone, J. L. (2010). A theory of leadership in human cooperative groups. *Journal of Theoretical Biology*, 265(4):633–646.
- Hurwicz, L. (1996). Institutions as families of game forms. *The Japanese Economic Review*, 47(2):113–132.
- Johnson, A. W. and Earle, T. (2000). *The evolution of human societies: From foraging group to agrarian state*. Stanford University Press, Stanford.
- Johnson, G. A. (1982). Organizational structure and scalar stress. In *Theory and Explanation in Archaeology*, pages 389–421. New York: Academic Press.
- Johnstone, R. A. and Manica, A. (2011). Evolution of personality differences in leadership. *Proceedings of the National Academy of Sciences*, 108(20):8373–8378.
- Judge, T. A., Bono, J. E., Ilies, R., and Gerhardt, M. W. (2002). Personality and leadership: A qualitative and quantitative review. *Journal of Applied Psychology*, 87(4):765–780.
- Mattison, S. M., Smith, E. A., Shenk, M. K., and Cochrane, E. E. (2016). The evolution of inequality. *Evolutionary Anthropology*, 25(4):184–199.
- Ostrom, E. (1990). *Governing the commons*. Cambridge University Press.
- Perret, C., Powers, S. T., and Hart, E. (2017). Emergence of hierarchy from the evolution of individual influence in an agent-based model. In *Proceedings of the European Conference on Artificial Life 2017*, pages 348–355, Cambridge. MIT Press.
- Pielstick, C. D. (2000). Formal vs. informal leading: A comparative analysis. *Journal of Leadership Studies*, 7(3):99–114.
- Pindyck, R. S. and Rubinfeld, D. L. (2001). *Microeconomics*. Prentice Hall.
- Powers, S. T. and Lehmann, L. (2014). An evolutionary model explaining the Neolithic transition from egalitarianism to leadership and despotism. *Proceedings of the Royal Society B: Biological Sciences*, 281(1791):20141349–20141349.
- Sherif, M., Harvey, O. J., White, B. J., Hood, W. R., Sherif, C. W., and Green, C. D. (1954). Intergroup conflict and cooperation: The Robbers cave experiment. Technical report, Houghton Mifflin Company.
- Van Vugt, M., Ahuja, A., and Van Vugt, M. (2011). *Naturally selected: the evolutionary science of leadership*. HarperBusiness.
- Wright, S. (1931). Evolution in mendelian populations. *Genetics*, 16(2):97–159.

Generating urban morphologies at large scales

Juste Raimbault^{1,2,3,*} and Julien Perret^{1,4}

¹UPS CNRS 3611 ISC-PIF

²CASA, UCL

³UMR CNRS 8504 Géographie-cités

⁴Univ. Paris-Est, LaSTIG STRUDEL, IGN, ENSG

* juste.raimbault@polytechnique.edu

Abstract

At large scales, typologies of urban form and corresponding generating processes remain an open question with important implications regarding urban planning policies and sustainability. We propose in this paper to generate urban configurations at large scales, typically of districts, with morphogenesis models, and compare these to real configurations according to morphological indicators. Real values are computed on a large sample of districts taken in European urban areas. We calibrate each model and show their complementarity to approach the variety of real urban configurations, paving the way to multi-model approaches of urban morphogenesis.

Introduction

The study of forms of the built environment, and more precisely of the urban environment, has been the subject of different disciplines such as architecture, urban planning, or geography, with different approaches corresponding to various scales and processes (Moudon, 1997; Gauthier and Gilliland, 2006; Kropf, 2009). Artificial life approaches have in that context contributed to the study of generative urban processes, within the broader theoretical framework of morphogenesis (Doursat et al., 2012). Establishing typologies of urban morphologies, and understanding their link with underlying urban growth processes, is nowadays a crucial issue for sustainability as a large majority of the world population live in cities and energy consumption is closely related to urban form through e.g. mobility patterns and automobile dependence (Newman and Kenworthy, 2000).

Although there is neither a unified definition of urban form, nor unified generative models and quantitative indicators to measure it, several approaches are close to the spirit of artificial life and generative social science (Bonabeau, 1997; Epstein, 1999). Procedural modeling (Watson et al., 2008) aims at generating realistic cities, but is mostly focused on the visual impression given and does not consider realistic generative processes. It is furthermore developed largely at larger scales than the one of the district (Parish and Müller, 2001). Merrell et al. (2010) generate in that context plans for interior of buildings, whereas Cruz et al. (2017) use a cellular automaton model for building morphogenesis.

Approaches linked to urban planning have focused on the spatial distribution of land-use, at multiple resolutions (Liu et al., 2017), and proposed cellular automata models for urban sprawl, generally at the scale of the metropolitan area (Herold et al., 2003). For example, Horner (2007) proposes a link between urban form based on land-use and commuting.

Urban form can furthermore be characterized considering different components of the urban system, such as building themselves as in several examples given before, but also for example transportation networks such as road networks (Ye and Van Nes, 2014). To what extent these layers are complementary remains an open question, despite a few investigations coupling the two such as Raimbault (2018c) suggesting indeed complementary dimensions. Regarding the geometrical properties of building layouts at the scale of a district, that we denote to simplify as urban form at a large scale, systematic characterizations and generative models remains rather rare. Achibet et al. (2014) for example describes a model of co-evolution of building layout and road network.

This paper proposes a first step towards a systematic understanding of generative models of the urban form, at a large scale. The approach taken here is similar to the one taken by Raimbault (2018a), which computes urban form indicators at a mesoscopic scale (metropolitan area) and calibrates a reaction-diffusion morphogenesis model. We consider real urban configurations at the scale of the district (fixed spatial window of 500m), compute their morphological characteristics, and use these measures to calibrate different generative models of urban layouts at the same scale. Our contribution is twofold: (i) we synthesize a set of indicators relevant at this scale, and compute them on a large sample of real urban configurations in European urban areas; (ii) we provide three different generative models complementary in the type of processes taken into account, and calibrate these models on the real morphological measures. We show therein the complementarity of the different processes to produce the variety of real urban forms considered. This is to the best of our knowledge the first time several generative models at this scale are systematically compared

on a large number of real configurations through quantitative measures.

The rest of this article is structured as follows. First, we present the methods used in our work, including the measures allowing the comparison of urban forms, the proposed generative models and the method used to retrieve real urban configurations. The results of the proposed approach are then explained together with the tools used in the calibration of the models. Finally, the results are discussed.

Methods

The approach taken requires both a robust way to quantify urban forms, through indicators that can be understood as features in the sense of machine learning, and generative models.

Quantifying urban forms

The quantification of urban form is in itself covered by a vast literature. Recent work have proposed to apply deep learning techniques directly on vector data, such as Moosavi (2017) does, for a worldwide classification of road networks. Such an approach avoids the question of isolating relevant features. However, as we aim at calibrating generative models, our quantification will make more sense with interpretable measures. Boeing (2018) proposes an extensive review of existing measures from a large extent of disciplines, their implications for planning and design, and the relation with urban complexity. Webster (1995) uses image processing techniques such as contrast or Fourier analysis, to extract synthetic descriptions of urban areas from satellite imaging. Fumega et al. (2014) provide a typology of cities in relation with energy consumption in the perspective of climate change. Rode et al. (2014) relate indicators of urban form with residential heat-energy demand. Other complexity-related approaches such as fractal dimensions have been introduced as for example by Batty and Longley (1987).

In practice, we use a variety of indicators capturing different aspects, each being detailed below. We consider the local urban space as a square grid of width \sqrt{N} with cells $1 \leq i \leq N$, and an urban configuration is a binary function $s_i \in \{0; 1\}$ on these cells. For the computation of indicators, we consider underlying complementary networks, the building network B defined as nodes in centroids of occupied cells and links between two occupied direct neighbor cells (one cell unit of distance between centroids), and the free space network \bar{B} defined similarly on empty cells. This raster representation is convenient as compatible with the various types of indicators and generators as described below. We will consider $\sqrt{N} = 50$ in the following, and real windows of width $500m$.

Basic indicators Simple descriptive indicators considered are (i) the total building density $A = \frac{1}{N} \cdot \sum_i s_i$; (ii) the

number of buildings given by the number of connected components of B ; (iii) the average building area, i.e. the average size of B connected components; (iv) Moran index capturing spatial autocorrelation (see Raimbault (2018a) for its definition in a similar setting), with a simple inverse distance weight function; (v) average distance between non-empty points (which also captures a level of concentration).

Network indicators We also use indicators computed with the underlying networks: the average detour computed in the free space network \bar{B} , computed by randomly sampling 50 pairs of points in a connected component of \bar{B} and computing the ratio between the network distance and the euclidian distance $d_{\bar{B}}/d_E$. This measures captures in a way the sinuosity of streets from a mobility viewpoint. We also consider the average size of open connected areas as the average size of the connected components of \bar{B} .

Mathematical morphology indicators Finally, indicators inspired from the field of mathematical morphology (Serra, 1983) have already been applied to the quantification of urban form as for example by Pesaresi and Bianchin (2003). Mostly used in image processing, these techniques proceed to the convolution of the image with a filter for example to simplify some morphological detail, what can be interpreted as a kind of spatial smoothing. We use here an indicator based on erosion with a filter of smallest size, which here removes points which 4 closest neighbors are not also occupied. We consider the total number of steps to fully erode the image, which is linked to building size. Similarly, using the operation of dilation, which in the contrary occupies points with at least one occupied neighbor, we consider the total number of dilation steps to fully fill the grid. This captures the size of open spaces. We do not consider indicators linked to opening and closing operations, as these would require more complex filters and for example their behavior as a function of kernel size.

Combining these morphological indicators, we have a total number of 9 indicators that can be computed on any binary grid, and that we will use in the following to compare real grids with generated grids of the same size.

Generative models

We detail now the generative models introduced. The models developed below capture both bottom-up self-organizing processes and top-down planning processes. They also include different urban aspects, either filling the space with built artefacts or focusing on linking empty spaces. A null model is also considered, to ensure the relevance of the morphological measures and the adjustments on these. It consists in a random grid generator, where each cell is occupied if a random uniform number between 0 and 1 is below a density parameter d_R .

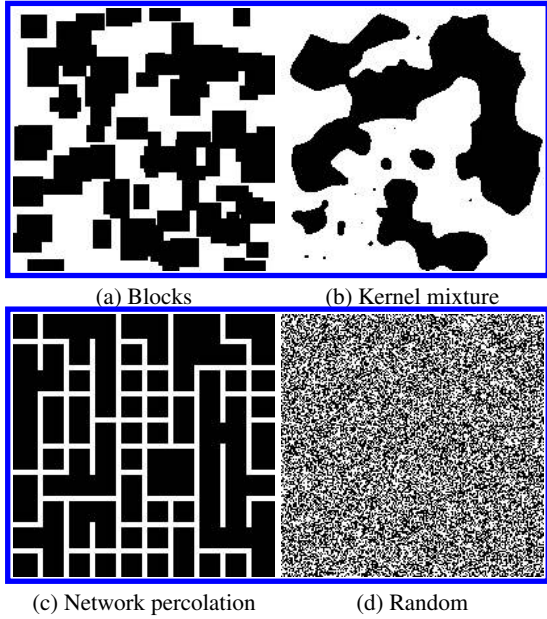


Figure 1: Examples of patterns produced by the synthetic generators.

Blocks generator The most simple “realistic” generator is similar to procedural modeling or marked point processes and distributes building blocks into the space (see Fig. 1a). Given a number N_B of blocks, random positions are drawn and block of a random height and width (with minimal value m_B and maximal value M_B as parameters) are placed at these.

Kernel mixture generator Kernel mixture are a classical way to represent the spatial distribution of population density in an urban area (Anas et al., 1998) (see Fig. 1b). They remain relevant at our scale, as they can be interpreted as a superposition of “density hotspots”, as can the planning or the self-organization of a district can be. Given a number of centers N_K , $\vec{x}_{1 \leq j \leq N_K}$ random position are drawn in the grid, at which kernels are applied, such that $s_i = \mathbb{1}_{d_i \geq \theta_K}$ where the density d_i for the point at position \vec{y}_i is given by

$$d_i = \frac{1}{N_K} \cdot \sum_j \exp(-\|\vec{x}_j - \vec{y}_i\|/d_K) \quad (1)$$

where d_K is a range parameter giving the extent of kernels.

Network percolation model The last generator we used is based on network percolation, in the spirit of capturing the constraints imposed by flows traversing a given urban area (see Fig. 1c). While the two previous generator were based on building processes, this one relies on streets, and thus on processes linked to transportation. The idea is to link a fixed number N_P of border points, which can be understood as entrances/exits of the area. Starting with a grid network



Figure 2: Samples extracted from OpenStreetMap.

without links and nodes at a regular spatial sampling (fixed with a step of 5 units in our case), an iterative procedure (i) draws a random number and adds a random link at an empty potential link if it is smaller than a parameter called the percolation probability p_P ; (ii) computes the largest connected component of the network and the number of nodes of this component on the boundary of the world; (iii) stops if this number is equal to the parameter N_P . Cells not covered by the resulting giant component are then occupied, at the exception of cells within a neighborhood L_P of a link of the giant component. This way, this component can be understood as a circulating area linking N_P entrances and exits, with a constraint on width through L_P .

Note that our generators will be “fairly compared” in terms of calibration, as they have the same number of parameters (although we do not introduce any information criteria that would yield the same penalization for overfitting). We show in Fig. 1 visual representations of some outputs of each generator, including a random generator (see Fig. 1d).

Results

Simulation results and real measures are available on the dataverse repository at <https://doi.org/10.7910/DVN/LGK0US>. Source code is available on the git repository of the project at <https://github.com/openmole/spatialdata>. The model and indicators were coded in scala language for performance purposes. This furthermore allows a seamless integration into the OpenMOLE workflow engine for model exploration (Reuillon et al., 2013), which provides methods for numerical experiments (in our case sampling methods) and transparent access to high performance computation environments (model simulation were run on the European Grid Infrastructure for an equivalent of one year and one month CPU time).

Real measures

We compute the morphological indicators given above on a large sample on real urban areas. For practical compu-

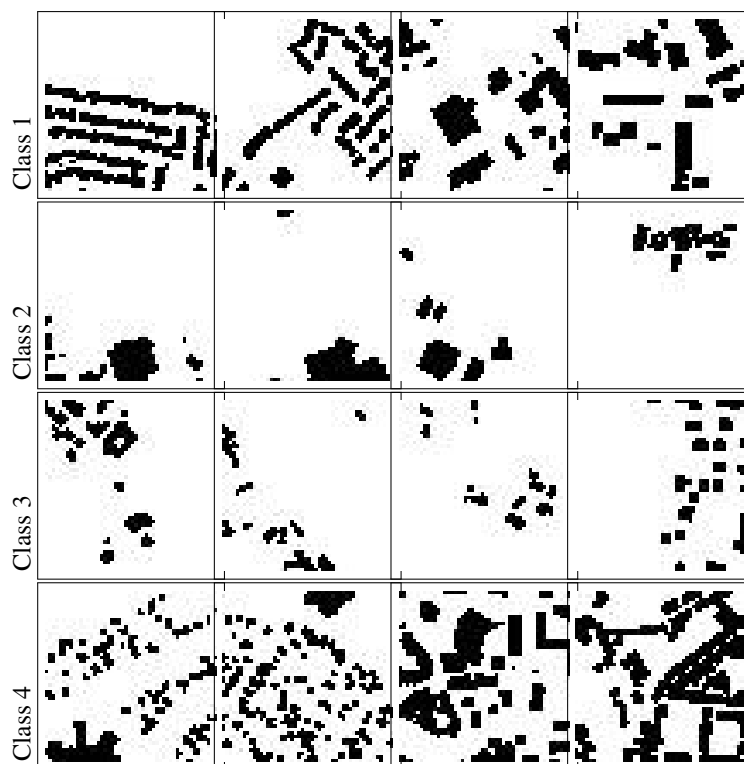


Figure 3: Typology applied to OpenStreetMap samples. The first class correspond mostly to a high density of linear buildings, recalling modern urbanism projects; the second disparate large buildings that can corresponds to industrials buildings in the outskirts of cities; the third disparate small buildings closer to periurban settlements; the fourth are denser and more complicated patterns evoking historical urban centers.

tational reasons, we restrain our geographical area of study to European functional urban areas as provided by Bretagnolle et al. (2019). We expect to already have a good representativity, although not universal, of existing urban forms with this sample, as it is known that European cities already have a significant morphological diversity (Le Néchet, 2015). We collect building layouts from OpenStreetMap (illustrated in Figure 2), as this source has been shown to have a good quality especially in Europe (Mooney et al., 2010). Using the `osmosis` tool, buildings are filtered from the openstreetmap raw dump for Europe (downloaded from <http://download.geofabrik.de/> in March 2019) and inserted into a Postgis database, which can then be efficiently queried for a specific bounding box. Indeed, although the library developed provides a direct access to the OpenStreetMap API, query limitations do not allow such a systematic sampling. We sample $N = 72,000$ points into polygons corresponding to urban areas, first by selecting the area with a uniform selection weighted by population of areas, then by drawing uniform spatial coordinates within the polygon with a polygon sampling heuristic.

After removing the empty areas and areas with a too low (lower than 0.05) or a too high (higher than 0.8) density, we end with 17,612 real points on which the morphological measures are computed. The effective dimension is rela-

tively low, echoing literature on urban form at other scales, as the first principal component on normalized indicators captures 70.3% of variance, the second a cumulated proportion of 85.9% and the third 92.8%. The order of magnitude are similar to the ones found by Schwarz (2010) for example. The first component captures low density (coefficient -0.43 for density) but clustered configurations (-0.35 for average distance), confirmed by the positive influence of dilation steps (0.44). On the contrary, the second component captures dispersed configurations (negative Moran and positive average distance) with large blocks (negative dilation steps).

We obtain a broad variety of forms, measures such as the Moran index varying between 0.02 (small disparate settlements) to 0.93 (one huge block), with a median at 0.10 (several medium size buildings). Similarly, the number of dilation steps varies from 3 (narrow streets only) to 80 (mostly open spaces) with a median at 26. The mean density is 0.21, what means that around 21% of the soil is covered with building in average in the space we sampled, confirming that the most of urban areas are not dense contrary to the highly dense centers which are a minority.

To obtain typical representative points, we proceed to an unsupervised clustering on the two first principal components of these points. Using a k-means algorithm (5000

stochastic repetitions), varying the number of clusters shows an endogenous transition in the within-cluster variance proportion, suggesting to take $k = 4$. Examples within the classes are shown and commented in Fig. 3. The centroids will be used as typical objectives for model calibration.

We can also consider the distribution of these measures within sampled urban areas. Keeping the areas with more than 10 sample points, we obtain 219 areas, for which we can compute the proportion of points within each morphological cluster. An Herfindhal diversity index on these proportions $p_k \in [0, 1]$ computed as $h = 1 - \sum_k p_k^2$ ranges between 0.31 and 0.75 with an average of 0.63, suggesting very different profiles of urban areas. A chi-squared test between the country and a discretization with 10 levels of this diversity is not significant ($p=0.7$), but the diversity index negatively correlates ($\rho = -0.11$, Fisher 95% confidence interval $[-0.24, 0.02]$) with longitude, meaning that Western cities are more diverse than Eastern cities, and more slightly with latitude ($\rho = 0.08 [-0.04, 0.21]$).

Model simulation and calibration

A simulation experiment provides an insight into the patterns produced by the different generators in the morphological space. We sample the parameter space using a Latin Hypercube Sampling, with 10000 points for each generator respectively, and with 100 stochastic repetitions for each parameter point. This sampling is achieved with scripting the models into the OpenMOLE platform (Reuillon et al., 2013).

Regarding the stochastic variability of generators, we compute for each indicator and each parameter point the sharpe ratios on repetitions, defined as the ratio between the estimated average and the estimated standard deviation. The indicators with the lowest values (high values indicate a low influence of stochastic fluctuations in comparison to variations due to parameters) are Moran index with a minimum of 0.22 and a median of 4.8, and the average detour with a minimum of 0.7 and a median of 5.2 (what could have been expected for this one as it is stochastically estimated). All other indicators have minimal sharpe ratios above 1.5 and medians above 5.4, meaning that models are overall not much sensitive to stochastic fluctuations. This confirms that considering single realisations as representing one parameter set remains reasonable.

We turn now to the comparison of generated configurations with real configurations. We work in the projected two dimensional space of the two first principal components of real points described above, in order to capture the maximum of variability in the real point cloud rather than in the simulated one. Note that working in the full indicator space makes no sense given the effective dimensions obtained (the simulated point cloud captures 93% of variance at its third principal component, which is just a bit more than the real point cloud).

The point cloud of simulated and real points is shown in

Fig. 4. We do not plot ensemble averages but all simulated points, as discussed above regarding the low influence of stochasticity. First of all, we observe that the null model consisting in random grids is far from all other points (except a tiny fraction of the percolation generator in turquoise) and in a way describes a boundary in the projected indicator space. This control confirms the relevance of projected indicators and of their comparison. Then, as expected since generators were conceived to capture different generative processes of the urban form, the point clouds of each generator are rather disjoint in the morphological space. The percolation generator produces separate clouds which correspond to different value of the link width L_P parameter, and these are disjoint from the two other generators. The exponential mixture (green) and block (red) generators do overlap in a central area, but also have their own morphological “exclusion zone”, where the forms can not be generated by other generators considered here.

When looking at the real point cloud, we see that most of it is covered by some generated points, and that generators are complementary to approach all covered points. This is an important result in line with the targeted complementarity of generative processes, and advocates for multi-modeling in urban morphogenesis. Interestingly, there is an area not covered, corresponding to the transition between the percolation generator (narrow streets) and the block generator.

For each centroid of the clusters in the real point cloud described above, that can be considered as a typical calibration objective, we provide example of the closest real configuration and the closest simulated one. Visually, forms are rather satisfying, at the exception of the percolation generator fitting a complicated urban center. Indeed, this centroid (number 4) is at the boundary of the percolation point cloud, and the real point cloud is more difficultly captured in this area compared to the block and mixture generators.

To quantify the level of calibration of each generator regarding each centroid, and in average regarding stochastic repetitions, we provide in Table 1 the aggregated minimal values of distances, for each generator and each calibration objective, with their standard deviations. This mainly confirms the previous results, with however interesting variations: (i) for the second centroid, the exponential mixture is in average no longer the best, and furthermore has a higher variability; (ii) centroid three and four are the easiest to reach, despite the latest being in the boundary; (iii) the percolation generator performs well on this point and has a very low variability.

This experiment has therefore shown the possibility to calibrate the generative models on morphological measures against real configurations, and furthermore unveils their complementarity to approach the diverse existing forms.

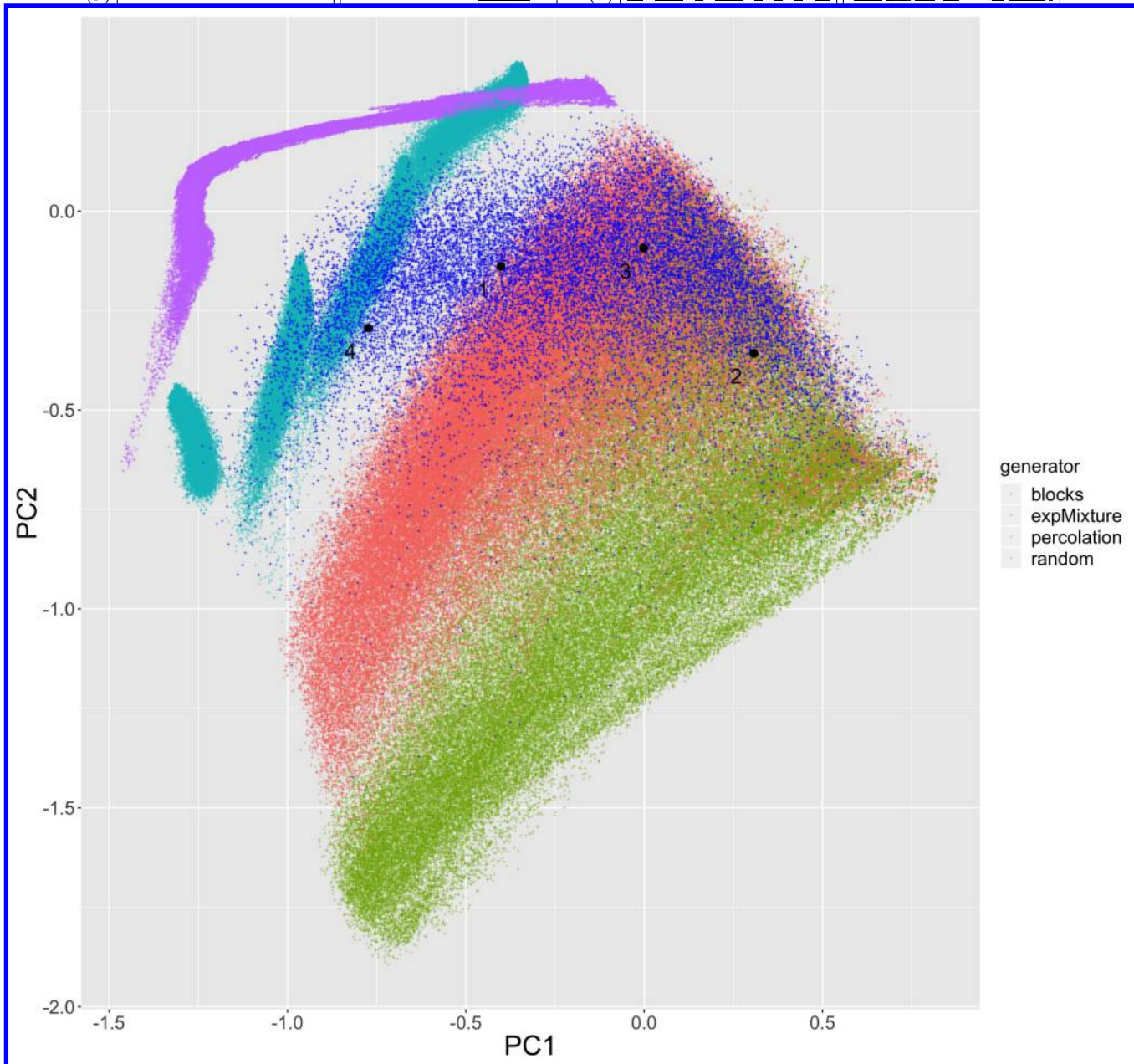
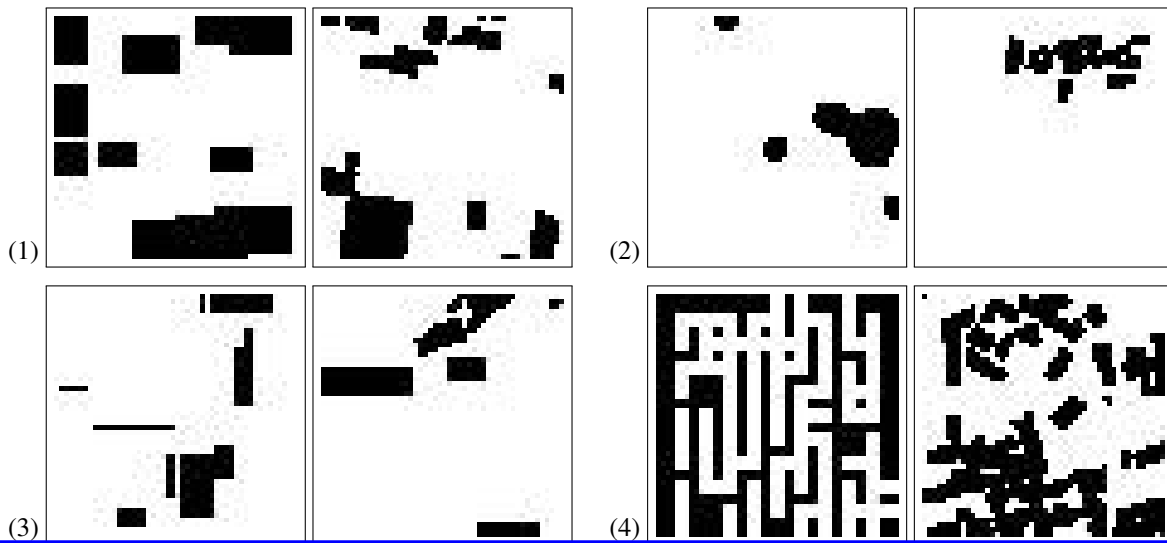


Figure 4: Comparison of real morphologies with patterns produced by the synthetic generators. All points are projected into the first two components computed on the real measures only. Points in dark blue correspond to real configurations. The type of generator is given by the color in legend. Dark circle correspond to the centroids of clusters on which the calibration is done, their number giving the corresponding configurations shown above. For each, we successively show the best synthetic configuration ((1) block generator with $N_B = 12, m_B = 5, M_B = 12$, (2) exponential mixture generator with $N_K = 6, d_K = 3.8, \theta_K = 0.42$, (3) block generator with $N_B = 11, m_B = 1, M_B = 14$, (4) percolation generator with $p_P = 0.35, N_P = 5, L_P = 3.7$), and the real configuration from OSM beside each.

	Random	Blocks	Exp. Mixture	Percolation
Centroid 1	0.424 ± 0.011	0.106 ± 0.063	0.303 ± 0.101	0.325 ± 0.019
Centroid 2	0.809 ± 0.022	0.164 ± 0.099	0.184 ± 0.141	0.947 ± 0.019
Centroid 3	0.428 ± 0.019	0.095 ± 0.054	0.109 ± 0.064	0.541 ± 0.019
Centroid 4	0.515 ± 0.005	0.311 ± 0.077	0.589 ± 0.149	0.083 ± 0.025

Table 1: Aggregated distance in the morphological space, for each generator and each calibration objective (real clusters centroids). Euclidian distance in the projected space are aggregated in average on stochastic repetitions, and the minimal average value is reported with its standard deviation.

Discussion

Our approach provides a first step towards systematic modeling of generative processes of urban form at large scales. Some direct limitations could be tackled in a short term. Testing slightly different processes and heuristics in generator may be a way to cover the part of the real point cloud which is missed by our generators. As it seems correspond to complicated urban centers, it may be however complicated without more elaborated models. Also, we did not use minimization algorithms to calibrate the generators, and the and a further step would consist in checking the robustness of our result using such optimization heuristics (genetic algorithms for example), but also diversity algorithms such as pattern space exploration proposed by Chérel et al. (2015), to ensure the effective feasible space of each generator.

This work can also be extended in several ways. First of all, we focused on the built environment but neglected transportation infrastructures, whereas spatial network morphogenesis models have been proposed for example by Courtat et al. (2011) or Raimbault (2018b) in a multi-modeling approach. Taking into account multiple dimensions of the urban system is an important extension and hybrid models such as co-evolution models (Raimbault et al., 2014) should be investigated. Our approach can also be a preliminary step towards the study of urban sustainability issues, for example the relations between urban form and energy consumption (Le Néchet, 2015). Extending the generators with the third dimension, i.e. taking into account building heights as done by Brasebin et al. (2017), could also be an important component for the study of local energy efficiency.

Furthermore, we tested the complementarity of generators only in a static way. Adaptive and dynamic generators, combining processes of different nature within the same model with an endogenous switching or combination, would be an important direction to better understand urban morphogenesis. In the same context, the generators compared here had all the same number of parameters, but richer generators implying different numbers would require the use of information criterions to avoid overfitting, which, however, remains an unsolved issue for such generative simulation models (Piou et al., 2009).

Finally, as extensively reviewed above, the way to quan-

tify urban form strongly depends on the scale considered. A more integrative understanding of it would require multi-scale approaches able to relate these different definition and measures within a single multi-scalar framework.

Conclusion

We have proposed here a new insight into the generative simulation of urban morphologies at large scales, namely the scale of the district considering the layout of buildings. After computing morphological measures on a large sample of real urban areas, we showed the complementarity of different generators capturing various aspects of urban morphogenesis processes. Despite not implying generative agents (developers, inhabitants, companies) and thus staying close to procedural modeling, this work however paves the way towards a more systematic understanding of generative processes of urban form at this scale.

Acknowledgements

Results obtained in this paper were computed on the vo.complex-system.eu virtual organization of the European Grid Infrastructure (<http://www.egi.eu>). We thank the European Grid Infrastructure and its supporting National Grid Initiatives (France-Grilles in particular) for providing the technical support and infrastructure. This work is part of DynamiCity, a FUI project funded by BPI France, Auvergne-Rhône-Alpes region, Ile-de-France region and Lyon metropolis.

References

- Achibet, M., Balev, S., Dutot, A., and Olivier, D. (2014). A model of road network and buildings extension co-evolution. *Procedia Computer Science*, 32:828–833.
- Anas, A., Arnott, R., and Small, K. A. (1998). Urban spatial structure. *Journal of economic literature*, 36(3):1426–1464.
- Batty, M. and Longley, P. A. (1987). Fractal-based description of urban form. *Environment and planning B: Planning and Design*, 14(2):123–134.
- Boeing, G. (2018). Measuring the complexity of urban form and design. *URBAN DESIGN International*, 23(4):281–292.
- Bonabeau, E. (1997). From classical models of morphogenesis to agent-based models of pattern formation. *Artificial life*, 3(3):191–211.

- Brasebin, M., Chapron, P., Chérel, G., Leclaire, M., Lokhat, I., Perret, J., and Reuillon, R. (2017). Apports des méthodes d'exploration et de distribution appliquées à la simulation des droits à bâtir. In *Spatial Analysis and GEomatics 2017*.
- Bretagnolle, A., Guerois, M., and Pavard, A. (2019). Following the population of european urban areas in the last half century (1961-2011): the tradeve database. *Forthcoming in Cybergeog, European Journal of Geography*.
- Chérel, G., Cottineau, C., and Reuillon, R. (2015). Beyond corroboration: Strengthening model validation by looking for unexpected patterns. *PLoS ONE*, 10(9):e0138212.
- Courtat, T., Gloaguen, C., and Douady, S. (2011). Mathematics and morphogenesis of cities: A geometrical approach. *Physical Review E*, 83(3):036106.
- Cruz, C., Kirley, M., and Karakiewicz, J. (2017). Generation and exploration of architectural form using a composite cellular automata. In *Australasian Conference on Artificial Life and Computational Intelligence*, pages 99–110. Springer.
- Doursat, R., Sayama, H., and Michel, O. (2012). *Morphogenetic engineering: toward programmable complex systems*. Springer.
- Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, 4(5):41–60.
- Fumega, J., Niza, S., and Ferrão, P. (2014). Identification of urban typologies through the use of urban form metrics for urban energy and climate change analysis.
- Gauthier, P. and Gilliland, J. (2006). Mapping urban morphology: a classification scheme for interpreting contributions to the study of urban form. *Urban Morphology*, 10(1):41.
- Herold, M., Goldstein, N. C., and Clarke, K. C. (2003). The spatiotemporal form of urban growth: measurement, analysis and modeling. *Remote sensing of Environment*, 86(3):286–302.
- Horner, M. W. (2007). A multi-scale analysis of urban form and commuting change in a small metropolitan area (1990–2000). *The Annals of Regional Science*, 41(2):315–332.
- Kropf, K. (2009). Aspects of urban form. *Urban Morphology*, 13(2):105.
- Le Néchet, F. (2015). De la forme urbaine à la structure métropolitaine: une typologie de la configuration interne des densités pour les principales métropoles européennes de l'audit urbain. *Cybergeog: European Journal of Geography*.
- Liu, X., Liang, X., Li, X., Xu, X., Ou, J., Chen, Y., Li, S., Wang, S., and Pei, F. (2017). A future land use simulation model (flus) for simulating multiple land use scenarios by coupling human and natural effects. *Landscape and Urban Planning*, 168:94–116.
- Merrell, P., Schkufza, E., and Koltun, V. (2010). Computer-generated residential building layouts. In *ACM Transactions on Graphics (TOG)*, volume 29, page 181. ACM.
- Mooney, P., Corcoran, P., and Winstanley, A. C. (2010). Towards quality metrics for openstreetmap. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 514–517. ACM.
- Moosavi, V. (2017). Urban morphology meets deep learning: Exploring urban forms in one million cities, town and villages across the planet. *arXiv e-prints*, page arXiv:1709.02939.
- Moudon, A. V. (1997). Urban morphology as an emerging interdisciplinary field. *Urban morphology*, 1(1):3–10.
- Newman, P. and Kenworthy, J. (2000). Sustainable urban form: the big picture. *Achieving sustainable urban form*, pages 109–120.
- Parish, Y. I. H. and Müller, P. (2001). Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 301–308, New York, NY, USA. ACM.
- Pesaresi, M. and Bianchin, A. (2003). Recognizing settlement structure using mathematical morphology and image texture. *Remote sensing and urban analysis: GISDATA*, 9:46–60.
- Piou, C., Berger, U., and Grimm, V. (2009). Proposing an information criterion for individual-based models developed in a pattern-oriented modelling framework. *Ecological Modelling*, 220(17):1957–1967.
- Raimbault, J. (2018a). Calibration of a density-based model of urban morphogenesis. *PloS one*, 13(9):e0203516.
- Raimbault, J. (2018b). Multi-modeling the morphogenesis of transportation networks. In *Artificial Life Conference Proceedings*, pages 382–383. MIT Press.
- Raimbault, J. (2018c). An urban morphogenesis model capturing interactions between networks and territories. In *Mathematics of Urban Morphogenesis, D'Acci L., ed. Springer Birkhauser Mathematics*.
- Raimbault, J., Banos, A., and Doursat, R. (2014). A hybrid network/grid model of urban morphogenesis and optimization. In *4th International Conference on Complex Systems and Applications*, pages 51–60.
- Reuillon, R., Leclaire, M., and Rey-Coyrehourcq, S. (2013). Openmole, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8):1981–1990.
- Rode, P., Keim, C., Robazza, G., Viejo, P., and Schofield, J. (2014). Cities and energy: urban morphology and residential heat-energy demand. *Environment and Planning B: Planning and Design*, 41(1):138–162.
- Schwarz, N. (2010). Urban form revisited—selecting indicators for characterising european cities. *Landscape and Urban Planning*, 96(1):29–47.
- Serra, J. (1983). *Image analysis and mathematical morphology*. Academic Press, Inc.
- Watson, B., Müller, P., Veryovka, O., Fuller, A., Wonka, P., and Sexton, C. (2008). Procedural urban modeling in practice. *IEEE Computer Graphics and Applications*, 28(3):18–26.
- Webster, C. (1995). Urban morphological fingerprints. *Environment and Planning B: Planning and design*, 22(3):279–297.
- Ye, Y. and Van Nes, A. (2014). Quantitative tools in urban morphology: Combining space syntax, spacematrix and mixed-use index in a gis framework. *Urban morphology*, 18(2):97–118.

Fairness in Multiplayer Ultimatum Games Through Moderate Responder Selection

Fernando P. Santos¹ and Daan Bloembergen²

¹Department of Ecology and Evolutionary Biology, Princeton University, New Jersey, USA
fpsantos@princeton.edu

²Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
d.bloembergen@cwi.nl

Abstract

We study the evolution of fairness in a multiplayer version of the classical Ultimatum Game in which a group of N Proposers offers a division of resources to M Responders. In general, the proposal is rejected if the (average) proposed offer is lower than the (average) response threshold in the Responders group. A motivation for our work is the exchange of flexibilities between smart energy communities, where the surplus of one community can be offered to meet the demand of a second community. In the absence of any Responder selection criteria, the co-evolving populations of Proposers and Responders converge to a state in which proposals and acceptance thresholds are low, implying an unfair exchange that favors Proposers. To circumvent this, we test different rules which determine how Responders should be selected, contingent on their declared acceptance thresholds. We find that selecting moderate Responders optimizes overall fairness. Selecting the lowest-demanding Responders maintains unfairness, while selecting the highest-demanding individuals yields a worse outcome for all due to frequent rejected proposals. These results provide a practical message for institutional design and the proposed model allows testing policies and emergent behaviors on the intersection between social choice theory, group bargaining, competition, and fairness elicitation.

Introduction

Many social dilemmas in society can be formulated and studied using game theoretic methods (Gintis, 2000). In particular, the question how cooperation can come about in a society of self-interested individuals has attracted considerable interest in the research community (Axelrod and Hamilton, 1981; Hofmann et al., 2011; Rand and Nowak, 2013; Ranjbar-Sahraei et al., 2014; Santos et al., 2018). Typically such social dilemmas are cast as a normal form game, in which a set of players simultaneously and without prior communication choose an action to play, and the resulting joint action determines the payoff to each player. Despite the simplicity of these one-shot interactions, normal form games can still capture many of the intricate dynamics of complex strategic interactions (Axelrod and Hamilton, 1981; Skyrms, 2004).

One example of such a game is the Ultimatum Game (UG) (Güth et al., 1982), in which one player, the Proposer, offers a certain split of a resource to a Responder, who decides to accept or reject the offer. If accepted, the players receive their share per the offer; if rejected both players receive nothing. We propose and study a Multiplayer version of the classical Ultimatum Game, in which a group of N Proposers offers a division of resources to a group of M Responders. Henceforth we refer to this interaction as **NvM-person Ultimatum Game (NM-MUG)**. While a multiplayer version of UG was previously discussed in the context of one Proposer and N Responders (Santos et al., 2015, 2016, 2019), considering proposals by groups of Proposers is relevant in the context of rival public goods division, where 1) Proposers may be tempted to free-ride and lower their proposals expecting other Proposers to compensate and 2) the group sizes of Proposers and Responders may not match, reducing the per-capita share in one of the groups. In general, we assume that a proposal is rejected if the (average) proposed offer is lower than the (average) response threshold in the Responders group. We study under which conditions a fair outcome is achieved, in which Proposers offer a substantial split to the Responders. In particular, we study the mechanism by which the Responders are selected from the population in order to elicit the best deal.

The NM-MUG can be used to study social settings in which groups of people wish to negotiate a deal. For example, deals between companies or between national legislative bodies are often discussed by committees representing each side, and as a result the selection of committee members with specific individual strategies can have a great influence on the final result (Hagan et al., 2001). Multiplayer versions of the Ultimatum Game are also played in the context of group buying (Jing and Xie, 2011). A specific example motivating our work are smart energy communities, such as the Amsterdam pilot sites *Schoonschip*¹ and *De Ceuve*², in which a number of households share a single point of coupling with the national energy grid. Behind this point

¹<http://schoonschipamsterdam.org/en/>

²<https://www.jouliette.net/>

of coupling, the households can exchange energy flexibilities (demand and supply) locally and thus more efficiently (Chakraborty et al., 2018). The summed remaining flexibility of each community could be used in negotiation with a different community, as a second layer of local or regional energy exchange (Lezama et al., 2018). This suggests a multiplayer bargaining game between two groups (the Proposing and Responding communities) which fits well within the general layout of the NM-MUG.

We simulate this scenario by means of a co-evolutionary process in which groups of Proposers and Responders are repeatedly selected from separate populations. The NM-MUG is used to compute the resulting fitness of individuals in each population, which then evolve following imitation dynamics and mutation. When selecting randomly composed groups of Proposers and Responders (that is, each individual has an equal probability of being selected for the group of Proposers or Responders), we find that the average offer of Proposers and acceptance thresholds of Responders co-evolves to an unfair state where Proposers get (almost) all the share. From this baseline, it is possible to test mechanisms for selecting the Responder groups, aiming to find arrangements that equalize the average gains of both populations. This model can thus lay out directions for future research in the areas social choice, group bargaining, competition, and the emergence of fairness in co-evolving communities.

Background and Related Work

The **Ultimatum Game (UG)** is a well-known interaction paradigm, widely used to evince the conflict between payoff maximization and fairness — and the puzzling human preference for the latter (Güth et al., 1982). As mentioned in the previous section, in this game two players interact in two distinct roles. One is called the *Proposer* and the other is denominated *Responder*. The game is composed by two sub-games, one played by each role. First, some amount of a given resource, e.g. money, is conditionally endowed to the Proposer; this agent must then suggest a division with the Responder. Secondly, the Responder will accept or reject the offer. The agents divide the money as it was proposed, if the Responder accepts. By rejecting, none of them will get anything. The strategy set of the Proposers comprises any possible division of the resource. The strategies of the Responders are acceptance or rejection, contingent on the offer made. Often, Responders’ strategies are assumed to be probabilities of acceptance that are non-decreasing on the offer made. Frequently it is assumed that any Responder decision is codified in a threshold of acceptance: below this threshold offers are rejected (i.e., accepted with probability 0) and above the threshold offers are accepted with probability 1 (Page et al., 2000).

While the UG is a sequential game usually expressed in extensive-form, by having Responders deciding on their thresholds of acceptance in advance we can also formalize

this interaction as a normal-form game. In either case, the rational behaviour in the UG can be anticipated using traditional game-theoretical equilibrium analysis. Of special interest in this setting is the *sub-game perfect equilibrium* (Osborne et al., 2004), which can be inferred by applying the method of backward induction. The Responder, facing the decision of rejecting (earn 0) or accepting (earn some money, even if a really small quantity), would always prefer to accept. Secure about this certain acceptance, the Proposer will offer the minimum possible, maximizing her own share, thus yielding the equilibrium in which both the offer and the acceptance threshold are as close as possible to zero.

The UG is a 2-person game, however, many real-world situations require bargaining within (and between) groups of individuals. Here we focus on a multiplayer extension of the ultimatum game in which a group of N Proposers offers a division of resource to a group of M Responders (NM-MUG). A previous formalization of **Multiplayer Ultimatum Game (MUG)**, close to the one that we follow here, was proposed by Santos et al. (2015). In that work, a single Proposer makes an offer to a group of Responders. Individually, each Responder in the group states acceptance or rejection; the group of Responders as a whole accepts the offer provided that a minimum number of acceptances exist. A more recent study resorts to reinforcement learning (the Roth-Erev algorithm) to show that higher proposals are likely to emerge when stricter group decision rules (requiring more accepting Responders for group acceptance) are considered (Santos et al., 2016), also in the context of 1 Proposer versus N Responders. An alternative multiplayer (3-person) formulation of the UG was proposed by Takesue et al. (2017). Also, in a seminal work, Fehr and Schmidt (1999) explicitly considered the effect of competition between Proposers and Responders in a market game closely related with the UG. In this game, either 1) a group of sellers (Proposers) compete to sell one unit of a good to a buyer (Responder); or one Proposer suggests an offer that leads many Responders to compete against each other to accept it. In these market games, subjects tend to adopt unfairer strategies, differently to what happens with the 2-person UG and as predicted by the rationality self-interest model.

Nevertheless, both in the 2-person and the multiplayer ultimatum game, the predictions assuming perfect rationality were challenged by experimental and theoretical works (Fehr and Schmidt, 1999; de Jong and Tuyls, 2011; Santos et al., 2019). Instead of resorting to equilibrium notions of classical game theory to study the behavior of agents when interacting in a multi-Proposer multi-Responder ultimatum game, we adopt methods from population ecology, such as **evolutionary game theory (EGT)**. EGT has been used to analyze strategic interactions in several domains such as auctions (Phelps et al., 2004) or market dynamics (Bloembergen et al., 2015) (as an example). In a social context, EGT can describe individuals who revise their strategies

through social learning, being influenced by the behaviours and achievements of others (Sigmund, 2010). One of the most traditional tools to describe the dynamics of an evolutionary game model is the replicator equation (Taylor and Jonker, 1978). This equation, justified in a context of trait evolution in biology or cultural evolution across human societies, assumes that populations are infinite and evolution proceeds favouring strategies that offer a fitness higher than the average fitness of the population. However, it has also been argued that the replicator equation might not be an accurate model of human behaviour, due to its assumption of an infinite and well-mixed population, and that agent-based models might be more appropriate to study the social dynamics of fairness (Alexander, 2000).

Given these considerations, here we analyze the NM-MUG resorting to an agent-based model that similarly assumes that strategies performing better than average are selected over time. For that, we consider a pairwise comparison rule (Traulsen et al., 2006). As will be clarified below, we consider a finite population of agents. After playing several rounds, agents revise their strategy by observing a role-model agent, randomly picked from the population. Imitation (i.e., copying the strategy used by the role-model) occurs with a probability that grows with fitness difference: strategies performing better have a higher probability of being imitated. Under certain limits (large population size and low selection intensity) the replicator dynamics is recovered in this process (Traulsen et al., 2006).

NvM-Person Ultimatum Game

Let us start by describing the NvM-person (i.e., multi-Proposer, multi-Responder) Ultimatum Game, the interaction paradigm used throughout this paper. In any given NM-MUG interaction a group of Proposers makes an offer to a group of Responders. The offer made by the group results from a function of individual offers of Proposers in the group (e.g., the average); this offer is accepted if it is higher than a function of Responders' individual acceptance thresholds (e.g., if the offer is higher than the maximal threshold — guaranteeing that every Responder in the groups is satisfied — or if, again, the offer is higher than the Responders' average threshold). In case of acceptance, each Proposer receives the share she did not offer, which stresses the social dilemma in the Proposers' group: individually, each one has interest in offering the minimum possible but, in order to prevent a rejection, it is beneficial for all to have the largest possible collective offer.

In the context of smart energy communities, each group can be seen as a possible community, while the selection mechanism determines the attitude of its members. The proposals and response thresholds can be thought of as some combination of kWh and price, or the difference to the market price, illustrating the potential gain from the interaction.

Formally, we model the NM-MUG by two populations

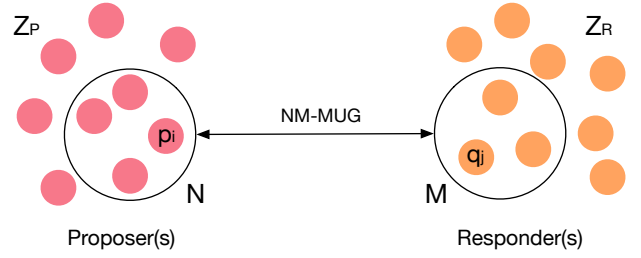


Figure 1: The NvM-person Ultimatum Game (NM-MUG). Groups of N Proposers and M Responders are drawn from a population of Proposers Z_P and Responders Z_R , respectively. The groups' joint proposal and threshold for acceptance determine the success of the interaction.

Z_P and Z_R , representing the Proposers and Responders, respectively (see Figure 1). Each individual i in the population of Proposers is defined by her personal proposal value $p_i \in [0, 1]$, for $i \in Z_P$. Similarly, Responders are defined by their individual threshold of acceptance $q_j \in [0, 1]$, for $j \in Z_R$. At each iteration, a group of Proposers $N \subseteq Z_P$ and Responders $M \subseteq Z_R$ is selected, following predefined rules³. These groups induce a joint proposal $\bar{p} = \mathcal{P}(N)$ and joint Responder threshold $\bar{q} = \mathcal{Q}(M)$. In a simple scenario (such as the one we will consider below) \mathcal{P} and \mathcal{Q} are the average function, i.e. $\bar{p} = |N|^{-1} \sum_{i \in N} p_i$ and $\bar{q} = |M|^{-1} \sum_{j \in M} q_j$. The proposal is accepted iff $\bar{p} \geq \bar{q}$. The question is now: how to select the groups of Proposers and Responders from each population?

Base Scenario

In the base scenario, the Proposers, forming a group of fixed size N , are selected randomly from Z_P . The joint proposal offered by the group is taken to be the average proposal of individuals in the group, $\bar{p} = |N|^{-1} \sum_{j \in N} p_j$. The Responder group M is composed of those Responders that are willing to accept \bar{p} , such that $j \in M : q_j \leq \bar{p}$.

In this case, the Responders in M will have a payoff

$$U_i^R = \min(\bar{p}, \bar{p} \frac{N}{M}), \quad (1)$$

whereas all Responders outside M earn 0. The min operator signifies that the Responders cannot jointly receive more than what the Proposers offer, nor can one individual consume more than a unit share. At the same time, Proposer i taking part in the collective proposal by group N , offered to the group of Responders M , will have a payoff of

$$U_i^P = \min(1 - p_i, (1 - p_i) \frac{M}{N}), \quad (2)$$

³To simplify notation, we use N and M interchangeably as the group size of Proposers and Responders, respectively, as well as groups of selected Proposers and Responders. When an explicit distinction is necessary we use $|N|$ and $|M|$ to denote group sizes.

where p_i is the proposal by individual i . Again, the min operator ensures that Proposers cannot jointly offer more than the Responders accept; unit offers that are not accepted are lost in the context of the deal. This loosely reflects a typical scenario in which flexibilities are exchanged between smart energy communities (Lezama et al., 2018), where each individual household has a maximum amount of flexibility it can offer, and the total sum of flexibilities exchanged between the communities should balance out in the deal.

Responder Competition Scenario

As we detail below (Section “Experiments and Results”) allowing any individual $i \in Z_P$ to take part in the group of Responders (those that will accept or reject the offer and profit from it) has the pernicious effect of inducing a long-term reduction in the average values of q adopted in the Responders’ population which, in turn, incentivises the Proposers to lower their p and enact highly unfair offers. Many institutional arrangements affecting the process of Responder selection can be tested, departing from the base scenario presented above. For now, we discuss the role of Responder competition based on a declared threshold of acceptance — partly inspired by Fehr and Schmidt (1999). While Proposers are still randomly selected, we sort the Responders’ declared thresholds of acceptance, partitioning the Responders based on this ordering, and select for the group the individuals declaring the thresholds ranked from the m^{th} to the $(m + M - 1)^{th}$ ascending position. As an example of extreme cases, $m = 0$ and $M = 10$ means that the 10 lowest acceptance thresholds are selected and, in a population of 100 Responders, $m = 90$ and $M = 10$ means that the 10 highest acceptance thresholds are selected.

In this case, assuming that $|N|$ and $|M|$ are fixed externally, proposals are accepted only whenever $\bar{p} \geq \bar{q}$, where $\bar{q} = |M|^{-1} \sum_{j \in M} q_j$, and M is formed by the demands q_i which, after being sorted in an ascending order, stand in the positions ranging from the m^{th} to the $(m + M - 1)^{th}$ position. We study the evolutionary trajectories of strategy adoption when different rules for the selection of Responders are introduced (i.e., different m and M).

Evolutionary Dynamics

In order to study the evolutionary dynamics associated with each Responder selection rule (m and M), we implement an agent-based model in which individuals resort to social learning to adapt their behavior over time (Algorithm 1). Initially, values of p and q characterizing each agent are sampled from a uniform distribution. For a large number of generations, individuals will adapt their values of p and q . In each generation, $|Z_P| + |Z_R|$ individuals are sampled with replacement, following a uniform probability; with a probability μ the selected individual will randomly explore the strategy space, adopting a random value of p (if Proposer) or q (if Responder). This is akin to a mutation in genetic

Algorithm 1: Pseudo-code of the main cycle of our simulations. Algorithm 2 sketches how $fitness(\cdot)$ is computed.

```

Initialize all  $p_i \in Z_P, q_i \in Z_R = X \sim \mathcal{U}(0, 1)$ 
for  $t \leftarrow 1$  to  $Gen_s$  do Main cycle of interaction and
strategy update:
  for  $j \leftarrow 1$  to  $Z_P + Z_R$  do Select agent to update:
    if  $X \sim \mathcal{U}(0, 1) < Z_P / (Z_P + Z_R)$  then Update
    Proposer strategy:
      /* Sample two agents from
        Proposer population */
       $A \leftarrow X \sim \mathcal{U}(1, Z_P)$  (agent to update)
       $B \leftarrow X \sim \mathcal{U}(1, Z_P)$  (model agent)
    else Update Responder strategy:
      /* Sample two agents from
        Responder population */
       $A \leftarrow X \sim \mathcal{U}(1, Z_R)$  (agent to update)
       $B \leftarrow X \sim \mathcal{U}(1, Z_R)$  (model agent)
    if  $X \sim \mathcal{U}(0, 1) < \mu$  then Mutation:
       $p_A \leftarrow X \sim \mathcal{U}(0, 1)$ 
    else Imitation:
       $f_A \leftarrow fitness(A)$ 
       $f_B \leftarrow fitness(B)$ 
       $prob \leftarrow 1 / (1 + e^{-\beta(f_B - f_A)})$ 
      if  $X \sim \mathcal{U}(0, 1) < prob$  then
         $p_A \leftarrow p_B + \text{imitation error} \sim \mathcal{U}(-\epsilon, \epsilon)$ 

```

evolution. With probability $1 - \mu$ the selected individual A will resort to imitation. In this case, a model agent B from the same population is selected. The fitness of both agents is calculated as the average payoff obtained in a large number of NM-MUG interactions (Algorithm 2).⁴ Imitation will occur with a probability that follows the Fermi function for pairwise comparison (Traulsen et al., 2006) such that

$$prob_{A \leftarrow B} = \frac{1}{(1 + e^{-\beta(f_B - f_A)})}$$

where f_A is the fitness of the imitator, f_B is the fitness of the model, and β is the so-called intensity of selection, controlling how dependent the imitation process is on agents’ fitness values. When imitation occurs, the value of p or q characterizing agent B will be adopted by agent A . The adopted strategies are subject to a small perturbation: we add a value between $-\epsilon$ and ϵ , sampled from a uniform probability distribution. We guarantee that $p_i, q_j \in [0, 1], \forall i, j$, truncating the adopted value if necessary.

⁴Note: while Proposers are drawn independently of their strategy, Responders are selected based on their q_i which is why we need to take this into consideration in Algorithm 2.

Algorithm 2: Sketch of fitness computation of individual A based on selection of M Proposers and N Responders.

```

Function  $fitness(A)$ 
   $accumulatedFitness = 0;$ 
  for  $i \leftarrow 1$  to  $Samples$  do
    if  $A \in Z_P$  then Select Proposers including  $A$ :
       $\lfloor$  Sample  $|N| - 1$  other Proposers
    else
       $\lfloor$  Sample  $|N|$  Proposers
    Select group of Responders  $M$  (for instance,
      ordering their  $q$  values, ascending, and picking the
      agents having the thresholds in the range  $m^{th}$  to
       $(m + |M| - 1)^{th}$ 
       $\bar{p} = \sum_{j \in N} p_j / |N|$ 
       $\bar{q} = \sum_{k \in M} q_k / |M|$ 
    if  $\bar{p} \geq \bar{q}$  then Proposal accepted:
      if  $A \in Z_R \wedge A \in M$  then Compute Responder
        payoff:
           $\lfloor$   $fitness \leftarrow U_A^R$  (using Equation 1)
        else Compute Proposer payoff:
           $\lfloor$   $fitness \leftarrow U_A^P$  (using Equation 2)
       $accumulatedFitness += fitness$ 
  return  $accumulatedFitness / Samples$ 

```

During the simulations, we record 1) the average strategy used in the population of Proposers and Responders, 2) the average acceptance rate of proposals, 3) the average fitness of Proposers and Responders and 4) the time-series of strategy adoption. We are particularly interested in understanding how strategy dynamics are impacted by different Responder selection rules (i.e., different values of m and M). We report these results next.

Experiments and Results

We simulate the NM-MUG as described previously (Algorithm 1) and present the results in the following. We average over 100 runs of 20,000 generations each, and we use 100 samples for each fitness computation (Algorithm 2). We set $Z_P = Z_R = 100$, $\mu = 0.001$, $\epsilon = 0.01$, and $\beta = 10$.

We are interested in the fairness of accepted deals which, for simplicity, we here define as (expected) payoff equality within and between the populations. For our scenarios, this means that fairness *between populations* is achieved when

$$\bar{p} = \frac{|M|}{|M| + |N|} \quad (3)$$

such that proposers and responders on average receive the same utility (by Equations 1 and 2). Fairness *within populations* is similarly achieved when all individuals share the same expected utility, which in our scenario means that $p_i = p_j, \forall i, j \in Z_P$ and $q_i = q_j, \forall i, j \in Z_R$. While

more elaborate measures of fairness are possible (de Jong and Tuyls, 2011), we leave these for future work and instead focus on the role of Responder selection in this paper.

Base Scenario

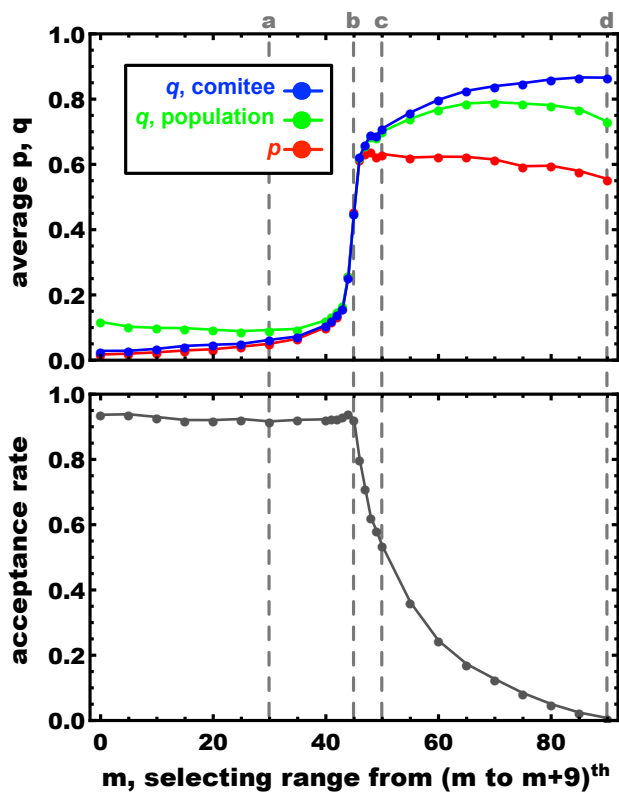
After simulating the co-evolving dynamics of agents playing the NM-MUG, and adapting their p and q strategies accordingly, we realized that the base scenario, where all accepting individuals are selected, nurtures long-term unfair divisions between Proposers and Responders. We verified that the p and q evolve, on average — taken over the whole population(s), over 20,000 generations and over 100 runs — to values close to 0.01 and 0.1, respectively. This results in a large proportion of rejected deals and, in case a deal is accepted, a highly unfair (between populations) outcome. The same result is obtained when selecting randomly composed groups of Responders with a fixed size M .

Responder Competition

We proceeded to test how competition for taking part in the Responder group affects these dynamics. We measured the average strategy usage, acceptance rate and fitness given a range of values for the rank parameter m , yielding both extreme (very low or very high m) and moderate ($m \approx Z_R/2$) Responder groups.

As Figure 2a conveys, increasing m increases the average values of p and q adopted by individuals in the long-run. Notwithstanding, selecting strict Responders — that have the highest values of q — is pernicious by leading to low acceptance rates, as evidenced by Figure 2a (bottom panel). Selecting groups that are characterized by the lowest values of q (low m) is disadvantageous for the Responders population as, over time, Proposers learn to offer extremely low proposals. Selecting groups formed by the highest values of q (high m) is equally harmful: due to the high fraction of proposals being rejected, individuals are unable to obtain high values of fitness. The optimal selection rule selects those representatives with a value of q close to the population median (i.e., $m \approx Z_R/2$), as evidenced in Figure 2b. Note that this figure presents the *relative average fitness* as a ratio with respect to the case $m = 0$. For example, when $m = 45$ a Responder receives on average approximately 32 times more payoff compared to $m = 0$. Conversely, in this case a Proposer earns slightly above 50% of what she would receive for $m = 0$. Using this relative scale allows an easier comparison with the base (unfair) scenario $m = 0$. As $|N| = |M|$ in this case, a between populations fair outcome is achieved whenever $\bar{p} \approx 0.5$.

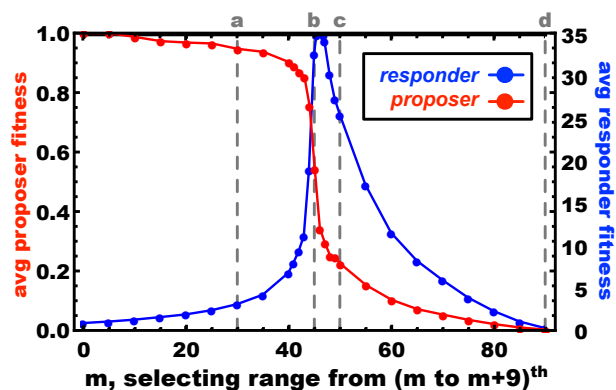
Changing m has a profound impact on the evolving dynamics of p and q , as shown in Figure 2c, where time-series corresponding to exemplifying runs for $m = 30$ (a), $m = 45$ (b), $m = 50$ (c), and $m = 90$ (d) are presented. We see a clear difference between situation (a), with high acceptance rates and low fairness, and (d), with almost arbitrary dynam-



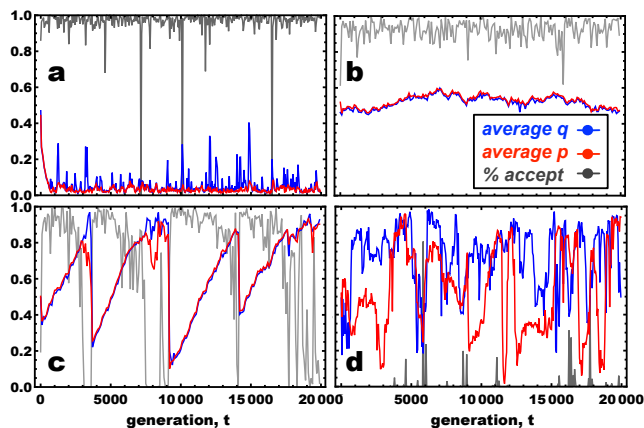
(a) The average p , population- q (taken over the whole population) and group- q (taken over the selected groups of Responders). On the bottom we show the average acceptance rate, which is representative of the utilitarian social welfare in our scenario. Responder competition causes both values of p and q to increase when m increases (top). However, too large values of m result in an increasing number of rejected proposals (bottom).

Figure 2: Responder competition: Proposals are made by random groups of Proposers with size $N = 10$ and the group of Responders (with size $M = 10$) is formed by the Responders with the m^{th} to the $(m + M - 1)^{\text{th}}$ highest values of q .

ics due to the low acceptance rate and thus indiscriminate fitnesses within the population. Situation (b), with (on average) median ranked Responder groups yields high acceptance rates while simultaneously maintaining fair (between population) proposals. Interestingly, when the choice of qs to form the Responder groups is dictated by $m = 50$ (c), a cycling dynamic is often observed, representing periods of fairness and unfairness that repeatedly succeed over time. We extend the analyses of the average strategy used over time by presenting, in Figure 3, the full distribution of proposals (left) and acceptance thresholds (right) within those populations for $m = 45$ (top) and $m = 50$ (bottom). We find that the distribution of strategies is kept close to the mean, thus suggesting high within-population levels of fairness. In addition, we confirm that the cycles observed in Figure 2c for $m = 50$ do not result from the populations



(b) The fitness of Responders is maximized when intermediate groups of Responders (i.e., with the median values of q) are selected to form Responders' group. We represent the relative average fitness, as a ratio taken over the base scenario $m = 0$.



(c) Example of time series for scenarios marked in panels (a) and (b).

being divided in groups with different strategies that grow and shrink alternately — instead all agents adopt a similar strategy throughout.

Effect of Proposer Group Size

Finally, we investigate the effect of increasing the Proposers' group size, N . As hypothesized, increasing N yields a stricter social dilemma for the Proposers, akin to a public goods game: individuals will maintain a low value of p , expecting to maximize their share while hoping that others propose an offer high enough to guarantee acceptance by the Responders. As observed in Figure 4, this dilemma is more pressing in larger Proposer groups, as the average p adopted decreases with N . We also plot the between-population fair proposal value (black dashed line). By increasing the Proposer group size, the minimum proposal required to ensure a fair offer is relaxed: for the same Responders' group size

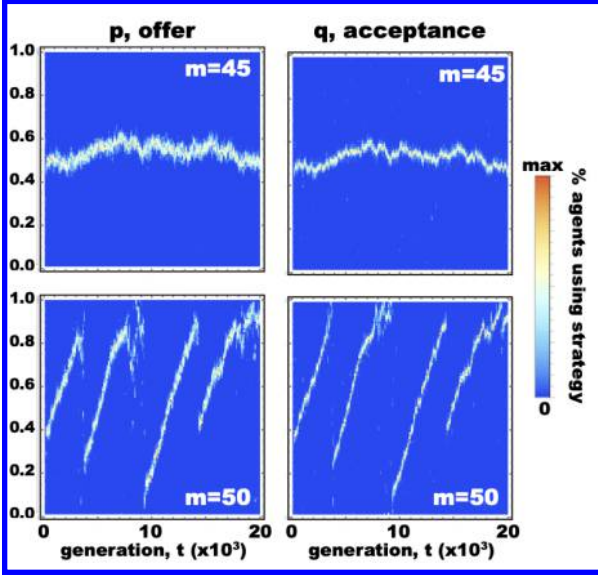


Figure 3: Distribution of strategies within the population for scenarios *b* (top) and *c* (bottom) highlighted in Figure 2. We find that agents adopt strategies that are close to the mean in each population, which implies high levels of within-population fairness.

(M), increasing N means that, individually, each Proposer is required to offer less, in order to maintain an even division between all Proposers and Responders involved in the transaction. In general, we find that fair proposals are easier to obtain when M is low and m is high. The average offer decreases whenever Proposers organize in larger groups.

Conclusion

In this paper we investigated the evolution of fairness within a new multiplayer version of the Ultimatum Game, the NM-MUG, in which a group of N Proposers offers a division of resources to a group of M Responders. Agent-based simulations showed that, in the absence of any Responder selection mechanism, the co-evolving populations of Proposers and Responders converge to a state in which proposals and acceptance thresholds are low, leading to unfair outcomes. This effect is more pronounced when the Proposers' groups are larger. We then investigated different Responder selection rules, contingent on their declared acceptance thresholds. We found that selecting extreme individuals is detrimental to the Responders' long-term payoff: selecting the lowest-demanding Responders incentives Proposers to do low offers, while selecting the highest-demanding Responders leads to many rejected offers. Moderate groups — i.e., selecting Responders with acceptance thresholds close to the population median — elicit the highest long-term gains for the Responders population as a whole, with high levels of fairness both between and within populations.

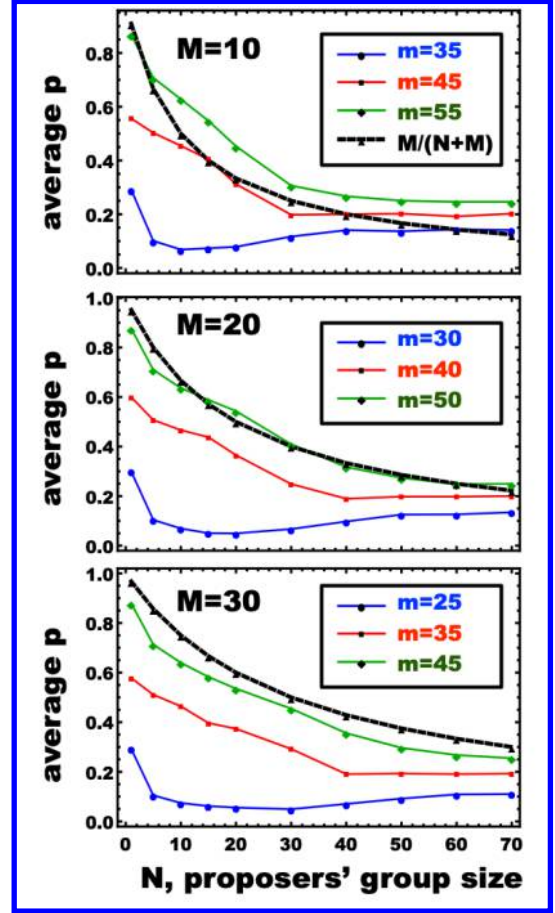


Figure 4: Effect of increasing Proposers' group size N on the average proposal level \bar{p} . We plot \bar{p} for Responders' group sizes $M = 10$ (top), $M = 20$ (middle), and $M = 30$ (bottom). Inside each panel, different Responders' selection rules (m) are plotted, moreover the black dashed line represents the between-populations fair proposal value (Equation 3), given the Proposers' and Responders' group sizes.

Although our model is based on several simplifying assumptions, due to its generality we believe that our results can nonetheless provide a practical message for institutional design (Bullock, 2016), both in the context of regional flexibility exchange between smart energy communities as well as in the broader context of group bargaining. The proposed model allows testing policies and emergent behaviors on the intersection between social choice theory, group bargaining, competition, and fairness elicitation.

We see many interesting avenues for further research based on our findings. A variety of selection rules can be envisioned and tested, for both the Responders as well as for Proposers. In addition, the utility functions used can be further tailored to specific real-world scenarios. Also, more elaborate measures of fairness can be investigated and could

even be incorporated into the utility function directly (as in e.g. de Jong and Tuyls 2011) or explicitly assuming traits such as empathy (Szolnoki et al., 2012), yielding potentially more complex and interesting dynamics. Finally, the effect of different selection rules can be analyzed in the context of spatially arranged individuals (Page et al., 2000; Szolnoki et al., 2012) or assuming iterated (multiplayer) ultimatum games (Ichinose and Sayama, 2014).

Acknowledgements

We are grateful to Michael Kaisers for helpful comments and feedback on earlier versions of this paper. This project has received funding in the framework of the joint programming initiative ERA-Net Smart Energy Systems' focus initiative Smart Grids Plus, with support from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646039. F.P.S. acknowledges support from the James S. McDonnell Foundation.

References

- Alexander, J. M. (2000). Artificial justice. In *Proceedings of Artificial Life VII*, pages 513–522.
- Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.
- Bloembergen, D., Hennes, D., McBurney, P., and Tuyls, K. (2015). Trading in markets with noisy information: An evolutionary analysis. *Connection Science*, 27(3):253–268.
- Bullock, S. (2016). Alife as a model discipline for policy-relevant simulation modelling: Might worse simulations fuel a better science-policy interface? In *Proc of the 2018 Conference on Artificial Life*, pages 28–29. MIT Press.
- Chakraborty, S., Baarslag, T., and Kaisers, M. (2018). Energy contract settlements through automated negotiation in residential cooperatives. In *2018 IEEE SmartGridComm*, pages 1–6.
- de Jong, S. and Tuyls, K. (2011). Human-inspired computational fairness. *Auton. Agents Multi Agent Syst*, 22(1):103–126.
- Fehr, E. and Schmidt, K. M. (1999). A theory of fairness, competition, and cooperation. *The Quarterly Journal of Economics*, 114(3):817–868.
- Gintis, H. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton University Press.
- Güth, W., Schmittberger, R., and Schwarze, B. (1982). An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior & Organization*, 3(4):367–388.
- Hagan, J. D., Everts, P. P., Fukui, H., and Stempel, J. D. (2001). Foreign policy by coalition: deadlock, compromise, and anarchy. *International Studies Review*, 3(2):169–216.
- Hofmann, L.-M., Chakraborty, N., and Sycara, K. (2011). The evolution of cooperation in self-interested agent societies: A critical study. In *Proc. of 10th Int. Conf. on AAMAS 2011*, pages 685–692.
- Ichinose, G. and Sayama, H. (2014). Evolution of fairness in the not quite ultimatum game. *Scientific Reports*, 4:5104.
- Jing, X. and Xie, J. (2011). Group buying: A new mechanism for selling through social interactions. *Management Science*, 57(8):1354–1372.
- Lezama, F., Soares, J., Hernandez-Leal, P., Kaisers, M., Pinto, T., and do Vale, Z. M. A. (2018). Local energy markets: paving the path towards fully transactive energy systems. *IEEE Transactions on Power Systems*.
- Osborne, M. J. et al. (2004). *An introduction to game theory*, volume 3. Oxford university press New York.
- Page, K. M., Nowak, M. A., and Sigmund, K. (2000). The spatial ultimatum game. *Proc. Royal Soc. B*, 267(1458):2177–2182.
- Phelps, S., Parsons, S., and McBurney, P. (2004). An evolutionary game-theoretic comparison of two double-auction market designs. In *International Workshop on Agent-Mediated Electronic Commerce*, pages 101–114. Springer.
- Rand, D. G. and Nowak, M. A. (2013). Human cooperation. *Trends in Cognitive Sciences*, 17(8):413.
- Ranjbar-Sahraei, B., Bou Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014). Evolution of cooperation in arbitrary complex networks. In *Proc. of the 13th Int. Conf. on Autonomous Agents and Multi-Agent Systems*, pages 677–684.
- Santos, F. P., Pacheco, J. M., Paiva, A., and Santos, F. C. (2019). Evolution of collective fairness in hybrid populations of humans and agents. In *Proceedings of AAAI'19*. AAAI Press.
- Santos, F. P., Santos, F. C., Melo, F. S., Paiva, A., and Pacheco, J. M. (2016). Dynamics of fairness in groups of autonomous learning agents. In Osman, N. and Sierra, C., editors, *Autonomous Agents and Multiagent Systems*, pages 107–126.
- Santos, F. P., Santos, F. C., and Pacheco, J. M. (2018). Social norm complexity and past reputations in the evolution of cooperation. *Nature*, 555(7695):242.
- Santos, F. P., Santos, F. C., Paiva, A., and Pacheco, J. M. (2015). Evolutionary dynamics of group fairness. *Journal of Theoretical Biology*, 378:96–102.
- Sigmund, K. (2010). *The calculus of selfishness*, volume 6. Princeton University Press.
- Skyrms, B. (2004). *The stag hunt and the evolution of social structure*. Cambridge University Press.
- Szolnoki, A., Perc, M., and Szabó, G. (2012). Defense mechanisms of empathetic players in the spatial ultimatum game. *Physical Review Letters*, 109(7):078701.
- Takesue, H., Ozawa, A., and Morikawa, S. (2017). Evolution of favoritism and group fairness in a co-evolving three-person ultimatum game. *EPL (Europhysics Letters)*, 118(4):48002.
- Taylor, P. D. and Jonker, L. B. (1978). Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1-2):145–156.
- Traulsen, A., Nowak, M. A., and Pacheco, J. M. (2006). Stochastic dynamics of invasion and fixation. *Physical Review E*, 74(1):011909.

Investigating the Origins of Cancer in the Intestinal Crypt with a Gene Network Agent Based Hybrid Model

Arturo Araujo^{1,4}, Hanxiao Zhang², Albert Rübber³ and Peter J Bentley^{1,4}

¹ Braintree Ltd, London, United Kingdom

² CoMPLEX, UCL, London, United Kingdom

³ Euregio Skin Cancer Center, Department of Dermatology, University Hospital RWTH, Aachen, Germany

⁴ Department of Computer Science, UCL, London, United Kingdom

a.araujo@braintree.com

Abstract

Colorectal cancer (CRC) is the second most common tumour in the world (Bray, 2018). It has been proposed that morbidity and mortality could be mitigated by screening methods that identify key genetic mutations in the DNA of a patient's biosample (Traverso, 2002). However, for this to work, a theoretical understanding of the most likely mutations that initiate malignant transformation, and how they affect subsequent microevolution, is needed. Specifically, we hypothesise that *there is a CRC-proliferative mutation that is more likely to be initially fixated in the crypt*. To investigate this, we developed an agent-based model of cells in the colon crypt that shows emergent biological homeostasis at the tissue level from the cellular and molecular interactions. We equipped each of the cells with a molecular gene network which, in their wildtype state, regulates homeostasis in the crypt and recapitulates known behaviour. We identified and modelled key genes implicated in CRC which, when mutated, alter the rate of death and division of cells. We used this model to study the biological first principles of the fixation of mutations, offering key spatial and temporal understanding of this process. We discuss the impact and clinical relevance of proliferative genetic mutations in isolation, pointing to the KRAS gene as a likely mutation to be initially fixed in the crypt.

Introduction

The development of colorectal cancer (CRC) is thought to occur in marked stages happening throughout an entire decade (Williams, 2016); however, the precise mechanisms of oncogenic initiation are still debated. Because of this slow development, CRC is a highly suitable system for investigating the emergence and accumulation genetic alternations that many cancers have in common. However, in vivo and in vitro models usually take place in a matter of days or weeks, not being able to fully recapitulate the microevolutionary oncogenic process. Current research points to CRC having an origin at the base of colonic crypts: flask shaped invaginations in the inner lining of the intestine, which produce new cells to support the tissue (Figure 1). Cells at the top of the crypt are continuously worn away by the process of nutrient absorption (through the villi) and passing food, and thus are being continually renovated by stem cells at the bottom. These stem cells divide to replace worn cells and may even displace other stem cells so that at a given time the whole crypt becomes monoclonal- a descendant of one single stem cell. Because of the high rate of division, it is here that key oncogenic mutations are thought to arise

(especially during cell division, which is the most vulnerable state for the DNA of the cell).

It is very difficult to investigate in vivo or in vitro the impact that key genetic mutations have in isolation. However, this gap in our understanding needs to be addressed to investigate how these initial mutations may shape the subsequent evolution of the disease, and thus offer some therapeutic insights. We propose to overcome the limitations of time, cost and tractability of wet-laboratory experiments with the aid of theoretical approaches, such as agent-based modelling and gene networks. The focus of this work is to determine whether *there is a CRC-proliferative mutation that is more likely to be initially fixated in the crypt*. To answer this question, we have abstracted the natural processes of crypt homeostasis and distilled current genetic knowledge into a network. With this realistic simulation, we can recreate the in vitro scenario that isolates the probability of key CRC implicated mutations to be fixated and thus become the steppingstone in oncogenic transformation.

Background

ALife has helped model emergent phenomena in human diseases such as the emergence and evolution of brain tumours (Swanson, 2003). Alife researchers have used the distilled essence of biological behaviour to develop nature-inspired computational techniques that will help us create better tools (Andrews, 2008). It is with this virtuous cycle of the development of ALife methods in mind that we seek to formalize the simulation of complex biosystems and apply these to the understanding of cancer (Araujo, 2010) (Rübber, 2013), with the ultimate goal of discovering novel therapeutic targets such as key switches in cell-cell communication (Bentley, 2014). The ALife methods that us and others have used, such as agent-based modelling, have allowed us to investigate the genetic mutations that occur at a molecular level, but which have repercussions at the tissue level; as well as the cross-talk between cell types that make up the tumour microenvironment (Araujo, 2014). Importantly, these modelling approaches can connect the different time and space scales are needed (Rejniak, 2010), and have shown the feasibility of modelling cells with internal genomes to study cancer initiation (Fontana, 2010).

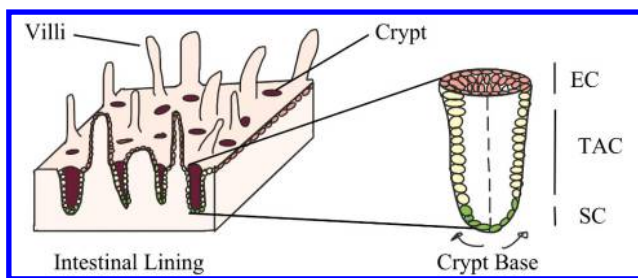


Figure 1. An intestinal crypt is an invagination in the inner lining of the intestine. Stem cells at the base constantly replenish the tissue.

Modelling Colorectal Cancer

There have been advances in the computational study of CRC, such as the effects of variation of cell cycle rate in that may disrupt homeostasis in colon crypts (Smallbone, 2013). Bravo and Axelrod measured the variation in stem cells, proliferating cells, and differentiated cells in multiple crypts in normal human biopsy specimens, offering a metric of the robustness with which crypts recover from chemotherapy and radiation scheduling protocols (Bravo, 2013). Meineke used a lattice-free cylindrical surface to model experimental data showing that cell movement is a consequence of mitotic activity (Meineke, 2001). Venturing into molecular modelling, Leeuwen developed a hybrid model that incorporates the WNT ligand gradient along the crypt axis that is shown to regulate the cell cycle and cell division (Van Leeuwen, 2009), being one of the first truly multiple-scale abstractions that link the subcellular, cellular and tissue level processes.

Agent-Based Modelling (ABM) theory has been further developed to study subcellular behaviour. Using such advances, Mirams was able to show theoretically that the probability of mutations fixation is only weakly associated with the destruction of WNT-dependent cell proliferation (Mirams, 2012). Introducing space into the already complex dynamics, Buske developed one of the first 3D crypt models to study combined effect of WNT and Notch signalling on cell proliferative behaviour (Buske, 2011). There are other ABMs, such as those proposed by (Dunn, 2013) and more recently by (Ingham-Dempster, 2017), that abstract the concept of anoikis (programmed cell death) for a systemic investigation of emergence behaviours and migration dynamics that are difficult to study in vitro. The researchers were able to localize cell death to a small region at the top of the crypt and identifying it as an emergent property in response to changes in cell proliferation rates.

Mutations in the Intestine Crypt Evolution

Recent work in genetics and molecular biology has helped identify key genetic alternations implicated in the tumorigenesis of CRC. These genes have been generally classified as either tumour-suppressor genes (TSGs), due to their anti-growth properties; or proto-oncogenes (POG), which drive proliferation and will be the focus of our investigation. CRC mutations inactivate TSG function, or increase POG function (the mutated version termed oncogene). Typically, mutations are recessive in TSGs but dominant in proto-oncogenes; this means both alleles need to be altered in TSGs

for loss-of-function to occur, whereas mutation in one of the copies of proto-oncogenes is enough to promote proliferative behaviour, and thus have a more measurable effect (Evan 2006). It has been calculated that in normal human cells, the average mutation rate is $\sim 2.5 \times 10^{-8}$ mutations/nucleotides (Nachman, 2000). Mutations can manifest as point mutations altering one specific gene or through small structural aberrations such as short gene duplications, deletions or inversions. Furthermore, cancers may demonstrate chromosomal instability (CIN), where defects during cell division leads to daughter cells with large chromosomal amplifications, deletions or whole chromosome rearrangements (aneuploidy). CIN that lead to changes in chromosome number or structure accounts for 85% of sporadic colorectal cancers (Tsang, 2014). These mechanisms may lead to the loss of a TSG or to a gain in POGs such as KRAS.

Although much work has been done to elucidate the genetic signalling pathways, the connections between all the genes involved in CRC oncogenesis is still not fully understood. The current evidence points to the existence of some key mutations that are consistent in CRC. Amplifications and mutations of POGs KRAS and BRAF have been consistently observed in CRC; especially in CIN tumours (Pino and Chung 2010). While some mutations have been studied both biologically and clinically, the knowledge on the role they have when combined with mutations in other OGs or TSGs is still limited. It is therefore of importance to determine the chances that different mutations have to get fixated in the crypt, as these would then synergise with subsequent mutations in the path to malignant transformation.

We hypothesise that *there is a CRC-proliferative mutation that is more likely to be initially fixated in the crypt*. To provide evidence for this, we will simulate key mutations in an agent-based model that can help us understand 1. how long does it take for a random mutation to be fixed in the crypt? and 2. how fast would a key mutation fixate in the crypt? We will perform multiple simulations on CIN conditions (such as copy number decreased to 1 or 0 or copy number increased to 3) and point mutation in each POG abstraction.

The System

In order to investigate the role that the individual mutations have in CRC initiation, we have designed a computational model that exhibits the same homeostatic behaviour of a healthy crypt. We have abstracted the behaviour at a cellular level and modelled each cell as a circular agent. As described in Figure 1, the colon crypt is an invagination that is in constant renewal. For representation, we have adapted the invagination into a two-dimensional plane made up of cells with a continuous boundary to the left and right of the cells, thus preserving the original three-dimensional geometry (Figure 2). In our computational model, cells are represented by agents that react to the morphogens in the microenvironment, enabling cells to divide, quiesce or die (Figure 3). During normal homeostasis, three populations of morphogen-regulated cells coexist: Stem Cells (SCs) at the bottom, Transit Amplifying Cells (TACs) in the middle and fully differentiated Epithelial Cells (ECs) at the top (Figure 2). Cells are physically able to push other cells in all directions, with higher probability of

pushing cells up or sideways up, a low probability of dividing sideways and an even lower probability of going downwards to the sides or downwards (Table 1). SCs proliferate at the bottom compartment, pushing cells up and supplying a fresh batch of TACs that eventually differentiate at the top of the crypt and are shed away.

We previously studied the intestinal glands in the colon, or colon crypts: invaginations in intestinal tissue that help absorb nutrients as food passes through them (Figure 1). We found that ALife techniques are ideal to address biological complex systems at the molecular, cellular and tissue level; and capable of shedding light on in vivo experiments that report seemingly different findings. Specifically, we were able to bridge different values reported for these contributors, and thus reconcile theories on which one is the biggest contributor the time it takes for a crypt to become the descendant of a single basal stem cell, also known as monoclonality (Araujo, 2018). In our work we focused on two key morphogens the process of cell renewal in the crypt: WNT (promoting the stem-cell phenotype) and EGF (promoting cell division and regulating cell differentiation). We modelled both morphogens as being maximum at the base of the crypt, as they are thought to be provided by Paneth cells which reside there (Sato, 2011). It is currently thought that the morphogens concentrations decrease in a gradient throughout the length of the crypt (Bach, 2000). In our homeostatic model, the WNT ligand concentration, keeping all the cells in contact with it in a stem cell phenotype, is completely depleted 10 to 30 μm (approximately one to three cell diameters) above the base. When cells are out of the WNT concentration, but still within the EGF gradient, they lose stem cell properties, start aging, and are able to divide proportionally to the bio-availability of EGF. Once these transit-amplifying cells are pushed outside of the EGF gradient (approximately 31 μm above the base of the crypt) they become fully differentiated epithelial cells, stop dividing and their likelihood of being shed away or dying is 100%. Besides EGF, other morphogens have been implicated in the regulation of TAC cells (Carulli, 2014) and EGF in the model is only representative of a putative morphogen acting by a gradient.

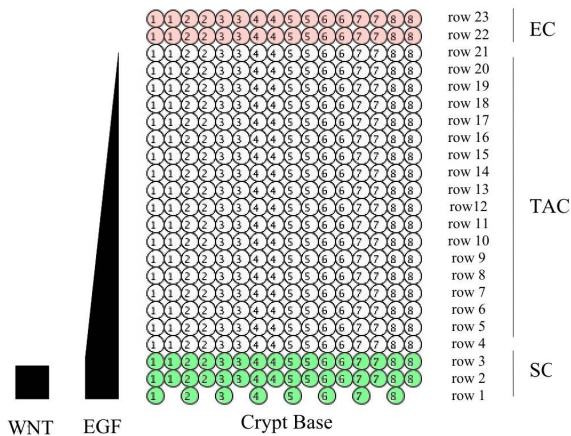


Figure 2. The compartments that make up the colon crypt model. The WNT morphogen (right) maintains cells in SC state, while EGF regulates cell division.

Our previous results (Araujo, 2018) show that the geometry of the crypt such as the total number of stem cells (Fletcher, 2012), the proportion of side cell displacement (Ritsma, 2015) and the number of basal stem cells (Kozar, 2013) all have a profound impact on the time to monoclonality. We showed that a niche of stem cells dividing from the bottom provide a continuous influx of new cells, and that eventually the dynamics (such as cell displacement), generate observable global effects such as a change in the frequency of monoclonality. This is important because it offers a metric for the possible fixation of mutations; but it doesn't consider the phenotype changes that an altered genotype might confer. To obtain a true metric by which genetic alterations will spread through the crypt it is important to include realistic abstractions of oncogenic mutated cells in the intestine.

The Algorithm

We had previously developed an ABM that recapitulates the known mechanics of the healthy crypt (Araujo, 2018). Novel in this work, we have completely redesigned our agents, giving them an internal genetic circuit for decision making. As in the original model, every cell is queried at every time step in an asynchronous update. A random cell that has not been previously updated during the time step is picked and follows the update algorithm by which it is given the chance to decide whether to die, divide or do nothing per cell cycle (Figure 3).

The cell identity is decided by its position with respect to the WNT concentration (SC) and the EGF gradient (TAC) though a probabilistic calculation. SC cells have a fixed rate of division, $divisionSC$, while under the influence of WNT, which might be altered by a mutation. Their probability of division 100% throughout the wildtype SC compartment is then calculated as:

$$P(SCdiv) = 100 * (divisionSC)$$

Wildtype TACs have 100% chance of dividing as they leave the WNT concentration but are still in contact with the EGF gradient. The wildtype TACs chance of division, $TACdiv$, decreases linearly to 0% as it travels to the top of the EGF gradient. To achieve this, we normalize the difference between the EGF and WNT, $\Delta grad$, and calculate the vertical distance, $yPos$, between the TAC cell and the end of the WNT ligand concentration, which modulates $divisionTAC$:

$$P(TACdiv) = 100 * (divisionTAC) * (yPos - \Delta grad) / \Delta grad$$

Where $divisionSC$ and $divisionTAC$ are biological parameters for wildtype cells, as shown in (Table 1). When a cell divides it pushes one of its neighbours, selected with the baseline probabilities shown in Table 1. The probability of death is 100% when leaving the EGF gradient. We model aging as a decrease in telomere length, therefore reducing the number of times a cell can divide (initially 5 divisions) and persist within the TAC-compartment. Each round, the algorithm queries the system for a user defined condition. For this work we will use one of three user-specific queries: 1. a set time (in days), 2. if the basal SCs have become monoclonal, and 3. if a mutation has disappeared from the basal SC compartment. If the user specific query is not met, it continues updating cells, and advancing to a new time step when every cell has been updated or stopping when the query has been met.

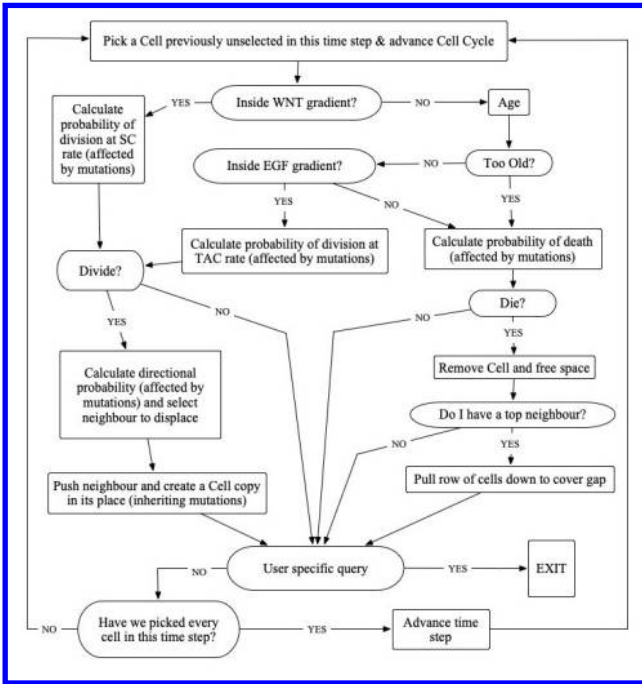


Figure 3. Algorithm for cell dynamics. The program ends when the user specific query (E.g. reaching monoclonality) is met.

Agent design

We designed the agents to offer different information that could be displayed visually with regards to key metrics of interest. In the wildtype visualization, SCs (cells inside the WNT ligand concentration) are tagged green, TACs (cells outside the WNT and in the EGF gradient) are marked white, and ECs (cells outside of both gradients) are coloured peach (Figure 2). To track the lineage of cells, we give a tag number to each of the initial basal SC which is inherited by their progeny throughout the simulation. In the initial baseline condition, as shown in Figure 2, we assume that cells that are in the column immediately on top of the SC are progeny of it and therefore inherit this number, which is displayed at the centre of each cell in the visualization. This gives a representation and a clear pathway of how cell mixing and eventually monoclonality occur. Other metrics such as the number of divisions and age are also stored in each agent and can be shown as the number displayed on each cell.

Biological Parameters

The structure of the seven gene network used in this paper was extracted from and confirmed against literature (De Roock, 2011) (Sartore-Bianchi, 2016) (Strubberg, 2017) (Pan, 2017) (Berg, 2012). Inputs (WNT and EGF levels at crypt positions), outputs (calculated cell properties) and trigger values (threshold levels for different outputs) of the gene network algorithm in the baseline case (no mutations), were selected based on the parameters from Table 1, while mutation impacts are shown in Figure 3. The gene output for copy number changes, gene inactivation/activation mutations, signal thresholds for output changes were selected to reproduce the expected mutation impact outputs (Figure 4 and Table 2).

The gene signalling pathway diagram was converted into a simple algorithm (Figure 4) which returns cell parameters for proliferation probability, death probability, cell cycle length and cell fate (stem cell, TAC or differentiated cell) based on the position of the cell in the crypt. Also included in the algorithm are abstracted impacts for activating or deactivating point mutations and copy number changes e.g. if KRAS was activated proliferation probability was increased. Each stage of the algorithm represents a gene receiving a signal of a certain strength either from the previous gene in the network or from the initial signalling protein (EGF or WNT).

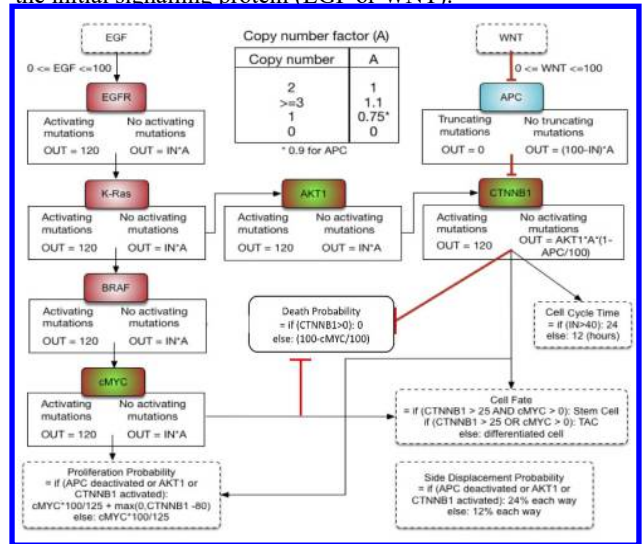


Figure 4. Simplified gene network. The network was constructed based on the mutation frequency of key CRC related genes documented in the COSMIC database (Forbes, 2017; cancer.sanger.ac.uk). Black arrows with solid line: gene activation; Black arrows with dash line: effects; Red arrows: inhibition; red boxes: proto-oncogenes; blue box: tumour suppressors. The input and output variables as well as the calculation for signal integration are a first approximation to reconcile in-vivo (Ritsma, 2015) and in-vitro (Snippert, 2013) data with yet-unquantified but known genetics (Forbes, 2017).

Parameters	Value	Reference
Cell Diameter	10 μm	(Bach, 2000)
Width and length of crypt	16 x 25 cells	(Bach, 2000; Totafurno, 1987)
SC division rate (<i>divisionSC</i>)	Once every 24 hours max	(Fischer, 2016)
TAC division rate (<i>divisionTAC</i>)	Once every 12-24 hours max	(Kozar, 2013)
Probability of Side Displacement per cell division (<i>SDisp</i>)	0.24 (0.12 each side) to 0.74	(Ritsma, 2015) (Snippert, 2013)
Probability of Downwards Displacement per cell division	0.08 for TACs, 0.00 for SCs	(Ritsma, 2015)

Table 1: Baseline parameters used to simulate a murine small intestine crypt.

Genetic Alteration	Change in Probability
Gene copy number increase to 3 or more	output gene signal > 100%
Gene copy number decrease to 1	output gene signal < 100% > 0%
Gene copy number decrease to 0	output gene signal = 0%
Activating mutation	output gene signal > 100% regardless of input
Minimum cell cycle length	12 hours (normally applies to TAC)
Maximum cell cycle length	24 hours (normally applies to SC)

Table 2- Altered probabilities from mutations in the abstracted gene network. The actual values of the variables as well as the calculations describing signal transduction are still being elucidated, and thus we have abstracted these values in a first approximation to recapitulate published data.

Model Validation 1: Positional Mitotic Index

We compare our model with a metric used by us and others in the past: mitotic activity, which causes a pressure-driven passive movement from the bottom to the top of the crypt, as described in (Meineke, 2001). We ran 100 simulations and defined our end user-query as a 30-day period. When tested, the average division per row (Figure 5) recapitulates the data for mitotic index distribution in the crypt presented in Fig 4.iv of (Sunter, 1979), where there is a maximum of mitotic activity a few rows after leaving the base of the crypt and decreases throughout the rest of the rows.

Model Validation 2: Neutral Drift Dynamics

We test for evidence of a neutral drift in the cellular dynamics of intestinal stem cells, and compare to results found by (Snippert, 2010). In their experimental system, a 14 basal stem cell population was calculated to have a probability of side displacement of 0.74, demonstrating neutral competition amongst SCs. In 100 simulations of such experiment (Figure 6) we find found that the dynamics exhibited by the updated model agree with that reported in Figure 7D of (Snippert, 2010).

Experiments

We hypothesise that *there is a CRC-proliferative mutation that is more likely to be initially fixated in the crypt*. To provide evidence for this, we focus on answering: 1. how long does it take for a random mutation to be fixed in the crypt? and 2. how fast would a key POG mutation fixate in the crypt? To investigate the time to mutation fixation and quantify the role that the individual mutations have in CRC initiation, we first establish a baseline to fixation. Subsequently we simulate experiments that explore mutational scenarios. Our experiments are:

Experiment 1- Baseline time to monoclonality. In their cell-tracking experiments Ritsma et al. measured the time it takes for one basal SC sub-clone to divide sideways enough to make the entirety of the crypt a descendant of this cell, or monoclonal

(Ritsma, 2015). The researchers performed this experiment with a number of crypts and describe in their results the percentage of crypts that have become monoclonal in a 140-day period. The researchers use 8 basal SC and 16 suprabasal SCs, each with a probability of side displacement of 0.24% (0.12 each side) per division. We perform 100 simulations with the baseline model, as shown in Figure 7, based on the accepted parameters shown in Table 1 and Table 2, and measure our results using this same methodology.

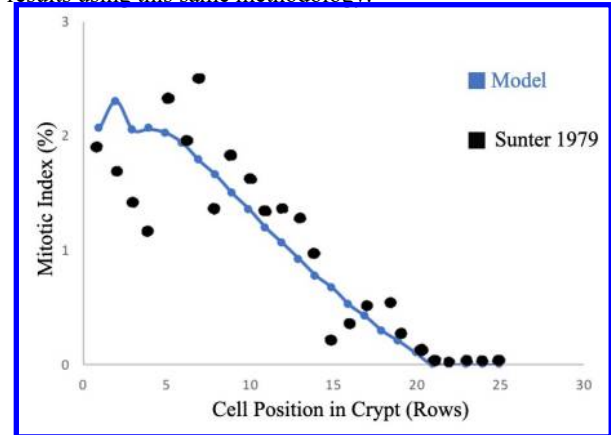


Figure 5. Mitotic index distribution in the crypt (upwards from the crypt as described in in Figure 2) of 100 simulated crypts with 8 basal and 16 suprabasal stem cells over a 30-day period. We compare it to data presented in Fig 4.iv of (Sunter, 1979).

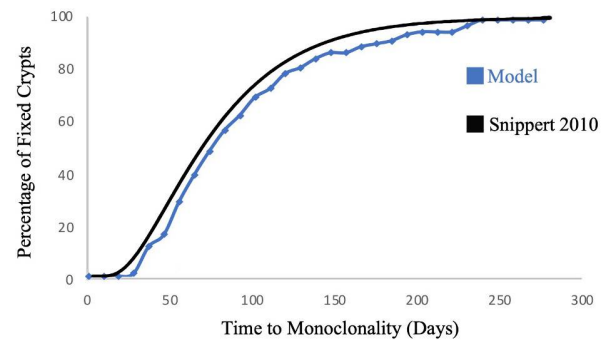


Figure 6. Frequency of monoclonal crypts over time as a percentage of surviving SC clones (out of initial 14) as predicted by neutral drift dynamics (Snippert, 2010).

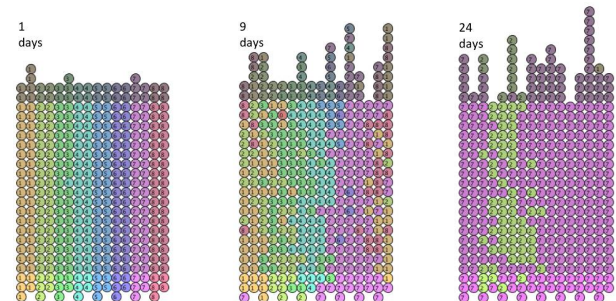


Figure 7. Progression from 8 stem cell lineages towards monoclonality of stem cell 7 (marked purple) over 24 days.

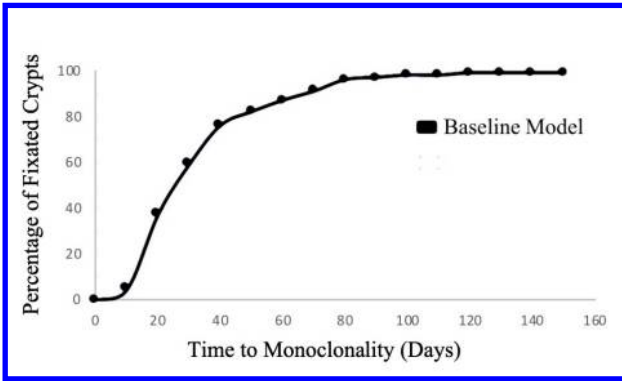


Figure 8: Simulation of biological data for monoclonality based on experiments reported on (Ritsma, 2015). From 100 simulations (as shown in Figure 7) we measure every 10 days the percentage of simulations that have become monoclonal. By day 140, 100% have reached this state.

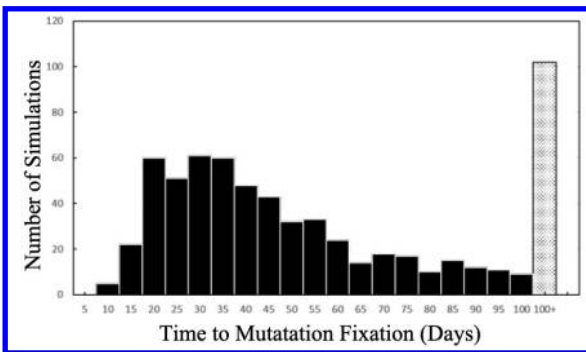


Figure 9. Time to a mutation fixation within 100 days (black boxes) from 4000 simulations. The grey box shows the number simulations that still contained a mutant at the basal SC compartment that could spread throughout the crypt given more simulated time.

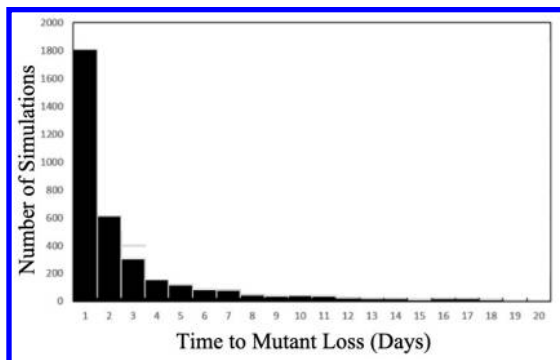


Figure 10. Time to a mutant loss from the crypt from 4000 simulations.

Experiment 2- Time to Mutation Fixation. In Experiment 2, a genetic mutation is randomly introduced into one of the basal stem cells (BSC). Mutations that lead to changes in the output signal of the gene in Figure 4 will ultimately alter proliferation probability, cell death probability and cell cycle length.

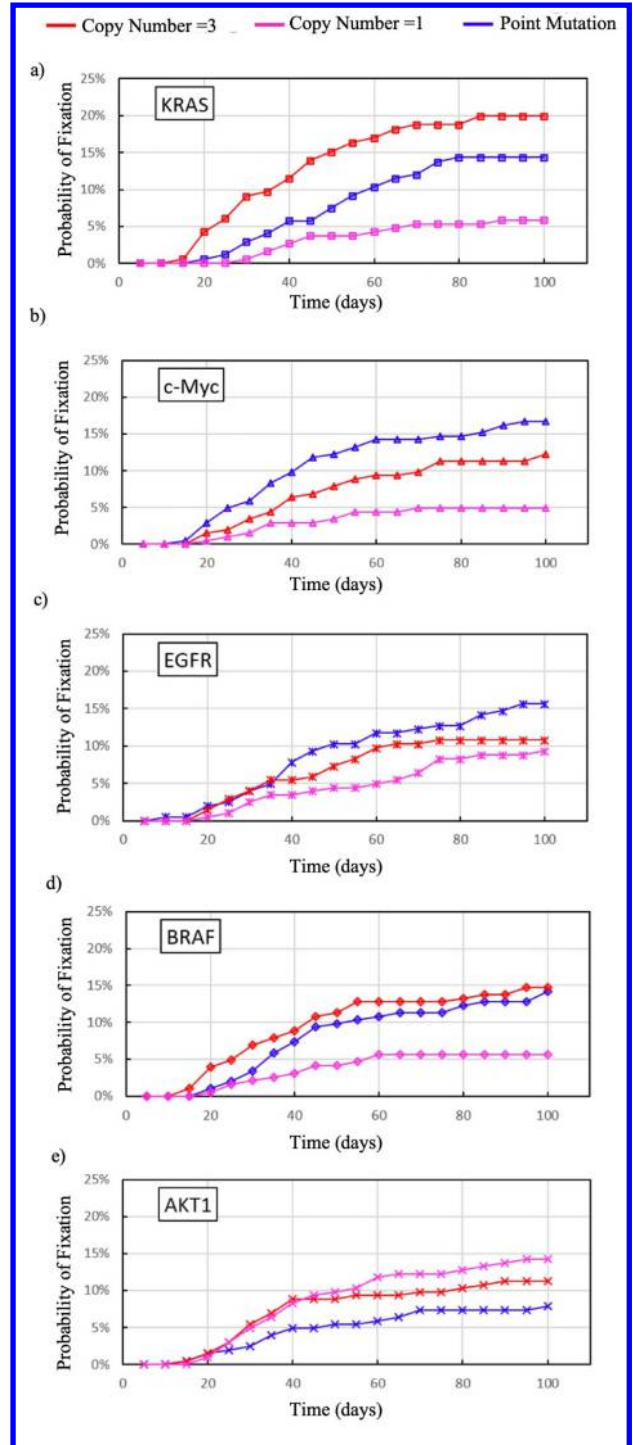


Figure 11. Probability of fixation for each genetic mutation as a cumulative distribution function for the oncogenes KRAS, c-Myc, EGFR, BRAF and AKT1.

The crypt is allowed to evolve for 100 days and the simulation is stopped when one of the three conditions is satisfied: 1. The time limit is reached. 2. Mutation is fixed. or 3. The mutation population is swept from the crypt. The

mutation is assumed to be fixed when the mutated lineage takes over all BSCs, as eventually, all cells in the crypt will have the same ancestor and inherit the same type of mutation. Similarly, the mutation is assumed dead when mutated ancestors is lost from the first row.

Experiment 3- Fixation Probability of Oncogenes. Finally, we simulate a point mutation and compare it to the gain of one gene and the loss of one alleles of the proto-oncogenes *EGFR*, *KRAS*, *BRAF*, *c-Myc*, and *AKT1*, with the altered probabilities shown in Figure 4. 200 simulations were run for each genetic mutation with basal stem cell number set to 8.

Results

Results 1: Establishing a baseline time to monoclonality. In our baseline simulations for monoclonality (Figure 8), 100% of the crypts become monoclonal, or fixed, by day 140 following the same trend as that reported by Ritsma (Ritsma, 2015). These dynamics are important to our understanding of the genetic evolution of the crypt, since they would give us an estimate of how fast we can expect a neutral mutation to spread through a healthy crypt. Using this as our baseline, we will proceed to analyse the results of Experiments 2, 3 and 4.

Results 2: Time to Mutation Fixation. Results from Experiment 2 show a distribution of the time it took for a mutation to become fixed in the crypt. The data collected from 4000 simulations ran for 100 days of simulated time shows that, if a mutation arises, it is mostly likely to be fixed within 20 to 40 days (Figure 9). Interestingly, 102 simulations contained at least one mutant cell which could dominate the crypt given more simulated time.

Figure 10 summaries the times at which the mutated ancestor was washed out from the crypt, showing that the survival rate for a mutation is low. Most mutant cells were swept from the crypt within the first 2 days. The results suggest that mutant domination is a slow and inefficient process, as most mutant cells are replaced by WT clone in a stochastic manner (Figure 10). This highlights the importance of monoclonal conversion in cancer initiation as it prevents the accumulation of mutations.

Results 3: Fixation Probability of Oncogenes. Figure 11 shows the probability of fixation for known proto-oncogenes under the scenario of a point mutation (blue), an increase in copy number (red) and loss of a gene (pink). The more likely scenario for true oncogenes activation is an activating mutation or an increase in copy number via CIN. Broadly, the cumulative distribution function for the oncogenes *KRAS*, *c-Myc*, *EGFR*, *BRAF* and *AKT1* under our theoretical conditions (Figure 4) is under 20% by day 100. Specifically, *AKT1* showed an increase in fixation when losing copy numbers (Figure 11.e). Some oncogenes have also been implicated in key cell maintenance processes, so having more than one role makes their precise contribution difficult to predict. Models such as this have helped us detangle the isolated effect of single mutations and suggest a metric for their contribution in oncogenesis. In the case of *c-Myc* (Figure 11.b) and *EGFR* (Figure 11.c), the model suggests a higher frequency of fixation via point mutations,

while gene amplification leads to the highest fixation of *BRAF* (Figure 11.d) and *KRAS* (Figure 11.a). The model points to *KRAS* having the highest likelihood of being fixated. These distributions, although in general agreement with reported behaviour in literature, do not by themselves explain a mutation fixation reported to be more than 25% for *APC* and possibly higher for *KRAS* (Vermeulen, 2013). Further, recent studies show that more than 97% of *KRAS* mutations in CRCs could be due to point mutations, conferring perhaps more plasticity (Bronte, 2015). Our results suggest that the precise mechanisms of oncogenic alterations for the fixation of mutations are not yet fully understood. Collectively, these results suggest that the theoretical impact of individual initiating mutations must be refined to fully capture the complexity joint mutations, which will be the next step in our research.

Conclusions

In this work, we investigated whether *there is a kind of CRC-proliferative mutation that is more likely to be initially fixated in the crypt*. We focused on modelling the *EGFR* and *WNT* genes and their connectivity, with special interest on *CRC* implicated proto-oncogenes. The evidence provided by the abstracted model suggests that *KRAS* could be heavily responsible for cancer initiation. In addition, in agreement with biological observations, oncogenes generally have higher fixation probability for activating mutations and lower probability for non-activating mutations. These results align with clinical evidence. Clinical studies for the rest of genetic mutations in *CRC* initiation are limited. In particular, the mechanism underlying aberrant activation of *c-Myc* is unknown. A major reason for this is that majority of these genetic mutations may occur in later stages of cancer progression or presented in an alternative pathway to *CRC*, thus further evaluation on the genetic network is required. The simplified model was able to provide a theoretical insight into the nature and significance of each genetic mutation in early tumour formation, and we aim for it to help both biologist and computational cancer researchers to interesting areas of exploration.

For our next step, we aim to further investigate the genetic and epigenetic interdependencies in *CRC* initiation, focusing on *in silico* experiments that cannot be done *in vitro* or *in vivo*, but which may have a significant impact on colorectal adenoma (Lao, 2011). The current model already allows shared information across multiple scales, so it can be readily extended to consider a multiple sequence of mutations. Also, epigenetic events (i.e. DNA methylations) that affect the regulation of key genes could be added to the molecular level. Quantifying the impact from both genetic and epigenetic abnormalities can help us shed light on non-intuitive mechanisms underlying cancer initiation. Finally, if information such as patient characteristics and carcinogen influence is available, the model could yield clinically relevant results.

The crypt model aims to investigate individual contribution of key genetic mutation in early stage of colorectal cancer. Our simulated results are in general agreement with evidence from the literature, and we believe that the next step is to further extend this bridge to clinically-relevant human data for therapeutic discoveries. Much work remains to be done to fully understand the *CRC* pathogenesis.

Acknowledgments

We would like to thank Braintree Ltd and UCL CoMPLEX for funding, fruitful discussions, and facilitating this collaborative and interdisciplinary research.

References

- Andrews, Paul S. . 2008. "Investigating Patterns for the Process-Oriented Modelling and Simulation of Space in Complex Systems." ALIFE.
- Araujo, A, L M. Cook, C C. Lynch, and D Basanta. 2014. "An Integrated Computational Model of the Bone Microenvironment in Bone-Metastatic Prostate Cancer." *Cancer Research* 74(9):2391–2401.
- Araujo, Arturo, A Rubben, David Basanta, and Peter J. Bentley. 2018. "Testing Three Hypotheses of the Contribution of Geometry and Migration Dynamics to Intestine Crypt Evolution." Pp. 1–8 in.
- Araujo, Arturo, Peter J. Bentley, and Buzz Baum. 2010. "Modelling the Role of Aneuploidy in Tumour Evolution." ALIFE.
- Bach, S P., A G. Renahan, and C S. Potten. 2000. "Stem Cells: the Intestinal Stem Cell as a Paradigm." *Carcinogenesis* 21(3):469–76.
- Bentley, Katie, Andrew Philippides, and Erzsébet R. Regan. 2014. "Do Endothelial Cells Dream of Eclectic Shape?." *Developmental Cell* 29(2):146–58.
- Berg, Marianne and Kjetil Soreide. 2012. "EGFR and Downstream Genetic Alterations in KRAS/BRAF and PI3K/AKT Pathways in Colorectal Cancer: Implications for Targeted Therapy.." *Discovery medicine* 14(76):207–14.
- Bravo, Rafael and David E. Axelrod. 2013. "A Calibrated Agent-Based Computer Model of Stochastic Cell Dynamics in Normal Human Colon Crypts Useful for in Silico Experiments.." *Theoretical Biology and Medical Modelling* 10(1):66.
- Bray, Freddie . 2018. "Global Cancer Statistics 2018: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries." *CA A Cancer Journal for Clinicians* 68(6):394–424.
- Bronte, Giuseppe . 2015. "New Findings on Primary and Acquired Resistance to Anti-EGFR Therapy in Metastatic Colorectal Cancer: Do All Roads Lead to RAS?." *Oncotarget* 6(28):24780–96.
- Buske, Peter . 2011. "A Comprehensive Model of the Spatio-Temporal Stem Cell and Tissue Organisation in the Intestinal Crypt" edited by D. Lauffenburger. *PLoS Computational Biology* 7(1):e1001045–14.
- Carulli, Alexis J., Linda C. Samuelson, and Santiago Schnell. 2014. "Unraveling Intestinal Stem Cell Behavior with Models of Crypt Dynamics." *Integrative Biology* 6(3):243–16.
- De Roock MD, Wendy, Veerle De Vriendt MSc, Nicola N. MD, Prof F. C. MD, and Prof S. T. MD. 2011. "KRAS, BRAF, PIK3CA, and PTEN Mutations: Implications for Targeted Therapies in Metastatic Colorectal Cancer." *Lancet Oncology* 12(6):594–603.
- Dunn, Sara-Jane, Inke S. Näthke, and James M. Osborne. 2013. "Computational Models Reveal a Passive Mechanism for Cell Migration in the Crypt" edited by J. Vera. *PLoS ONE* 8(11):e80516–18.
- Evan, Gerard I. 2006. "Can't Kick That Oncogene Habit." *10(5):345–47.*
- Fischer, Jared M. . 2016. "Single Cell Lineage Tracing Reveals a Role for TgfβR2 in Intestinal Stem Cell Dynamics and Differentiation." *Proceedings of the National Academy of Sciences of the United States of America* 113(43):12192–97.
- Fletcher, Alexander G., Christopher J. W. Beward, and S J. Chapman. 2012. "Mathematical Modeling of Monoclonal Conversion in the Colonic Crypt." *Journal Of Theoretical Biology* 300(c):118–33.
- Fontana, Alessandro. 2010. "An Artificial Life Model for Carcinogenesis.." ALIFE.
- Forbes, Simon A. . 2017. "COSMIC: Somatic Cancer Genetics at High-Resolution." *Nucleic Acids Research* 45(D1):D777–83.
- Ingham-Dempster, Tim, Dawn C. Walker, and Bernard M. Corfe. 2017. "An Agent-Based Model of Anoikis in the Colon Crypt Displays Novel Emergent Behaviour Consistent with Biological Observations." *Royal Society Open Science* 4(4):160858–13.
- Kozar, Sarah . 2013. "Continuous Clonal Labeling Reveals Small Numbers of Functional Stem Cells in Intestinal Crypts and Adenomas." *Stem Cell* 13(5):626–33.
- Lao, Victoria V. and William M. Grady. 2011. "Epigenetics and Colorectal Cancer." *Nature Reviews Gastroenterology & Hepatology* 8(12):686–700.
- Meineke, F A., C S. Potten, and M Loeffler. 2001. "Cell Migration and Organization in the Intestinal Crypt Using a Lattice-Free Model.." *Cell Proliferation* 34(4):253–66.
- Mirams, Gary R., Alexander G. Fletcher, Philip K. Maini, and Helen M. Byrne. 2012. "A Theoretical Investigation of the Effect of Proliferation and Adhesion on Monoclonal Conversion in the Colonic Crypt." *Journal Of Theoretical Biology* 312(C):143–56.
- Nachman, M W. and S L. Crowell. 2000. "Estimate of the Mutation Rate Per Nucleotide in Humans.." *Genetics* 156(1):297–304.
- Pan, Tianhui, Jinghong Xu, and Yongliang Zhu. 2017. "Self-Renewal Molecular Mechanisms of Colorectal Cancer Stem Cells.." *International Journal of Molecular Medicine* 39(1):9–20.
- Pino, M S. and D C. Chung. 2010. "The Chromosomal Instability Pathway in Colon Cancer." *Gastroenterology* 138(6):2059–72.
- Rejniak, Katarzyna A. and Alexander R. A. Anderson. 2010. "Hybrid Models of Tumor Growth." *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* 3(1):115–25.
- Ritsma, Laila . 2015. "Intestinal Crypt Homeostasis Revealed at Single-Cell Level by in Vivo Live Imaging." *Nature* 507(7492):362–65.
- Rubben, A and O Nordhoff. 2013. "A Systems Approach Defining Constraints of the Genome Architecture on Lineage Selection and Evolvability During Somatic Cancer Evolution." *Biology Open* 2(1):49–62.
- Sartore-Bianchi, A, F Loupakis, G Argilés, and G W. Prager. 2016. "Challenging Chemoresistant Metastatic Colorectal Cancer: Therapeutic Strategies From the Clinic and From the Laboratory." *Annals of Oncology* 27(8):1456–66.
- Sato, Toshiro . 2011. "Paneth Cells Constitute the Niche for Lgr5 Stem Cells in Intestinal Crypts." *Nature Publishing Group* 469(7330):415–18.
- Smallbone, Kieran and Bernard M. Corfe. 2013. "A Mathematical Model of the Colon Crypt Capturing Compositional Dynamic Interactions Between Cell Types." *International Journal of Experimental Pathology* 95(1):1–7.
- Snippert, Hugo J. . 2010. "Intestinal Crypt Homeostasis Results From Neutral Competition Between Symmetrically Dividing Lgr5 Stem Cells." *Cell* 143(1):134–44.
- Snippert, Hugo J., Arnout G. Schepers, Johan H. van Es, Benjamin D. Simons, and Hans Clevers. 2013. "Biased Competition Between Lgr5 Intestinal Stem Cells Driven by Oncogenic Mutation Induces Clonal Expansion." *EMBO reports* 15(1):62–69.
- Strubberg, Ashlee M. and Blair B. Madison. 2017. "MicroRNAs in the Etiology of Colorectal Cancer: Pathways and Clinical Implications." *Disease Models & Mechanisms* 10(3):197–214.
- Sunter, J P., D R. Appleton, M S. Dé Rodriguez, N A. Wright, and A J. Watson. 1979. "A Comparison of Cell Proliferation at Different Sites Within the Large Bowel of the Mouse.." *Journal of anatomy* 129(Pt 4):833–42.
- Swanson, Kristin R., Carly Bridge, J D. Murray, and Ellsworth C. Alvord Jr. 2003. "Virtual and Real Brain Tumors: Using Mathematical Modeling to Quantify Glioma Growth and Invasion." *Journal of the Neurological Sciences* 216(1):1–10.
- Totafurno, J, M Bjercknes, and H Cheng. 1987. "The Crypt Cycle. Crypt and Villus Production in the Adult Intestinal Epithelium." *Biophysical Journal* 52(2):279–94.
- Traverso, Giovanni . 2002. "Detection of APC Mutations in Fecal DNA From Patients with Colorectal Tumors." *The New England journal of medicine* 346(5):311–20.
- Tsang, Andy H.-F. 2014. "Current and Future Molecular Diagnostics in Colorectal Cancer and Colorectal Adenoma." *World Journal of Gastroenterology* 20(14):3847–12.
- Vermeulen L, Morrissey E, van der Heijden M, Nicholson AM, Sottoriva A, Buczaccki S, et al. Defining stem cell dynamics in models of intestinal tumor initiation. *Science*. 2013;342: 995–998.
- Williams, Marc J., Benjamin Werner, Chris P. Barnes, Trevor A. Graham, and Andrea Sottoriva. 2016. "Identification of Neutral Tumor Evolution Across Cancer Types." *Nature Genetics* 48(3):238–44.

A Hexagonal Cell Automaton Model to Imitate Physarum Polycephalum Competitive Behaviour

Abubakr Awad¹, Wei Pang¹, David Lusseau² and George M. Coghill¹

¹School of Natural and Computing Sciences, University of Aberdeen, UK

²School of Biological Sciences, University of Aberdeen, UK
abubakr.awad@abdn.ac.uk

Abstract

Slime mould (Physarum) may not have brains, but they are capable of solving many significant and challenging problems. Existing models for studying the intelligent behaviour of Physarum have overlooked its foraging behaviour under competitive settings. In this research, we propose a new model based on Cellular Automata (CA) and Reaction Diffusion (RD) system, where multiple Physarum interact with each other and with their environment. The novelty of our model is that the Physarum has six neighbours at equidistant (hexagonal CA), furthermore, we have extended the model to 3D and multi-dimensional CA grid. The growth of Physarum is determined by the balance between attraction force towards food resources (determined by mass and quality) and repulsion forces between competing Physarum according to their power (mass) and hunger motivation. To validate this model, numerical experiments were conducted. Physarum with more mass succeeded in engulfing a larger number of food resources with high quality in shorter time (number of iteration). It also occupied larger area of the grid (territory) and excluded its competitors. We also conducted empirical analysis to compare the time complexity between the hexagonal and Moore neighbourhood, and it showed that hexagonal neighbourhood is more efficient than Moore in terms of computational cost. To the best of our knowledge, we are the first to present Physarum in competition mathematical model and the algorithms inspired from such a model has demonstrated its promising performance in solving several real world problems such as mobile wireless sensor networks, and discrete multi objective optimization problems.

Introduction

Swarm intelligence is one of the most interesting topics dealing with the collective behaviour of decentralized and self-organized systems. It consists of a population of simple agents which can communicate locally with each other and to their local environment. These interactions can lead to the emergence of very complex global behaviour (Tan and Shi, 2017). A variety of swarm intelligence algorithms for optimization problems such as particle swarm optimization (Eberhart and Kennedy, 1995), ant colony optimization (Dorigo et al., 1996), Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007) have been developed with increasingly wide applications in real-world.

In recent years, cellular computational models based on single cellular organisms function, become an important branch of biology-inspired computing; Bacterial colonies (Kim et al., 2007) and Physarum Polycephalum (Slime mould) (Adamatzky, 2010) are examples. Just like social insects and animals, Physarum too exhibits swarm intelligence; it shares many features of collective behaviour such as synchronization, communication, positive feedback, leadership, and response thresholds (Reid and Latty, 2016). The primitive intelligence of Physarum polycephalum is mostly demonstrated during its plasmodium stage (a giant amoeba-like multi-nucleated single cell). Physarum senses gradients of chemo attractants and repellents and forms a mycelial yellowish vascular network in search of nutrition (Cavender, 1995). The Physarum foraging behaviour consists of two simultaneous self-organized processes of expansion (exploration) and shrinkage (exploitation) (Nakagaki et al., 2001). Physarum protoplasmic flux is changing in a continuous way with the change of external environmental conditions (chemotaxis, phototaxis and thermotaxis) (Cavender, 1995; Jones et al., 2017). This characteristic allows Physarum to have great potentials in dealing with network graph-optimization problems in dynamic environment.

It has been demonstrated that Physarum is capable of finding the shortest path between two points using a simple heuristic while foraging for food (Nakagaki et al., 2000). This has inspired computer scientists to develop novel, bio-inspired algorithms capable of solving many hard problems (Adamatzky, 2010). Much research has confirmed and broadened the range of its computation abilities to spatial representations of various graph problems. Please refer Sun (2017); Zhang et al. (2016b) for more detailed discussions.

In this paper, we present a novel mathematical model to simulate multiple Physarum foraging behaviour in competition settings, based on the trade-offs between their motivation for food, the value of resources (food patch quality), and in the presence of competitors. We assume that the individual skills of competition is more efficient to achieve an optimal balance between exploration and exploitation, and fundamental for the process social collaboration and popu-

lation diversity. We assume that the individual skills of competition is more efficient to achieve an optimal balance between exploration and exploitation, and fundamental for the process social collaboration and population diversity.

The rest of this paper is structured as follows: In Section 2, some related work is reviewed. The novelty of our model is discussed in Section 3, then we proceed to explain our proposed model in Section 4. Numerical examples are given to demonstrate how the proposed model simulate multiple Physarum decision making in competition settings in Section 5. Finally, the paper ends with conclusions and suggestions for further researches.

Related Work

Mathematical Models

Many mathematical models have been proposed to study the Physarum foraging behaviour, for example the flow-conductivity model (Tero et al., 2007), the cellular model (Gunji et al., 2008), the multi-agent model (Jones, 2011), and the shuttle streaming model (Siriwardana and Halgamuge, 2012) are examples. These models were able to solve problems such as finding the shortest path in directed or undirected networks (Zhang et al., 2014), designing supply chain networks (Zhang et al., 2016a), and simulating transport networks (Tsompanas et al., 2015). Up to now these models consider the foraging behaviour when only one Physarum is presented. They did not address the responses of individual Physarum in competition settings, which can lead to the emergence of very complex global behaviour, far beyond the capability of individual Physarum.

Physarum Competitive Behaviour

Competition is generally referred to the negative effects caused by the presence of neighbours, usually by reducing the availability of resources (exploitation-competition). Competition is very important in driving natural selection as a superior competitor can eliminate an inferior one from the area, resulting in competitive exclusion (Hardin, 1960). Imperialist Competitive Algorithm (ICA) (Bernal et al., 2017) and Competition Over food Resources (COR) (Mohseni et al., 2014) are examples of competition algorithms.

There is increasing evidence that simple organisms like Physarum have complex social behaviours including cooperation and competition. Physarum is capable of making complex foraging decisions based on trade-offs between risks, hunger level and food patch quality (Latty and Beekman, 2011). Physarum always initiated foraging behaviour quicker if it was hungry and in the presence of a competitor (Stirrup and Lusseau, 2019). We still do not fully understand how competitive behaviour is integrated in the Physarum decision-making processes in realistic competition settings, so we started by postulating possible heuristics that Physarum might employ in competitive environments.

Work Novelty

Model Selection

After reviewing the literature on several models, we have decided to extend Tsompanas et al. (2016) model, which is based on Cellular Automata (CA) and Reaction-Diffusion (RD) system, to simulate the foraging behaviour of Physarum in the presence of competition. However this model needs modification to address new requirements of competition settings.

Why Cellular Automata and Reaction-Diffusion Systems?

After Wolfram (1984), CA have received extensive academic interest for their fundamental characteristics and capabilities to effectively simulate physical systems, biological systems, and solving scientific problems (Feynman, 1982). CA can capture the essential features of systems where global behaviour arises from the collective effect of simple components, which interact locally. Gunji et al. (2008) have developed a model based on CA to simulate the motion of Physarum as a local protoplasmic flow system.

RD systems are mathematical models which correspond to several physical phenomena. The most common is the change in space and time of the concentration of one or more chemical substances, which causes the substances to spread out over a surface in space. RD systems are naturally applied in chemistry, however, they can also describe dynamical processes of non-chemical nature like in biology (Kondo and Miura, 2010).

Basic Principles for Physarum Mathematical Model in Competition Settings

In the following subsections, we will show how our model differs from the previous model of Tsompanas et al. (2016), which considered only single Physarum foraging behaviour.

Hexagonal Cell Automaton Neighbourhood There are different types of neighbours that can be considered during the experiment design to capture Physarum diffusion direction. Some researchers decided to use four neighbourhood of Von-Neumann (Fig. 1-A) as in Shirakawa et al. (2015), however diagonal diffusion of Physarum (North East, North West, South East and South West) is longer; but it still can occur with lower probability. Other researchers used eight neighbourhoods of Moore (Fig. 1-B) as in Tsompanas et al. (2016). For these reasons, we have considered a new model, where Physarum will have six neighbours (hexagonal pattern) (Fig. 2-A) at equidistant. This will allow circular diffusion of Physarum in contrast to the two of the most famous neighbourhood (Von-Neumann neighbourhood and Moore neighbourhood). Furthermore, this hexagonal grid has the densest packing, the 3D voxels are more sphere-like, and has the highest volumetric quotient (Nagy and Strand, 2009).

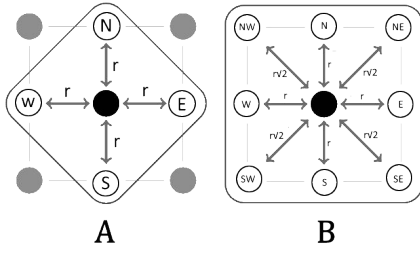


Figure 1: Von Neumann / Moore neighbourhoods.

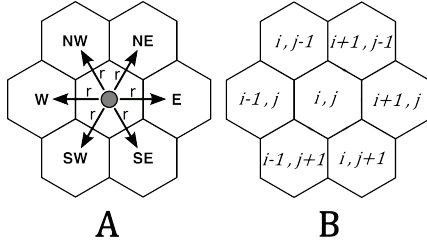


Figure 2: Hexagonal neighbourhood.

Modelling Multiple Physarum and Multiple Food Resources Unlike previous models based on single Physarum, we considered a Physarum competitive behaviour, where a group of Physarum with different power (masses) and motivation (Hunger / Satiety) each having autonomous behaviours react to each other and their own local environment. We also modelled the presence of multiple food resources with different quality.

Attraction / Repulsion Forces affecting Physarum Exploration Unlike previous models which addressed only the attraction force of Physarum towards a food resource, we considered other forces acting on Physarum based on meta-heuristics inspired from Physarum behaviour in a competition setting. We assume that competing Physarum will exert repulsion forces on each other which will affect the evolution of the whole system. We created a new formula to compute two forces acting on Physarum: (A) the chemo attraction force based on the combination of chemical mass and chemical quality, and (B) the repulsion negative forces that competing Physarum exert on each other.

Modelling Physarum Shrinkage (Exploitation) We introduced a new rule on the Physarum diffusion process to imitate the natural process of Physarum shrinkage, where if a Physarum cell is not contributing to the path towards food resource for a certain time (number of iterations), the whole Physarum mass is migrated to the nearest neighbour cell contributing to the path of food resource attraction.

The Proposed Physarum Competition Mathematical Model

As stated before, our proposed model is based on CA and RD systems. We will discuss in details in the following two sections the model state and model rules.

Cellular Automaton (CA) Model State

In order to model the Physarum foraging behaviour in competition settings, where a set of Physarum ($P = p_1, p_2 \dots p_m$) are competing on a set of Chemicals (food resources) ($CHM = chm_1, chm_2 \dots chm_n$). We consider a CA grid in the two-dimensional space, where it is divided into a matrix ($X \times Y$) of identical hexagon cells, each cell $c_{(i,j)}$ at location (i, j) in the grid has six neighbours as shown in Figure 2-B. Moreover, this CA space can be extended to multi-dimensional hexagonal grids as Hexagonal/Body-Centered-Cubic (bcc) grid using the following definition:

Definition 1. Let $\mathbf{p} = (p(1), p(2), \dots, p(n))$ and $\mathbf{q} = (q(1), q(2), \dots, q(n))$ be two points in Z^n . \mathbf{q} is 1-neighbour to \mathbf{p} if $q(1) \equiv q(2) \equiv \dots \equiv q(n) \pmod{2}$ and $|p(i) - q(i)| \leq 1$ for $1 \leq i \leq n$ and $\sum_{i=1}^n (p(i) - q(i))^2 \leq n$.

These grids are the three-dimensional "equivalent" to the two-dimensional hexagonal grid (Nagy and Strand, 2009).

The state of a cell $c_{(i,j)}^t$ at iteration t is described by its type. Initially all cells are empty until it is occupied by food resource and/or Physarum, an obstacle cell (Ex:- Wall), or remain empty (Equation 1). Chemical is defined by its mass and quality while Physarum is defined by its mass and hunger motivation.

$$CT_{(i,j)} = \begin{Bmatrix} FREE \\ OBSTACLE \\ PHYSARUM \\ CHEMICAL \end{Bmatrix} \quad (1)$$

where $CT_{(i,j)}$ represents the cell type.

Cellular Automaton (CA) Model Rules

The CA model rules are mainly based on the diffusion equation (Chopard and Droz, 1991) combined with Physarum heuristics in competition settings. These rules are applied on both Physarum's mass to define the exploration of Physarum on the search space and on food resources' mass to define their diffusion over the grid.

Physarum Competition Heuristics

- (i) A cell can be occupied by one or more chemical class (Ex:- Food resource) and by at most one Physarum in each cell.

- (ii) Chemo attraction forces exerted on Physarum will be a function of food resource (with different mass and quality) and Physarum hunger motivation. If Physarum is satisfied, it would appreciate the quality of chemical rather than the mass, and if it is hungry, vise versa.
- (iii) Competing Physarum will exert repulsion forces on each other which will be calculated as negative force.
- (iv) When two Physarum are competing for the same cell, the one with higher power (mass) will occupy it.
- (v) Food resource engulfed by a Physarum will be excluded, and this Physarum mass will grow at this point according to this food resource quality.
- (vi) In the Physarum exploration phase, if the mass of the Physarum is less than a critical value the Physarum will stop diffusing to prevent the Physarum from spreading all over the board (as exhibited by Physarum in real experiments).
- (vii) In the Physarum exploitation phase, the Physarum tend to shrink and collect its body mass to move towards resource of attraction.
- (viii) Physarum hungry motivation is a parameter that increases with the number of iteration it was unable to find food resource.
- (ix) When a Physarum engulfs food resource, it will be satiated (reset hungry motivation), and stop searching for food for a certain time (number of iterations).

CA Diffusion Equations Every cell occupied by chemical at iteration (t) uses the values of its six neighbours cell to calculate the value of the mass at iteration ($t + 1$). The total chemical mass for a cell $c_{(i,j)}$ at iteration ($t + 1$) is described in Equation 2.

$$CM_{(i,j)}^{t+1} = CM_{(i,j)}^t + \sum_{(k,l)} \begin{cases} CD * (CM_{(k,l)}^t - CM_{(i,j)}^t) & \text{if } C_AA_{(i,j),(k,l)} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\forall(k, l) : \begin{aligned} i - 1 &\leq k \leq i + 1, \\ j - 1 &\leq l \leq j + 1, \\ k &\neq l \end{aligned}$$

where,
 $CM_{(i,j)}^{t+1}$ defines the diffusion of chemical mass for the next generation ($t + 1$) at cell $c_{(i,j)}$.
 $CM_{(i,j)}^t$ is the current mass of the chemical at iteration (t) for cell $c_{(i,j)}$.
 $CM_{(k,l)}^t$ is the current mass of the chemical at iteration (t) for neighbouring chemical cell $c_{(k,l)}$.
 CD is the chemical diffusion coefficient.

$C_AA_{(i,j),(k,l)}$ defines whether chemical at cell $c_{(i,j)}$ is available to diffuse towards a neighbouring cell $c_{(k,l)}$ as defined in Equation 3.

$$C_AA_{(i,j),(k,l)} = \begin{cases} 0, & \text{if } CT_{(k,l)} = \text{"OBSTACLE"} \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

Every cell occupied by Physarum at iteration (t) uses the values of its six neighbours cell to calculate the value of the mass at iteration ($t + 1$). The total Physarum mass for a cell $c_{(i,j)}$ at iteration ($t + 1$) is described in Equation 4.

$$PM_{(i,j)}^{t+1} = PM_{(i,j)}^t + \sum_{(k,l)} \begin{cases} (PF * PD)(PM_{(k,l)}^t - PM_{(i,j)}^t) & \text{if } P_AA_{(i,j),(k,l)} = 1, \\ & PM_{(i,j)}^t > Diff_Limit \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$\forall(k, l) : \begin{aligned} i - 1 &\leq k \leq i + 1, \\ j - 1 &\leq l \leq j + 1, \\ k &\neq l \end{aligned}$$

$$PF = 1 + P_AttForce_{(i,j),(k,l)} + P_RepForce_{(i,j),(k,l)}$$

where,
 $PM_{(i,j)}^{t+1}$ defines the diffusion of Physarum mass for the next generation ($t + 1$) at cell $c_{(i,j)}$.

$PM_{(i,j)}^t$ is the current mass of Physarum at cell $c_{(i,j)}$.

$PM_{(k,l)}^t$ is the current mass of neighbouring Physarum at cell $c_{(k,l)}$.

$P_AA_{(i,j),(k,l)}$ defines whether Physarum at cell $c_{(i,j)}$ is available to diffuse towards a neighbouring cell $c_{(k,l)}$ as defined in Equation 5.

PD is the Physarum diffusion coefficient.

$Diff_Limit$ is the limit which in which the Physarum mass must exceed in order to diffuse.

PF is the sum of forces affecting Physarum.

$P_AttForce_{(i,j),(k,l)}$ defines the value of attraction forces applied on Physarum at cell $c_{(i,j)}$ coming from its neighbouring cell $c_{(k,l)}$ as defined in Equation 6.

$P_RepForce_{(i,j),(k,l)}$ defines the value of repulsion forces applied on Physarum at cell $c_{(i,j)}$ exerted by its neighbouring cell $c_{(k,l)}$ as defined in Equation 7.

$$P_AA_{(i,j),(k,l)} = \begin{cases} 1, & \text{if } CT_{(k,l)} = \text{"FREE"} \\ 1, & \text{if } CT_{(k,l)} = \text{"PHYSARUM"}, \\ & PID_{(i,j)} = PID_{(k,l)} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where,

$PID_{(i,j)}$ is the ID of the Physarum at cell $c_{(i,j)}$.

As seen in the equation calculating the Physarum diffusion, there are attraction and repulsion forces affecting the diffusion of Physarum. The attraction forces as described in Equation 6 determines the movement of Physarum towards the chemical (food resource). It is a function which combines chemical (mass and quality) and Physarum motivation (hungry/satiated).

$$P_AttForce_{c_{(i,j),(k,l)}}^t = \begin{cases} WM_{(i,j)} * \frac{CM_{(k,l)}^t}{Total_NCM_{(i,j)}} + WQ_{(i,j)} * \frac{CQ_{(k,l)}^t}{Total_NCQ_{(i,j)}}, & \text{if } CM_{(k,l)}^t = MAX(CM_{(i,j)}^t) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$WM_{(i,j)} = \frac{PH_{(i,j)}^t}{100}, WQ_{(i,j)} = \frac{100 - PH_{(i,j)}^t}{100}$$

where,

$WM_{(i,j)}$ is the weight assigned to the chemical mass that will attract Physarum to it.

$CM_{(k,l)}^t$ is the current mass of the chemical at iteration (t) for cell $c_{(k,l)}$.

$Total_NCM_{(i,j)}$ is the total summation of chemical mass for the first neighbour of cell $c_{(i,j)}$.

$WQ_{(i,j)}$ is the weight assigned to the chemical quality that will attract Physarum to it.

$CQ_{(k,l)}^t$ is the current quality of chemical at iteration (t) for cell $c_{(k,l)}$, where $CQ \in [0 - 10]$; 0: Low quality, 10: High quality.

$Total_NCQ_{(i,j)}$ is the total summation of chemical quality for the first neighbour of cell $c_{(i,j)}$.

$MAX(CM_{(i,j)}^t)$ is a function which returns the maximum value of chemical mass among the neighbourhood of cell $c_{(i,j)}$.

$PH_{(i,j)}^t$ is the hunger motivation of the Physarum, where $PH \in [0 - 100]$; 0:Satiated, 100:Hungry.

The repulsion forces as described in Equation 7 represent the competition between Physarum, in which every Physarum tries to repel other competitors. This function is based on the neighbour Physarum mass of the opposite direction.

$$P_RepForce_{c_{(i,j),(k,l)}}^t = \begin{cases} \frac{PM_{opp(k,l)}^t}{Total_NPM_{(i,j)}} & \text{if } PID_{(i,j)} \neq PID_{opp(k,l)}, \\ & PM_{opp(k,l)}^t > Rep_Limit \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where,

$PM_{opp(i,j)}^t$ is the neighbour Physarum mass of the opposite direction of cell $c_{(i,j)}$.

$Total_NPM_{(i,j)}$ is the total summation of Physarum

Table 1: Parameters values for the experiments

Parameter	Value	Parameter	Value
CD	0.1	SD	0.05
CM	3000	PM	[1,000-10,000]
CQ	[0-10]	PH	80
$DIFF_LIMIT$	5	REP_LIMIT	5

mass for the first neighbour of cell $c_{(i,j)}$.

Rep_Limit is a limit where Physarum must reach to repel neighbouring Physarum.

Experimental Results

The core model was implemented in Java with Processing package <https://processing.org/> being used for graphical simulation. To validate our model, we have conducted two experiments using the same parameters of diffusion equation as mentioned in Tsompanas et al. (2016) (Table 1). Each experiment was conducted for 100 times to get unbiased results and were statistically analysed using SPSS package.

Experiment (I): Competition and Decision Making

In this experiment design, two Physarum of different masses (high power and low power Physarum) will be competing for multiple (six) food resources placed in a (50×50) hexagonal grid. The aim of the experiment is to show how many food resources are engulfed by the high and low power Physarum and the time (iteration) needed to engulf food resources according to their quality.

This experiment has two different competition settings. In the first setting (A), the two competing Physarum will be randomly placed, and the 6 food resources will be randomly placed each having the same quality. In the second setting (B), the two competing Physarum will be placed in the middle of the grid, three food resources with high quality will be placed randomly in the upper part (North field) of the grid, and three food resources with low quality will be placed randomly in the lower part (South field) of the grid. This experiment was also conducted in duplicate, where the two Physarum exchange location to nullify the chance of better location.

The results were statistically assessed using independent samples t-test to compare means, in the first setting (A) of the experiment, it is shown that Physarum of higher mass (power) succeeded in engulfing larger number of food resources (3.71 ± 1.23) versus (2.29 ± 1.23) by Physarum with lower mass (power) with p -value < 0.001 . Nearly the same results were obtained in the duplicate experiment after changing the two Physarum location which proved that the distance between Physarum and food resource influence

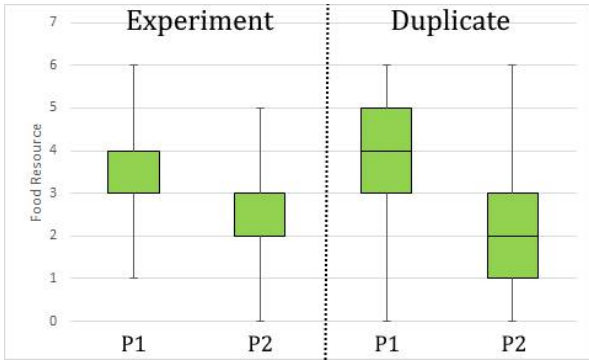


Figure 3: Represents the first setting (A) of Experiment (I) where Physarum of higher mass (P1) engulfed more food resources than Physarum of lower mass (P2).

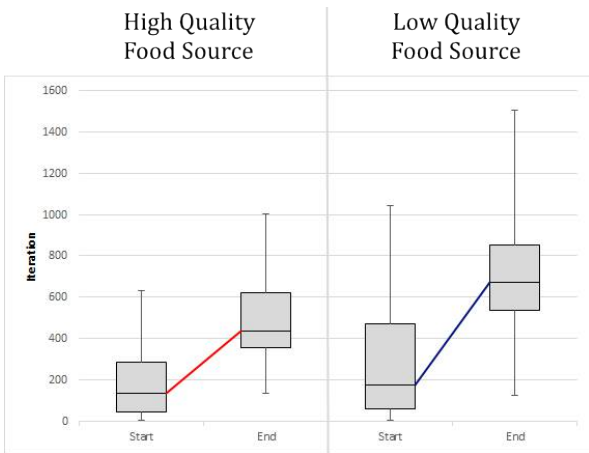


Figure 4: Represents the second setting (B) of Experiment (I) where Physarum started and ended engulfing food resources of high quality (F1) faster than food resources of low quality (F2) as indicated by less number of iterations.

competition but is not the only determining factor. (See Figure 3). In the second setting (B) of the experiment, Physarum forages faster for food resource of high quality as indicated by having fewer iterations (See Figure 4), Physarum start engulfing food resource of high quality after (180 ± 155) iterations and ends by (489 ± 210) iterations versus (279 ± 280) , and (698 ± 242) iterations for food resource of low quality with p -value < 0.001 .

Experiment (II): Physarum imitation of natural Competition scenario

The aim of the experiment is to show the competition and the interaction between multiple Physarum over limited supply of food resources, and territory. In this experiment, ten Physarum of different masses ranging from 1000 to 10000 were randomly placed over CA grid to compete over six food resources, three of them are of high quality (HQF),

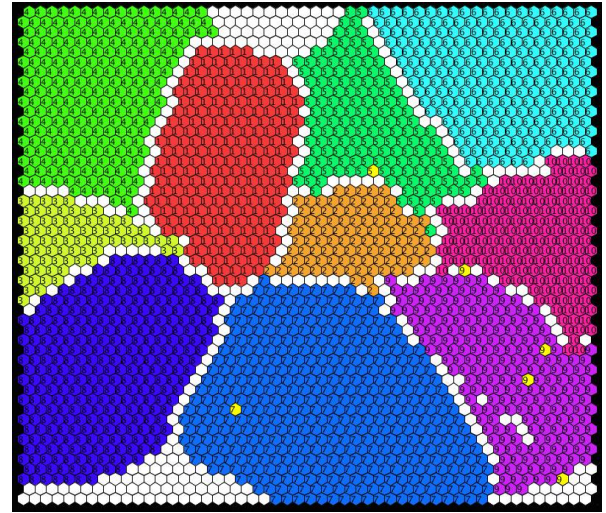


Figure 5: Graphical simulation of 10 Physarum foraging behaviour in competition settings after engulfing all food resources, where yellow cells indicate engulfed food resource and the number inside indicates Physarum (ID) engulfed this food resource, white cells indicate empty cells and other coloured cells indicate Physarum and the number inside indicates the ID of the Physarum (ranging from 1 to 10) and their mass range from 1,000 to 10,000 respectively.

and the other three of low quality (LQF). A graphical simulation of 10 Physarum in competition setting, illustrating food engulfment and territory area, is shown in Figure 5 with video demonstration in this hyperlink <https://youtu.be/oKTGVtanEjE>.

In this experiment we have used bivariate correlation (Pearson correlation coefficient) to assess the effect of Physarum power (mass) on the number of food resources engulfed, territory area (number of cells occupied by each Physarum), and its survival (number of iteration for Physarum to vanish). The results of our experiment showed that Physarum mass significantly correlate with the number of food resources engulfed whether of HQF or LQF with r -value 0.105, 0.102 and p -value $< 0.001, 0.001$ respectively. Physarum with bigger mass (best competitor) was able to occupy larger territory with r -value 0.249 and p -value < 0.001 , and would always exclude its competitors and has longer survival with r -value 0.128 and p -value < 0.001 .

Finally, we conducted empirical analysis to compare the computational complexity between hexagonal and Moore neighbourhood. In this analysis, we measured the time complexity using our own defined function that counts the number of execution of the diffusion equation. It showed that hexagonal neighbourhood on average is efficient than Moore neighbourhood by 11%.

Conclusions & Future Work

In this study, we presented complex patterns observed in Physarum polycephalum generated in competition settings. This model is based on Cellular Automata and Reaction-Diffusion system, where the growth of Physarum is the resultant of chemo attraction towards food resources and repulsion forces between competing Physarum. The existing models are based on imitating single Physarum foraging behaviour and it did not address the skills of individual Physarum in competition settings. To the best of our knowledge, this is the first time Physarum will have been simulated in a hexagonal grid, which is more applicable to Physarum natural diffusion in a circular pattern to equidistant cell neighbours.

Experimental results clearly showed that our model was able to simulate Physarum complex competition behaviours, where multiple Physarum compare information on reward (food resources' mass and quality), and negative effects of competing neighbours according to their hunger motivation in order to make correct and adaptive decisions. Physarum with higher mass (best competitor) succeeded in engulfing a larger number of food resources, and was able to occupy larger area of the grid (territory). Also According to the competitive exclusion principle, Physarum less suited (lower mass) to compete for resources will die earlier than strong competitors as measured by number of iterations, which is an important hypothesis in natural selection.

Our model has been proved to be useful in solving multiple origin-destination network optimization problems as in mobile wireless sensor networks (Awad et al., 2019a, 2018), and discrete multi objective optimization problems (Awad et al., 2019b). This new model will be feasible for biologists to carry out wet-lab experiments for model validation and increasing our understanding of the possible heuristics that Physarum use in complex foraging decisions in competition settings.

Acknowledgement

Abubakr Awad research is supported by Elphinstone PhD Scholarship (University of Aberdeen). Wei Pang, George Coghill, and David Lusseau are supported by the Royal Society International Exchange program (Grant Ref IE160806).

References

- Adamatzky, A. (2010). *Physarum Machines: Computers from Slime Mould*. World Scientific.
- Awad, A., Pang, W., and Coghill, G. (2018). *Physarum inspired model for mobile sensor nodes deployment in the presence of obstacles*, volume 200 of *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*.
- Awad, A., Pang, W., and Coghill, G. M. (2019a). *Physarum inspired connectivity and restoration for wireless sensor and actor networks*, volume 840 of *Advances in Intelligent Systems and Computing*.
- Awad, A., Usman, M., Lusseau, D., Coghill, G. M., and Pang, W. (2019b). A physarum-inspired competition algorithm for solving discrete multi-objective optimization problems. In *Genetic and Evolutionary Computation Conference Companion (GECCO 19 Companion)*, GECCO '19, New York, NY, USA. ACM.
- Bernal, E., Castillo, O., Soria, J., and Valdez, F. (2017). Imperialist competitive algorithm with dynamic parameter adaptation using fuzzy logic applied to the optimization of mathematical functions. *Algorithms*, 10(1).
- Cavender, J. (1995). Myxomycetes: A handbook of slime molds. *Bioscience*, 45(11):795–797.
- Chopard, B. and Droz, M. (1991). Cellular automata model for the diffusion equation. *Journal of Statistical Physics*, 64(3-4):859–892.
- Dorigo, M., Maniezzo, V., and Colnori, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.
- Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE.
- Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488.
- Gunji, Y. P., Shirakawa, T., Niizato, T., and Haruna, T. (2008). Minimal model of a cell connecting amoebic motion and adaptive transport networks. *Journal of theoretical biology*, 253(4):659–667.
- Hardin, G. (1960). The competitive exclusion principle. *Science*, 131(3409):1292–1297.
- Jones, J. (2011). Influences on the formation and evolution of physarum polycephalum inspired emergent transport networks. *Natural Computing*, 10(4):1345–1369.
- Jones, J., Mayne, R., and Adamatzky, A. (2017). Representation of shape mediated by environmental stimuli in physarum polycephalum and a multi-agent model. *International Journal of Parallel, Emergent and Distributed Systems*, 32(2):166–184.
- Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471.
- Kim, D. H., Abraham, A., and Cho, J. H. (2007). A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, 177(18):3918–3937.
- Kondo, S. and Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620.
- Latty, T. and Beekman, M. (2011). Speed-accuracy trade-offs during foraging decisions in the acellular slime mould physarum polycephalum. *Proceedings of the Royal Society B: Biological Sciences*, 278(1705):539–545.

- Mohseni, S., Gholami, R., Zarei, N., and Zadeh, A. R. (2014). Competition over resources: A new optimization algorithm based on animals behavioral ecology. In *Proceedings - 2014 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2014*, pages 311–315.
- Nagy, B. and Strand, R. (2009). Neighborhood sequences on nd hexagonal/face-centered-cubic grids. In *International Workshop on Combinatorial Image Analysis*, pages 96–108. Springer.
- Nakagaki, T., Yamada, H., and Tth, . (2000). Maze-solving by an amoeboid organism. *Nature*, 407(6803):470.
- Nakagaki, T., Yamada, H., and Tth, . (2001). Path finding by tube morphogenesis in an amoeboid organism. *Biophysical chemistry*, 92(1-2):47–52.
- Reid, C. R. and Latty, T. (2016). Collective behaviour and swarm intelligence in slime moulds. *FEMS microbiology reviews*, 40(6):798–806.
- Shirakawa, T., Sato, H., and Ishiguro, S. (2015). Construction of living cellular automata using the physarum plasmodium. *International Journal of General Systems*, 44(3):292–304.
- Siriwardana, J. and Halgamuge, S. K. (2012). Fast shortest path optimization inspired by shuttle streaming of physarum polycephalum. In *2012 IEEE congress on evolutionary computation*, pages 1–8. IEEE.
- Stirrup, E. and Lusseau, D. (2019). Getting a head start: the slime mold, physarum polycephalum, tune foraging decision to motivational asymmetry when faced with competition. *arXiv e-prints*, page ar:1905.06534. 1905.06534; Provided by the SAO/NASA Astrophysics Data System.
- Sun, Y. (2017). Physarum-inspired network optimization: A review. *CoRR*, abs/1712.02910. 1712.02910.
- Tan, Y. and Shi, Y. (2017). Editorial: Special section on bio-inspired swarm computing and engineering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(1):1–3.
- Tero, A., Kobayashi, R., and Nakagaki, T. (2007). A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of theoretical biology*, 244(4):553–564.
- Tsompanas, M.-A. I., Sirakoulis, G. C., and Adamatzky, A. (2016). *Cellular Automata Models Simulating Slime Mould Computing*, pages 563–594. Advances in Physarum Machines. Springer.
- Tsompanas, M. A. I., Sirakoulis, G. C., and Adamatzky, A. I. (2015). Evolving transport networks with cellular automata models inspired by slime mould. *IEEE Transactions on Cybernetics*, 45(9):1887–1899.
- Wolfram, S. (1984). Computation theory of cellular automata. *Communications in mathematical physics*, 96(1):15–57.
- Zhang, X., Adamatzky, A., Yang, X. S., Yang, H., Mahadevan, S., and Deng, Y. (2016a). A physarum-inspired approach to supply chain network design. *Science China Information Sciences*, 59(5).
- Zhang, X., Gao, C., Deng, Y., and Zhang, Z. (2016b). *Slime Mould Inspired Applications on Graph-Optimization Problems*, pages 519–562. Advances in Physarum Machines: Sensing and Computing with Slime Mould. Springer International Publishing, Cham.
- Zhang, X., Wang, Q., Adamatzky, A., Chan, F. T. S., Mahadevan, S., and Deng, Y. (2014). An improved physarum polycephalum algorithm for the shortest path problem. *The Scientific World Journal*, 2014.

A simplified model of chromatin dynamics drives differentiation process in Boolean models of GRN

Michele Braccini¹, Andrea Roli^{1,4}, Marco Villani^{2,4}, Sara Montagna¹ and Roberto Serra^{2,3,4}

¹Dept. of Computer Science and Engineering, *Alma Mater Studiorum* Università di Bologna

²Dept. of Physics, Informatics and Mathematics, Università di Modena e Reggio Emilia

³Institute for Advanced Study (IAS), University of Amsterdam

⁴European Centre for Living Technology, Venice

m.braccini@unibo.it

Abstract

Cellular types of multicellular organisms are the stable results of complex intertwined processes that occur in biological cells. Among the many others, chromatin dynamics significantly contributes—by modulating access to genes—to differential gene expression, and ultimately to determine cell types. Here, we propose a dynamical model of differentiation based on a simplified bio-inspired methylation mechanism in Boolean models of GRNs. Preliminary results show that, as the number of methylated nodes increases, there is a decrease in attractor number and networks tend to assume dynamical behaviours typical of ordered ensembles. At the same time, results show that this mechanism does not affect the possibility of generating path dependent differentiation: cell types determined by the specific sequence of methylated genes.

Introduction

Eukaryotic cells are characterised by the organisation of DNA in a condensed structure, called chromatin. Chromatin is composed of nucleosomes, structures of DNA wrapped around octamers of histone proteins. Histone methylation and histone acetylation change—by adding methyl and acetyl groups to histones—the degree of compactness of the chromatin, in this way facilitating or obstructing gene expression. These processes are defined as epigenetic mechanisms¹.

Although methylation effects depend on the particular positions on histones on which it acts, it most often leads to tightly packed regions of chromatin called **heterochromatin** (Gilbert and Barresi, 2016; Perino and Veenstra, 2016; Schuettengruber and Cavalli, 2009). These regions are not accessible neither by transcription factors nor by RNA polymerases and so the expression of genes belonging to these DNA areas is inhibited. Biological cells exploit differential methylation to modulate their gene expres-

¹In a manner conforming with molecular biology, with the term epigenetics we refer to the series of heritable mechanisms not directly derived from changes in DNA that modify the cells' behaviour. This is not to be confused with the different concept of "epigenetic landscape" (Waddington, 1957; Huang, 2012), metaphor introduced by Waddington to represent the developmental landscape determined by the GRN.

sion during development and differentiation. It is important to note that, along lineages, the attained configurations of DNA methylation are inherited and progressively extended as cells become more specialised (Kim and Costello, 2017). Therefore, methylation contributes to maintain and stabilise the attained gene expressions that ultimately characterise the identities of the various cell states.

It is worth mentioning that methylation is tightly regulated by complex interactions, and that epigenetic dysregulation is very common in a lot of disorders, from cognitive, neurological and chronic diseases to cancer. Given the complexity of these mechanisms, the adoption of models can support the analysis of the role of epigenetics in pathophysiological processes. Several mathematical approaches have been proposed with the aim of disentangling the effects of epigenetics in development, differentiation and also in the establishment of aberrant cellular states—like cancer.

Noteworthy is the work (Miyamoto et al., 2015) in which the authors investigate the mechanisms of differentiation and cellular reprogramming introducing a continuous model of a minimal gene regulatory network (GRN) able to give rise to both pluripotent and differentiated states. In their modelling approach, an epigenetic process—introduced as a gene expression fixation—turns out to be important to increase the stability of the attained differentiated states and to reproduce with more accuracy the phenomenology of the reprogramming process.

In the works (Turner et al., 2017, 2013), the authors have ascertained that the addition of an epigenetic layer—in the form of Boolean switches that dynamically change the actual network topology—within recurrent neural networks lead to better performance in the achievement of certain target tasks, as compared to models without it.

To the best of our knowledge, the specific role of epigenetics in the dynamics of discrete models of GRN has been addressed only by (Bull, 2014). The author does not focus on the differentiation process as such, but instead, he evaluates the potential of Random Boolean networks (RBNs) with epigenetic control—which is interpreted as additional nodes that change the regular transcription dynamics—in NK land-

scapes (Kauffman and Levin, 1987).

Of fundamental relevance for this discussion are the works of Kauffman and Huang (Kauffman, 1969; Huang et al., 2009, 2005; Huang and Ingber, 2000) that have laid the foundations for a mathematical description of GRN dynamics in terms of dynamical systems, with attractors that model cell types.

Recently, an abstract mathematical model of differentiation based on Noisy RBNs has been proposed (Villani et al., 2011; Villani and Serra, 2013; Serra et al., 2010). This model has proven to be able to describe the most relevant properties of the differentiation process, such as different degrees of differentiation, stochastic and deterministic differentiation, and cell reprogramming. The model focuses on the dynamics of a single cell represented as an *autonomous* system² subject to intracellular noise. Cell types are defined as the portions of the space of states in which the dynamics remains trapped, under a specific noise level. Changes in the intracellular level of noise drive the differentiation process: high noise levels correspond to pluripotent cells while the low levels to fully differentiated ones. Experimental analysis on RBNs subject to stochastic dynamics (Braccini et al., 2018) and the successful evolution of networks able to attain not trivial differentiation dynamics (Braccini et al., 2017; Benedettini et al., 2014) proved the expressiveness and plausibility of this model.

Differentiation represents a major challenge for every model of gene regulatory networks that, like RBNs, is based on deterministic dynamical systems which asymptotically reach stable attractor states, to be identified with the different cell types. Indeed, under the action of the deterministic dynamics, a stable attractor does not change any longer so it must represent a fully differentiated cell type. Therefore cells which are found at intermediate differentiation levels (e.g. pluripotent cells) should be associated to transients—an unsatisfactory proposal, since it is known that there exist long-lived pluripotent cells, which should rather be represented by metastable states.

The way out of this conundrum requires a mechanism to escape from the deterministic attractors. While this mechanism is provided in our previously described model by means of intracellular noise, in this work we want to explore an alternative—complementary—possibility, i.e. that it is due to an external signal. In this way, the system is no longer autonomous, and escaping from the attractors of the corresponding deterministic system becomes possible. External signals are indeed known to affect embryo evolution, and the simplest way to describe their effect in a GRN model is that of clamping the values of some network nodes to fixed values.

²Here we adopt the terminology of dynamical systems in which the adjective *autonomous* is used to denote systems that are not subject to inputs, therefore their state may change in time only owing to internal mechanisms.

This paper is organised as follows. The next section describes the proposed model with its theoretical implications in BN models and its contribution to the understanding of the biological process of methylation. The subsequent section details the experimental setting of our *in silico* experiments and illustrates results that show the properties of the model. Finally, we conclude with an outlook to future work.

Model

Boolean networks are discrete-time and discrete-state dynamical models of GRN introduced by Kauffman (Kauffman, 1969, 1993). In their original formulation BNs can be represented by a directed graph with n nodes each having associated a Boolean variable x_i , $i = 1, \dots, n$ and a Boolean function $f_i = (x_{i_1}, \dots, x_{i_k})$ which depends on k other nodes, avoiding self loops. Despite their simplifications they proved to be suitable systems to represent the dynamics of biological GRNs to many level of abstractions (Graudenzi et al., 2011; Serra et al., 2006, 2007; Shmulevich et al., 2005).

As previously discussed, methylation—even if it is not the only phenomenon in place—has a non negligible impact on cell fate determination and maintenance. Here we are especially interested in its abstract role in simplified models of GRNs, namely in Boolean networks. Indeed, borrowing the idea of a progressive methylation state of the chromatin along the development and differentiation of biological cells, we propose an analogous mechanism in BN models. Similarly to what happens in the heterochromatin condition, the expression of some BN nodes is blocked to value 0; these nodes will be referred to as **frozen** in the following.

Theoretically, the formulation of this peculiar methylation mechanism implies a sort of simplification of the network, as it reduces the nodes that are actually subject to a dynamic update, and so restricting the number of combinations that the system itself can assume. Therefore, it is not a priori clear whether this mechanism can accommodate **path dependent differentiation**: cell types determined by the specific sequences of methylated genes.

This model relies on the hypothesis—to be verified in RBNs—that the progression of frozen nodes imposes the arrow of time of the differentiation process and, at the same time, different patterns of methylated nodes give rise to distinct lineages, and so cell types. Indeed, biological differentiation is characterised by the presence of different stages of differentiation and by progressively specialisation of cells.

A schematic representation of this *Boolean methylation-inspired mechanism* is depicted in Figure 1. In this work we undertake an experimental analysis of the main dynamical properties of RBNs subject to this process of progressive methylation. For this mechanism to be useful in a plausible BN differentiation model, it should (i) progressively stabilise the network and (ii) give origin to different lineages depending on the nodes chosen to be frozen. If these prop-

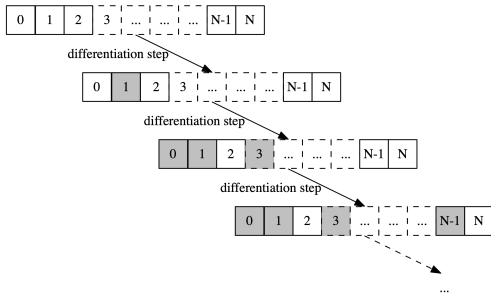


Figure 1: Schematic representation of the methylation mechanism introduced. Grey nodes represent frozen nodes (nodes constrained to assume the value 0, regardless of the actual values of its inputs). The specific patterns of frozen nodes in the ensembles that represent the state of the BN over time have no other meaning than to exemplify the methylation process introduced.

erties are attained in RBN ensembles, then we could suppose that evolution may act to tune the dynamics of the network so as to achieve a specific differentiation lineage tree. The choice of setting to 0 the nodes to be frozen is motivated by the inhibition effect of most methylation mechanisms and introduces an asymmetry in the RBNs model, as it progressively bias the Boolean functions to 0. However, this is not a limitation of the model, which can be extended to take into account also actions in which nodes are clamped to 1 and so provide even more variability in the lineages.

Results

The random Boolean networks used in these experiments are subject to a synchronous and deterministic dynamics, therefore fixed points and cycles are possible asymptotic states. For all the experiments, statistics are taken across 100 RBN with $n = 500$ and $k = 2$. We focused only on networks with $k = 2$ because the size of the network, combined with the other chosen parameters, would have made the experimental analysis computationally prohibitive. The Boolean functions are defined on the basis of the bias parameter p , which defines the probability to assign value 1 in a row of a node truth table. The variation of the parameter p makes it possible to determine the dynamical regime of the system (ordered, critical or chaotic) (Bastolla and Parisi, 1997): so, the limitation due to the choice of a specific connectivity is thus eliminated. Since we want to analyse the emerging generic properties induced only by the proposed methylation mechanism in ensembles of RBNs, we used an *exact* bias. Exact bias is computed by generating each time a random permutation of a vector of Boolean values with a length equal to the number of nodes in the network and a fraction p of 1's, and by using partitions of this vector to define the output values of Boolean functions. In this way, we remove

from the statistics any possible contribution produced by a variance in network dynamic regime. We generated RBNs with $p = 0.1$, i.e. in the ordered regime, and $p = 0.5$, corresponding to the critical regime. As results with ordered RBNs are rather uninformative, we only show results for critical RBNs.

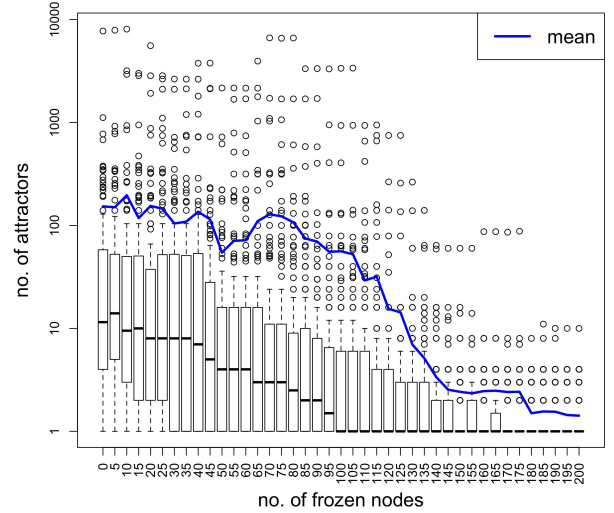


Figure 2: Distribution of the number of attractors for the configuration $n = 500$, $k = 2$, $p = 0.5$ as the number of frozen nodes increases from 0 to 200 with a frozen step of 5 nodes at a time. The continuous line illustrates the trend of the mean.

Attractor number distribution To providing the trend of the number of attractors as the fraction of frozen nodes increases we generated 100 RBNs and for each number of frozen nodes we performed a search of the attractors starting from 10^4 random initial states. The range of frozen nodes considered varies from 0 to 200 with a step of 5 nodes. Boxplots showing the the distribution of the number of attractors as a function of the number of frozen nodes are depicted in Figure 2, along with the mean of these distributions. As expected, the number of attractors decreases with the number of frozen nodes, even though it remains non negligible up to one fifth of frozen nodes. A question may arise as to how many attractors are fixed points, as one expects an increasing number of fixed points as the RBNs become more ordered. This expectation is indeed confirmed, as shown in Figure 3.

Derrida parameter With the aim of assessing the intuition suggesting a progressive shift towards an ordered regime of the ensemble of RBNs subject to the methylation mechanism, we computed the distribution of the Derrida parameter (Bastolla and Parisi, 1997) λ , computed after one

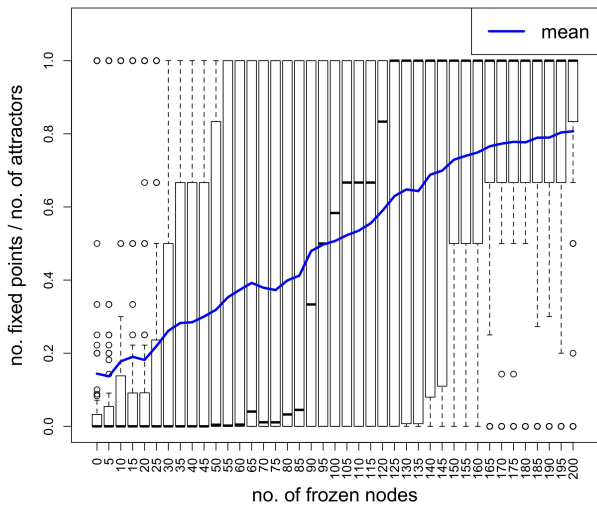


Figure 3: Distribution of the number of fixed points over the number of attractors for the configuration $n = 500$, $k = 2$, $p = 0.5$ as the number of frozen nodes increases from 0 to 200 with a frozen step of 5 nodes at a time. The continuous line illustrates the trend of the mean.

step. This parameter is evaluated by taking, for each state considered (10^3 in total), the means of the Hamming distances after one update between the state and the perturbed one (a logic negation of a single node value) in all not frozen nodes, taken one at a time. In particular, statistics report the distributions of the 100 means of the means, one parameter value for each RBN which summarises the overall behaviour observed along the 10^3 random states. For this investigation, we consider a number of frozen nodes represented by a percentage of $\{0, 10, 20, 50\}$ of all nodes. Figure 4 depicts the boxplots summarising the distribution of λ for the ensembles sampled; the trend towards an increasing order is confirmed (the results for $p = 0.1$, on the left in Figure 4, are provided as a comparison).

The results shown so far support the conjecture that a progressive freezing pushes RBNs towards order. One may argue that a result not in agreement with this expectation might indeed sound surprising, nevertheless it is important to assess it experimentally in particular because this trend is not trivial at all in finite-size RBNs. While in infinite-size RBNs just a tiny fraction of frozen nodes leads to a complete stasis of the network³, in finite-size RBNs we observe that the number of attractors and the Derrida parameter are kept at significant values even in the presence of a non-negligible fraction of frozen nodes. This result suggests that in finite-size RBNs, while a progressive freezing tends to increase order in network dynamics, it may still be open to variabil-

³a formal model of this behaviour is subject of ongoing work

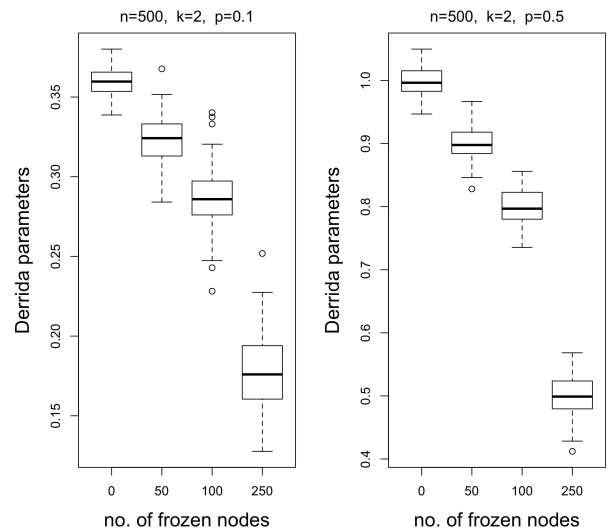


Figure 4: Distribution of the Derrida parameters as the number of frozen nodes increases, for both ordered and critical ensembles of 100 RBNs with $N = 500$.

ity. This last characteristic is relevant especially with respect to the possible paths across attractors that are feasible as the consequence of different choices in the nodes to be frozen.

Diversity estimation In previous sections we have summarised with *path dependent differentiation* the property of generating different cell types as a result of different sequences of methylated genes. We can characterise the tendency of this mechanism to give rise to this property by inspecting the *diversity* caused by different combinations of methylated genes at any attained differentiation stage. For this purpose, we generate for each state of the methylation process (state represented by the already frozen nodes and the attractor reached) 10^2 couples of triplets of nodes,⁴ among the non-already methylated nodes. This triplet is frozen while the network is in an asymptotic state, therefore after this perturbation the BN dynamics is subject to a transient and subsequently the network can either return to an attractor equal to the current one—except for the frozen triplet—or reach a different one. The freezing step may be taken at any state—i.e. *phase*—of the current attractor; as the phase of the attractor may be a source of variability and here we want to assess the contribution of the choice of frozen nodes only, once the attractor is reached after freezing a triplet of nodes, its minimum state according to the

⁴The choice of 3 nodes is somehow arbitrary, but motivated by the requirement of involving a small number of nodes to be frozen, while keeping the possibility of significantly perturbing the attractor. However, previous preliminary experiments on different network size and number of frozen nodes confirm the qualitative behaviour we show in this work.

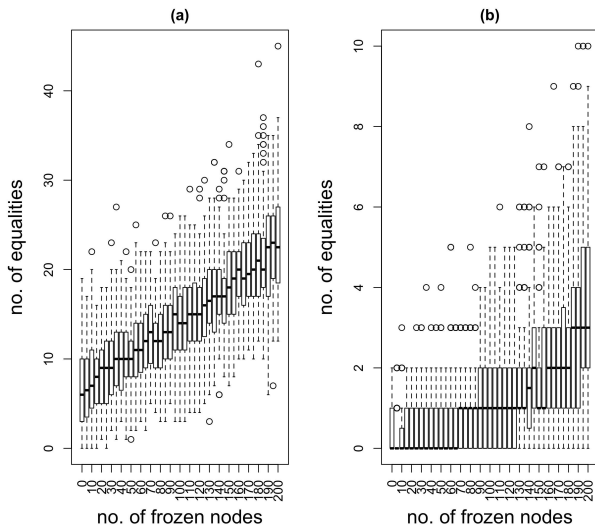


Figure 5: The trend of the number of equal reached asymptotic states considered in pairs and after removing the part of the already frozen nodes (x-values) and the set of nodes that constitute the triplets. **(a)** Triplets randomly chosen among all the non-frozen nodes, **(b)** triplets randomly chosen among the non-frozen nodes with value 1.

lexicographic order is chosen. As networks are random, this choice does not introduce any bias and in this way we rule out any possible contribution of attractor phase in the diversity of paths originated by freezing steps.⁵ The diversity is then measured depending on the characteristics of the new asymptotic states on which the dynamics settles after the triplet is frozen. As we aim at providing general results, not bound to a specific definition of *phenotype*⁶, which should be supported by motivations on a concrete biological case, we analyse the arising diversities in various condition. Particularly, we count:

- the number of equal reached asymptotic states considered in pairs and after removing the part of the already frozen nodes and the set of nodes that constitute the triplets;
- the differences among all the reached attractors caused by the generated triplets, by considering subset of genes (patterns in the following) of different sizes (10, 50, 100) randomly chosen;
- the differences among all the reached attractors caused by the generated triplets, by considering the states vectors in their entirety.

⁵In other words, we pose us in the condition of minimal diversity.

⁶See the following parts of the text for a more detailed discussion.

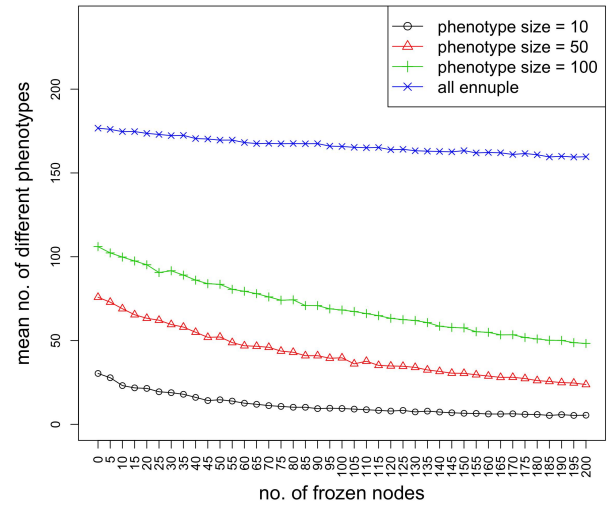


Figure 6: The trend of the number of *diversities* caused by 200 triplets of frozen nodes. **The triplets are randomly chosen among the non-already frozen nodes.** Diversities are measured by considering if the reached attractors are different (all tuples case) or by means of randomly chosen patterns (of sizes equal to 10, 50, 100) which select the nodes on which perform the comparison between the reached attractors.

By doing so we will have an overall picture of how this mechanism behaves in ensembles of RBNs, without limiting ourselves to particular points of view. As for the attractors distribution analysis, the range of frozen nodes considered varies from 0 to 200 with a step of 5 nodes. We stress that in this model the various degrees of differentiation are characterised by a distinct number of frozen nodes: the higher the number of frozen nodes the more differentiated the cell types. The triplets to be frozen are chosen at random among all the non already frozen nodes; we also made experiments with conditioning this choice to nodes that assume value 1 in the attractor state chosen for the perturbation. In this way, we can assess the highest level of variability that can be attained, as all the three nodes are actually perturbed by freezing.

The distribution of the frequency of equal pairs of attractors is shown in Figure 5; we observe that the median frequency of equal pairs increases from about 7/100 to 20/100 with the number of frozen nodes, while it is limited to low percentages when frozen nodes are chosen among the active ones (value 1). This result shows that the probability of choosing two different triplets⁷ leading to the same asymptotic state after being frozen is rather low; therefore, at least

⁷As triplets are chosen at random among at least 300 nodes, the fraction of equal ones is negligible.

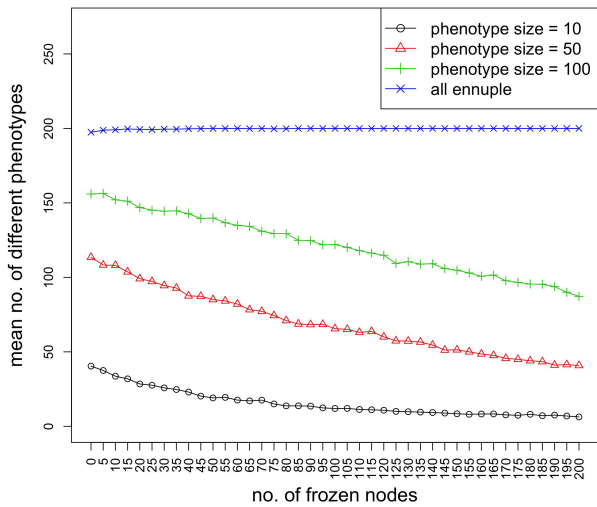


Figure 7: The trend of the number of *diversities* caused by 200 triplets of frozen nodes. **The triplets are randomly chosen among nodes that have value 1, in the chosen phase of the starting attractor.** Diversities are measured by considering if the reached attractors are different (all tuples case) or by means of randomly chosen patterns (of sizes equal to 10, 50, 100) which select the nodes on which perform the comparison between the reached attractors.

for RBNs with at most 2/5 of frozen nodes, the different paths generated by freezing are a significant fraction of all the possible ones, despite the tendency towards a more ordered regime. This observation is confirmed also by the statistics involving the total number of different patterns. With the term *pattern* we refer to a projection of the network dynamics in subset of nodes. So, patterns in this context define the observable phenotypes in a way strongly related to the concept of macrostate introduced in (Borriello et al., 2018; Moris et al., 2016). These latter results are shown in Figures 6 and 7. It is worth observing that, even when differences are estimated on the basis of 10 nodes, the fraction of overall different patterns is still non-negligible up to 100 frozen nodes out of 500.

These results support the hypothesis that different freezing patterns in RBNs are very likely to produce different trajectories along attractors, and therefore variability in differentiation paths can be attained also by means of this mechanism.

Conclusion

In this work we have explored the possibility of incorporating epigenetic mechanisms—methylation in particular—into BN models of GRNs. We focused on those processes responsible for high chromatin compaction, that influences

gene transcription by controlling the accessibility of DNA to transcription factors and RNA polymerases. Accordingly, in our model we progressively freeze—i.e. clamp to 0—a subset of nodes and analyse the impact of this modification on network dynamical features, namely on attractor number—in analogy with the number of cell types—, on the Derrida parameter—to assess the extent to which RBNs with frozen nodes tend to an ordered regime—and on attractor diversity as attained by different combinations of frozen nodes.

We observed that the number of attractors in RBNs decreases with the number of frozen nodes and the same does the Derrida parameter, suggesting that, from an ensemble point of view, the larger the fraction of frozen nodes the more ordered the RBNs. These results are in agreement with the intuition that, by clamping to 0 a fraction of RBN nodes, not only the state space is reduced with respect to the original network, but frozen nodes absorb perturbations and so they favour network stability. These properties are to some extent the abstract counterpart of progressive reduced alternatives and stability along differentiation stages. Moreover, results show a very interesting property of RBNs: they maintain diversity in terms of possible asymptotic states originating from different combinations of frozen nodes, both during the process of progressive freezing itself and in the final reached states. We assessed this diversity by means of three metrics, so as to attain general results. We found that different choices in nodes to be frozen are very likely to lead to different asymptotic states, implying that diverse differentiation paths can be generated. As expected, this diversity tends to decrease with the fraction of frozen nodes in the network.

As future work, we plan to add in our model mechanisms to reproducing open chromatin structure, where genes are made more accessible and their transcriptions eased. The combined effects of both closing and opening chromatin structure on attractors and other relevant features of BNs will be consequently analysed. Moreover, since epigenetic is expected to have an impact on cell type stability, we are devising a set of experiments to measure how attractor robustness changes along the path of differentiation, for example by measuring the impact of external signals—possibly modulated—during different stages of differentiation. To conclude, epigenetic is only one of the factors that are responsible for cell type transitions and definitions. Signalling cues, typically generated by other cells, are another crucial actor in the process of differentiation. In this perspective, we are planning to study models involving networks of BNs, so as to explore the possibility of modelling differentiation in a multi-cellular setting.

Acknowledgements

We gratefully thank Emanuele Domenico Giordano, Alice Pasini and Marilisa Cortesi of Laboratory of Cellular and Molecular Engineering “S. Cavalcanti” for the useful discus-

sions on the model and its biological significance. Andrea Roli is a member of the IndAM research group GNCS.

References

- Bastolla, U. and Parisi, G. (1997). A numerical study of the critical line of kauffman networks. *Journal of Theoretical Biology*, 187(1):117 – 133.
- Benedettini, S., Roli, A., Serra, R., and Villani, M. (2014). Automatic design of Boolean networks for modelling cell differentiation. *Evolution, Complexity and Artificial Life*, 708:77–89.
- Borriello, E., Walker, S. I., and Laubichler, M. D. (2018). A unified, mechanistic framework for developmental and evolutionary change. *arXiv preprint arXiv:1809.02331*.
- Braccini, M., Roli, A., Villani, M., and Serra, R. (2017). *Automatic Design of Boolean Networks for Cell Differentiation*, pages 91–102. Springer International Publishing, Cham.
- Braccini, M., Roli, A., Villani, M., and Serra, R. (2018). A comparison between threshold ergodic sets and stochastic simulation of boolean networks for modelling cell differentiation. In Pelillo, M., Poli, I., Roli, A., Serra, R., Slanzi, D., and Villani, M., editors, *Artificial Life and Evolutionary Computation – 12th Italian Workshop, WIVACE 2017, Revised Selected Papers*, volume 830 of *CCIS*, pages 116–128. Springer.
- Bull, L. (2014). Evolving boolean regulatory networks with epigenetic control. *Biosystems*, 116:36–42.
- Gilbert, S. F. and Barresi, M. J. F. (2016). *Developmental biology*, chapter 3, pages 50–52. Sinauer Associates Inc, XI edition.
- Graudenzi, A., Serra, R., Villani, M., Damiani, C., Colacci, A., and Kauffman, S. A. (2011). Dynamical properties of a boolean model of gene regulatory network with memory. *Journal of Computational Biology*, 18(10):1291–303.
- Huang, S. (2012). The molecular and mathematical basis of Waddington’s epigenetic landscape: A framework for post-Darwinian biology? *BioEssays*, 34(2):149–157.
- Huang, S., Eichler, G., Bar-Yam, Y., and Ingber, D. (2005). Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical Review Letters*, 94:128701:1–4.
- Huang, S., Ernberg, I., and Kauffman, S. (2009). Cancer attractors: A systems view of tumors from a gene network dynamics and developmental perspective. *Seminars in Cell & Developmental Biology*, 20(7):869 – 876.
- Huang, S. and Ingber, D. (2000). Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Experimental cell research*, 261(1):91–103.
- Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467.
- Kauffman, S. A. (1993). *The origins of order*. Oxford University Press.
- Kim, M. and Costello, J. (2017). Dna methylation: an epigenetic mark of cellular memory. *Experimental & molecular medicine*, 49(4):e322.
- Miyamoto, T., Furusawa, C., and Kaneko, K. (2015). Pluripotency, differentiation, and reprogramming: a gene expression dynamics model with epigenetic feedback regulation. *PLoS computational biology*, 11(8):e1004476.
- Moris, N., Pina, C., and Arias, A. M. (2016). Transition states and cell fate decisions in epigenetic landscapes. *Nature Reviews Genetics*, 17(11):693.
- Perino, M. and Veenstra, G. (2016). Chromatin control of developmental dynamics and plasticity. *Developmental Cell*, 38(6):610–620.
- Schuettengruber, B. and Cavalli, G. (2009). Recruitment of polycomb group complexes and their role in the dynamic regulation of cell fate choice. *Development*, 136(21):3531–3542.
- Serra, R., Villani, M., Barbieri, A., Kauffman, S., and Colacci, A. (2010). On the dynamics of random Boolean networks subject to noise: attractors, ergodic sets and cell types. *Journal of theoretical biology*, 265(2):185–93.
- Serra, R., Villani, M., Graudenzi, A., and Kauffman, S. A. (2006). On the distribution of small avalanches in random boolean networks. In Ruusovori, P. and et al, editors, *Proceedings of the 4th TICSP workshop on computational systems biology*, pages 93–96. Tampere: Juvenes print.
- Serra, R., Villani, M., Graudenzi, A., and Kauffman, S. A. (2007). Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *Journal of Theoretical Biology*, 246(3):449–460.
- Shmulevich, I., Kauffman, S. A., and Aldana, M. (2005). Eukaryotic cells are dynamically ordered or critical but not chaotic. *PNAS*, 102(38):13439–13444.
- Turner, A. P., Caves, L. S., Stepney, S., Tyrrell, A. M., and Lones, M. A. (2017). Artificial epigenetic networks: automatic decomposition of dynamical control tasks using topological self-modification. *IEEE transactions on neural networks and learning systems*, 28(1):218–230.
- Turner, A. P., Lones, M. A., Fuente, L. A., Stepney, S., Caves, L. S., and Tyrrell, A. M. (2013). The incorporation of epigenetics in artificial gene regulatory networks. *BioSystems*, 112(2):56–62.
- Villani, M., Barbieri, A., and Serra, R. (2011). A dynamical model of genetic networks for cell differentiation. *PLoS one*, 6(3):e17703.
- Villani, M. and Serra, R. (2013). On the dynamical properties of a model of cell differentiation. *EURASIP Journal on Bioinformatics and Systems Biology*, 2013(1):4.
- Waddington, C. H. (1957). *The strategy of the genes*. London: Allen and Unwin.

A reaction-diffusion model for simulating the oscillatory expansion of biofilms

Taishi Mikami¹, Munehiro Asally^{2,3,4}, Takeshi Kano¹ and Akio Ishiguro¹

¹Research Institute of Electrical Communication, Tohoku University, Sendai, Miyagi, Japan 980-8577

²School of Life Sciences; ³Warwick Integrative Synthetic Biology Centre;

⁴Bio-Electrical Engineering Innovation Hub, University of Warwick, Coventry, United Kingdom CV4 7AL

Correspondence: mikamin@iec.tohoku.ac.jp

Abstract

Collective dynamics is a behavior of living systems that can improve their survivability in harsh and complex environments. Towards improving the vulnerability of engineering systems against power-source limitations, we focused on an oscillatory-growth dynamics of *Bacillus subtilis* biofilms. We developed a minimal reaction-diffusion model that captures the essence of the bacterial growth, nutrient consumption and electrical signalling. Numerical simulation of the model successfully recapitulated the oscillatory dynamics of bacterial biofilms. Thus, our model provides a first step forward towards designing biofilm-inspired engineering systems such as swarm robots and power supply networks.

Collective dynamics enables a variety of living systems to survive under unpredictable and complex natural environments where available food sources are limited (Carter and Wilkinson (2015)). Such collective dynamics could offer excellent novel bio-inspired designs to improve the survivability of engineering systems. To this end, we argue that extracting simple and abstracted mathematical structure underlying the collective dynamics is important for developing robust engineering systems that can survive in severe and unpredictable environments.

We focus on the oscillatory expansion of biofilm as a model case of collective dynamics. Biofilms are structured communities of microbes which can be found on almost any surfaces; e.g. a slime in the kitchen sink. It has been shown that *B. subtilis* biofilms can exhibit oscillatory dynamics of colony growth, which is coordinated by bacterial electrical signalling and a metabolic co-dependence between interior and peripheral cells (Liu et al., 2015; Prindle et al., 2015). Importantly, this collective growth oscillation improves the survivability of biofilms as a whole when exposed to antibacterial chemicals.

While the models proposed in the previous studies well describe the physiological processes of electrical signalling or growth oscillation (Liu et al., 2015; Prindle et al., 2015), designing engineering systems based on these findings requires further integration and simplification of the core design principle of biofilm collective dynamics. In this study,

we developed a unified mathematical framework that accounts for both biofilm growth oscillation and electrical signalling, using simple reaction-diffusion equations (Turing (1990)).

Based on the findings of the previous literature (Liu et al., 2015; Prindle et al., 2015), we considered three key elements of the biofilm oscillation: namely, bacterial density $v(r, t)$, nutrient $u(r, t)$ and electrical signal $z(r, t)$. Electrical signalling is mediated by potassium waves, which depolarize cells and consequently suppress the nutrient uptake (Prindle et al., 2015). We simplified the model by not distinguishing the biofilm-interior cells from the peripheral cells. Since the biofilm dynamics appeared symmetric to all radial directions (Liu et al., 2015), we simulated the model in one-dimensional environment. The time evolutions of each elements are described by reaction-diffusion equations as follows:

$$\dot{v}(r, t) = -k_v v + D_v \cdot \nabla^2 v + k_f v \cdot f(k_1(u - k_2 z - \beta)), \quad (1)$$

$$\dot{u}(r, t) = -k_u u + D_u \cdot \nabla^2 u + u_{in} - k_f v \cdot f(k_1(u - k_2 z - \beta)), \quad (2)$$

$$\dot{z}(r, t) = -k_z z + D_z \cdot \nabla^2 z + k_{vz}(1 - f(k_h(u - \alpha))), \quad (3)$$

where $k_v, k_u, k_z, k_f, k_1, k_2, k_{vz}, k_h, \alpha$, and β are positive constants, D_v, D_u , and D_z denote the diffusion coefficients, u_{in} denotes an external input. The function f denotes a sigmoid-function $f(x) = \frac{1}{1+e^{-x}}$. The first and second terms of Eq.(1)-(3) denote the decay and diffusion, respectively. The third term in Eq.(1) represents the bacterial activities to uptake nutrients from their external environments. This becomes larger with greater nutrient availability $u(r, t)$ and larger cell density $v(r, t)$. Electrical signalling, $z(r, t)$, has an opposite impact to this term: larger z give rise to smaller values. The fourth term of Eq.(2) shows $u(r, t)$ decreases due to the uptake of nutrient by cells. The third term of Eq.(3) represents that bacterial cells efflux potassium ions under starvation (lower $u(r, t)$).

The simulation was performed under the following initial

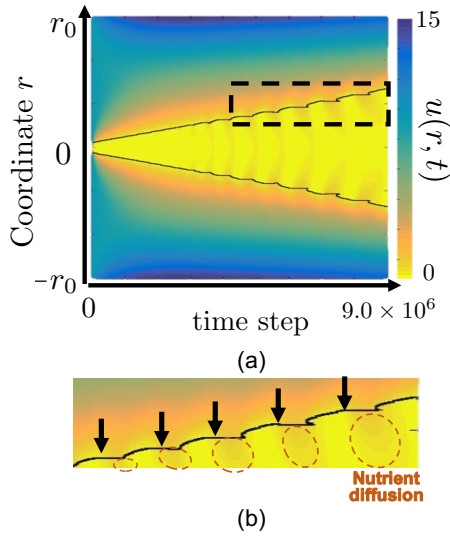


Figure 1: Simulation results (a) Spatio-temporal plot of $u(r, t)$. Black lines denote the contour lines of $v(r, t) = 9.7682$. (b) Enlarged view of the framed area in Fig.1(a)

conditions:

$$v(r, 0) = \begin{cases} v_0 & (|r| \leq s_0) \\ 0 & (s_0 < |r| < r_0) \end{cases} \quad (4)$$

$$u(r, 0) = u_0 \quad (|r| \leq r_0) \quad (5)$$

$$z(r, 0) = 0 \quad (6)$$

where v_0 , s_0 , and u_0 are positive constants. The simulation space was bounded at $r = \pm r_0$, and the Neumann boundary condition was used. To mimic the experimental condition in the previous work (Liu et al., 2015), an external source of nutrient is given at both ends of the simulation space.

$$u_{in} = \begin{cases} u_1 & (|r| = r_0) \\ 0 & (otherwise) \end{cases} \quad (7)$$

where u_1 is a positive constant. The parameter values employed in the simulation were $k_v = 0.01$, $k_u = 0.1$, $k_z = 10$, $k_f = 5.5$, $k_1 = 100$, $k_2 = 90$, $k_{vz} = 5.5$, $k_h = 300$, $\alpha = \beta = 2.1$, $D_v = 10$, $D_u = 4000$, $D_z = 5000$, $v_0 = 50$, $u_0 = 10$, $s_0 = 10$, $r_0 = 300$, $u_1 = 300$, and $dt = 0.000001$.

Fig.1(a) shows spatio-temporal plots of $u(r, t)$. $v(r, t)$ and the low value area of $u(r, t)$ expand synergistically, and exhibit an oscillation (movie: https://youtu.be/1Y_cp_rwEFI). An enlarged view of cell density over space and time (Fig. 1(b)) clarifies the oscillation: the expansion periodically halts (indicated by black arrows). These results demonstrate that our mathematical model successfully recapitulated the biofilm oscillatory expansion. Note that

the oscillation disappeared when parameters changed. For example, when k_f is large/small, the nutrition u remained small/large and does not oscillate.

The plot of $u(r, t)$ in Fig.1(b) showed that the nutrients diffuse into the biofilm as the biofilm edge stops expanding. This result can be explained by the following mechanism. Biofilm expands as bacterial cell density increases by consuming nutrients. This results in nutrient depletion in interiors, which triggers the electrical signalling that inhibits the nutrient uptake. When the electrical signal diffuses and reaches to the periphery, the peripheral cells stop taking up nutrient, which allows the diffusion of nutrient to the interior. The interior cells which get nutrient stop the electrical signalling and the signal disperses gradually. Then, the biofilm expands again. This process well agrees with the mechanism proposed in the previous works (Liu et al., 2015; Prindle et al., 2015); thus, our model well captures the essence of biofilm oscillation.

In conclusion, we proposed a simple mathematical framework that captures both biofilm oscillation and electrical signalling. The key of the proposed model is that individuals exposed to nutrient depletion emit signals so that other individuals relieve the situation by refraining from taking up nutrients. The mathematical framework for this altruistic strategy marks an important step towards designing highly survivable engineering systems, beyond biofilms. We believe that this study may lead to new control schemes for various engineering systems such as swarm robotics (Barca and Sekercioglu, 2013) and power supply networks (Kantamneni et al., 2015).

Acknowledgements

This work is in part supported by the Royal Society International Exchanges grant to MA (IEC/R3/183114).

References

- Barca, J. C. and Sekercioglu, Y. A. (2013). Swarm robotics reviewed. *Robotica*, 31(3):345–359.
- Carter, G. G. and Wilkinson, G. S. (2015). Social benefits of non-kin food sharing by female vampire bats. *Proceedings of the Royal Society B: Biological Sciences*, 282(1819):20152524.
- Kantamneni, A., Brown, L. E., Parker, G., and Weaver, W. W. (2015). Survey of multi-agent systems for microgrid control. *Engineering applications of artificial intelligence*, 45:192–203.
- Liu, J., Prindle, A., Humphries, J., Gabalda-Sagarra, M., Asally, M., Dong-yeon, D. L., Ly, S., Garcia-Ojalvo, J., and Süel, G. M. (2015). Metabolic co-dependence gives rise to collective oscillations within biofilms. *Nature*, 523(7562):550.
- Prindle, A., Liu, J., Asally, M., Ly, S., Garcia-Ojalvo, J., and Süel, G. M. (2015). Ion channels enable electrical communication in bacterial communities. *Nature*, 527(7576):59.
- Turing, A. M. (1990). The chemical basis of morphogenesis. *Bulletin of mathematical biology*, 52(1-2):153–197.

Analyzing Evolution of Avian Influenza using detailed Genotypic and Antigenic Models and Phylodynamic Simulation

Liam Mosley¹ and Dhananjai M. Rao¹

¹Miami University, Oxford, OHIO 45056. USA.
raodm@miamiOH.edu

Abstract

Avian Influenza Viruses (AIV), specifically H5N1, are highly adaptive and mutate continuously throughout their life-cycle. The accumulation of constant mutations causes antigenic drift, leading to the spread of epidemics which result in billions of dollars in socioeconomic losses each year. Consequently, the containment of AIV epidemics is of vital importance. Computational approaches to the study of epidemiology, such as phylodynamic simulations, enhance *in vivo* analysis by examining the impact of ecological parameters and evolutionary traits, as well as forecasting the rise of future variants. We propose an improvement on existing phylodynamic simulation models through the introduction of: ❶ actual Hemagglutinin (HA) protein sequences, ❷ simulating mutations, ❸ and implementing an amino-acid level antigenic analysis algorithm to model natural selection pressure. In contrast to prior approaches that use abstract antigenic models, our method uses and yields actual HA strains enabling robust validation and direct application of results to inform vaccine design. We assess the validity of our method against the current World Health Organization (WHO) H5N1 nomenclature phylogram for 3 countries. Our calibration and validation experiments use > 10,000 simulations with 1,000s of different parameter settings requiring over 2,500 hours of computing time. Our results show that our calibrated models yield the expected evolutionary characteristics but with a compromise of ~10× longer simulation times.

Introduction

Avian Influenza Viruses (AIVs), specifically H5N1 serotype, cause billions of dollars of socio-economic losses every year. Endemic in multiple species of waterfowl, H5N1 transmits both directly between hosts as well as indirectly via environmental contamination. Influenza strains that fall under the subtype H5N1 are able to spread to poultry, in turn causing widespread devastation. One of the more prominent examples of its impact was between the years of 2014-2015 where over 45 million chickens and turkeys were culled in order to stop the spread of a major epidemic (Giridharan and Rao, 2016).

AIV epidemics are perpetuated by continuous change to the nucleotide structure of the protein haemagglutinin (HA) which defines the receptor shape on the surface of influenza

viruses. Small changes to the protein structure are introduced over time, accumulating into larger changes that drastically morph the shape of the receptors on AIVs. The accumulation of mutations in phylogenetic code is called antigenic drift, and is the primary source of epidemics.

Challenges with current *in vivo* methods

There are a variety of approaches used in the containment of AIVs such as livestock isolation, vaccination of at-risk populations, and culling of infected hosts. Vaccinations are the primary method used to prevent epidemics, allowing recipients to gain immunity against the most common influenza strains in their region (WHO, 2012). Vaccine design in respect to *in vivo* analysis involves the collection of viral data from infected hosts, tracking host migration patterns, and sequencing collected viral data in order to make informed decisions on the prevalence of different AIV strains. This process can be lengthy in regards to the evolutionary time line of AIV epidemics, requiring 10 to 18 months to finalize analysis. The demand for new strain selection dictates vaccine candidates to be identified every 6 to 8 months. Another drawback to this methodology is that the analysis is reactionary and does not allow analysts to predict future epidemics. Limitations also arise due to the spatial and temporal locality of surveillance and sampling. Hence, *in vivo* analysis is ineffective alone when it comes to informing H5N1 containment efforts.

In silico approaches & shortcomings

Computational analysis methods enhance *in vivo* efforts by providing a platform for predictive modeling with results delivered in the span of days or weeks. Of particular interest are phylodynamic simulations (discussed in detail in Section [Background and related works](#)) which enable exploration of the effects of selection pressure, ecological parameters, and regional factors on the spread of viruses to forecast future epidemics. Future forecasts are of particular importance as they can be used to inform time lines for design of new vaccines and validate containment measures (Bedford et al., 2012; Giridharan and Rao, 2016; Volz et al., 2013).

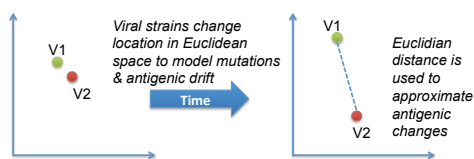


Figure 1: Overview of the current state-of-the-art antigenic modeling approach

The evolutionary and antigenic models used in phylodynamic simulations play a critical role in enabling effective modeling and analysis. Current state-of-the-art phylodynamic simulations merely use an abstract 2-dimensional or multidimensional space to represent evolutionary changes as summarized in Figure 1. Mutations in viral strains are modeled by changing their associated coordinate values. The euclidean distances are then used to approximate phylogenetic differences and their ensuing antigenic differences.

Shortcomings of state-of-the-art: The contemporary genetic and antigenic modeling approaches shown in Figure 1 have several shortcomings, including:

1. They do not model the actual viral strains. Consequently, mutations are grossly approximated.
2. Due to abstract nature, the mutation rates in the simulation do not directly reflect mutation rates observed in nature, such as those reported by Dang et al. (2010).
3. The antigenic characteristics are also approximated. Consequently, all mutations are deemed equally significant, a stark contrast to actual antigenic behaviors that primarily arise from mutations to epitope regions.
4. Since the phylodynamic simulations cannot output actual viral strains, forecasting to inform vaccine design is not straightforward.

The aforementioned shortcomings significantly limit the applicability of current phylodynamic methods.

Proposed enhancements: Our contributions

In this study we propose and assess an alternative antigenic model. It addresses the aforementioned shortcomings of the Euclidean model (shown in Figure 1) via the following three improvements, namely:

1. We propose to use actual HA sequence(s) instead of the abstract Euclidean model, starting with the root HA sequence (A/turkey/England/5092/1991) corresponding to the root of the WHO H5N1 nomenclature (WHO, 2012).
2. We simulate realistic mutations based on observed mutation rates in nature as reported by Dang et al. (2010). However, the mutation rates are further calibrated to characterize phylogenetic diversity in a given region.

3. Antigenic diversity is measured using an amino-acid level comparison algorithm, called P-Epitope proposed by Gupta et al. (2006).

This paper presents a detailed overview of our proposed enhancements in [Methods](#) section. [Experiments & Validation](#) section presents results from experiments conducted to calibrate and verify our model enhancements. In addition, results from sensitivity analysis are also discussed to identify influential parameters in the model. [Conclusions](#) presents concluding remarks along with a summary of our envisioned future work.

Background and related works

Phylodynamic models are used to characterize the epidemiological and evolutionary characteristics of viruses (Volz et al., 2013). Computational phylodynamic simulations typically use agent-based models in conjunction with discrete time simulation. Simulations enable analysis of the interplay between ecological processes and viral phylogenies. Figure 2 represents an abstract view of the ecological process that our phylodynamic simulations recreate. Waterfowl hosts are seeded with an initial viral strain and the virus then begins to mutate. For up to 8 days the virus will be shed from infected individuals, with the potential to infect not only other waterfowl but also the environment the host has contact with (Wibawa et al., 2014). Water sources are particularly vulnerable and can harbor infections for up to 20 days (Roche et al., 2014). Host immunity prevents hosts from acquiring a new infection if the virus strain is antigenically similar to a recent previous infection. The mutations that occur within individual hosts accumulate over time, causing antigenic drift. Antigenic drift causes new viral strains to diverge from their ancestral lineage, enabling them to escape host immunity and cause new infections. Figure 2 summarizes this process and exemplifies how new virus lineages diverge into new clades, or groupings, of viruses.

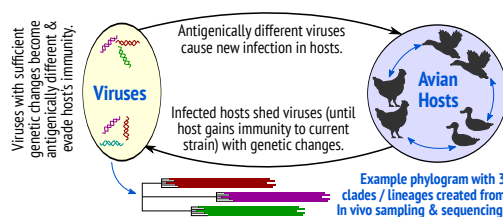


Figure 2: Ecological model of the influenza life cycle

The current leading practices for phylodynamic modeling were introduced by Gog and Grenfell, who utilize the classical model for epidemiological studies (Gog and Grenfell, 2002). This model is based on classic compartmental epidemic models, such as: susceptible (S), exposed (E), infective (I), and recovered (R). However, for avian influenza which is endemic in waterfowl, we use a S-I-S model. Hosts

that are susceptible have not been in contact with a specific viral strain but can potentially be infected. A viral strain can infect a host only if an antigenically similar strain is not present in the immune history of a host. A host remains in the infective category until the infection has run its course. While in the infective compartment, the host can spread the virus to susceptible hosts that it comes into contact with. In addition, an infective host also contaminates its environment by shedding the virus. After the host acquires immunity against a viral strain they transition back to the susceptible state and the cycle repeats.

PhySim: A phylodynamic simulator

PhySim is the computational implementation of the ecology of avian influenza shown in Figure 2. PhySim is an adaptation of a general simulation tool named Antigen (Bedford et al., 2012). As discussed by Giridharan et al. (Giridharan and Rao, 2016), PhySim’s enhancements include: ① enabling simulation of avian influenza strains, ② simulation of multiple species with different birth and death rates; ③ births occurring only during specific brooding seasons rather than throughout the year; ④ genetic and antigenic properties of viruses are independently modeled; ⑤ antigenic distances between simulated HA strains are estimated using the cross-immunity approach proposed by Gog et al.; ⑥ phylogenetic trees are constructed based on genetic differences rather than difference in emergence times; and ⑦ infection rates and infective periods account for seasonal variations in the countries.

Written in Java, PhySim uses Gillespie’s Stochastic Simulation Algorithm (SSA) with Tau-Leap optimization. In order to simulate epidemic progression with sufficient accuracy, PhySim uses a time step of 0.1. PhySim uses an Individual-Based Model (IBM) for modeling epidemic progression. PhySim only moves hosts between the susceptible and infective compartments to form an S-I-S model where the exposure of infections is simulated by probability at the point of contact and hosts move directly to infective, once the infection dies off in the host they are returned to the susceptible compartment. An S-I-S model is used due to the endemic nature of H5N1 in waterfowl, the Exposed compartment is modeled as a transition from S to I as infections immediately take hold in the host. Hosts become immediately susceptible to new infections after recovery as the R compartment becomes the transition from I back to S. Figure 3 represents a broadened view of how the S-I-S and ecological models for our simulation interact.

We are able to analyze the impact of a variety of parameters summarized in Table 1. Hosts represent a group of waterfowl from a specific species, where multiple species can be present in each simulation. We can target specific countries and model the spread and mutation of an influenza virus for that country by setting parameter values specific for the waterfowl found in the region. Nigeria and Turkey

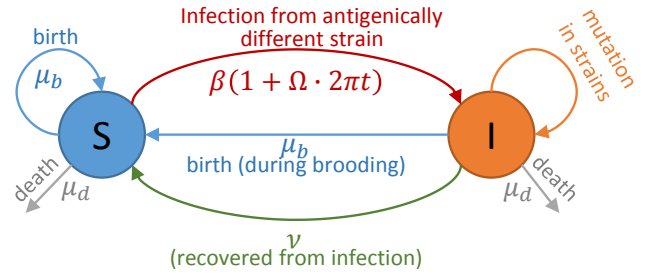


Figure 3: Ecological and SIS Modeled with example parameters from PhySim

were identified as high risk countries using a combination of Phylodynamic and Phylogeographic analysis and will be the focus of our attention (Giridharan and Rao, 2016). A point of interest in our model is that we take seasonal temperature fluctuation into consideration when determining transmission of potential infections. By using a sinusoidal curve as a modulation factor we can increase the chance of infection in colder seasons and decrease the chance of transmission during warmer months.

μ_b	Species specific daily birth rate during brooding season
μ_d	Species specific daily death rate derived from lifespan
β	Contact rate (direct) between hosts for a region
Ω	Sinusoidal seasonal modulation factor
ψ	Average daily phenotypic mutation rate
ν	Inverse of infectious period

Table 1: A sample of PhySim ecological parameters for a multi-species simulation model

The models for phylodynamic simulations can be validated by setting ecological parameters such as those in table 1 to produce phylograms that mirror those constructed from in vivo analysis. By examining the parameters that produce the best matching phylograms we are able to deduce what factors play the most impact on the spread of AIVs and can succinctly inform vaccine design. Simulations are run with a burn-in period of 15 years to simulate the time leading up to current day, by stepping past the burn-in period simulations are able to then effectively predict what the evolutionary landscape will look like in the future. Parameters such as contact rate can be abstractly represented as live-stock isolation, and features such as average mutation rate can be mapped to vaccination efforts.

Comparisons with recent related works

Current phylodynamic simulations (Bedford et al., 2012; Giridharan and Rao, 2016) model AIV strains as abstract 2D-vectors representing evolutionary data. Euclidean distance between two vectors represents phylogenetic distance between two viral strains, we will call this the geometric

approach (Gog and Grenfell, 2002). Antigenic similarity is merely approximated using Euclidean distances as summarized in Figure 1.

Our research proposes to improve upon the current modeling standards used in phylodynamic simulations by relying on a new measure of antigenic distance called P-Epitope (Gupta et al., 2006). P-Epitope has been shown to have a higher correlation to vaccine efficacy when compared to other measures of antigenic distance such as P-Sequence which is the current measure used by the WHO (Gupta et al., 2006).

In order to utilize P-Epitope as a measure of antigenic distance our representation of AIV strains in PhySim has been enhanced. We propose to implement amino acid sequencing and use an amino acid substitution model to represent competing viral strains and adjust our mutation model to reflect the current 2D-vector approach. Our work is distinguishable from recent state-of-the-art in a variety of ways. Instead of distancing ourselves from biological functions we instead embrace the computational complexity of working with protein sequence data in order to derive a simulation model that better reflects vaccine efficacy and natural mutation. The amino acid substitution model we chose to implement in our approach has been shown to be a top performer in regards to predicting future phylograms using machine learning approaches (Dang et al., 2010). This work is similar to ours in that we are also striving to predict future phylograms, but our work distinguishes itself from the machine learning approaches explore by (Dang et al., 2010) in that we are producing HA protein sequences in a more organic way through selection pressure, environmental influence, and relying on strain comparisons using a method that is more closely related to vaccine efficacy.

Methods

PhySim is progressed on a daily basis, and actions are controlled used a time step value (e.g. delta=0.1 means a day is divided into 10 time steps). The daily rate of contact and mutation are defined through input parameters. Simulation runs are conducted to match with WHO H5N1 nomenclature – *i.e.*, starting with 1991, with 15 years of burn-in time to produce strains for 2006–2010 for constructing phylograms.

Hosts are introduced and removed from both the susceptible and infective compartments at the same rate on a daily basis in order to maintain a stable population for each waterfowl species. The birth and death rates account for abundance and lifespans of different high-risk waterfowl species (that are endemic to a given region), including – *A.Acuta* (Northern Pintail), *A.Crecca* (Common Teal), *A.Fuligula* (Tufted Duck), *A.Penelope* (Eurasian Wigeon), *L.Canus* (Common Gull), *L.Limosa* (Black-tailed Godwit), *P.Pugnax* (Ruff), and *V.Vanellus* (Northern Lapwing).

Antigenic model enhancements

PhySim was originally adapted to simulate changes in the nucleotide structure of H5N1 HA protein sequences. One of the major assumptions of this model is that changes in protein sequences are uniform and random, this is known to not be the case under in vivo analysis and is a limitation to the original geometric model. FLUModel, an amino acid substitution matrix derived from a database of currently spreading H5N1 strains, is our solution to this limitation (Dang et al., 2010). Given a parameter value for t a substitution matrix can be calculated from an instantaneous rate of change matrix and steady state vector for amino acids in HA proteins. FLUModel was derived from the same set of H5N1 protein sequences that we are looking to recreate.

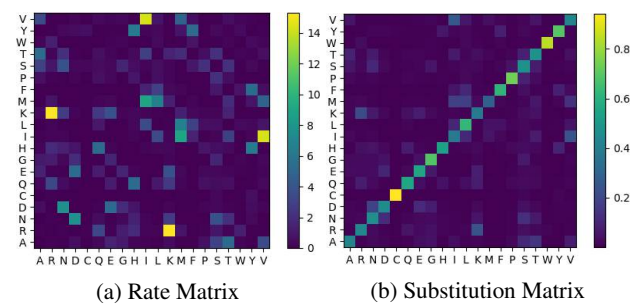


Figure 4: Using matrix exponentiation figure (a) is transformed into figure (b), this example is done using a large value for t to show the differences in substitution rates

Figure 4 illustrates the transformation from a rate of change matrix to a substitution matrix. Given the rate of change matrix q and a steady state matrix π we can get the ensuing substitution matrix $P(t)$ using the following calculations:

$$q_{xy} = \pi r_{xy}, q_{xx} = -\sum_{x \neq y} q_{xy}$$

$$P(t) = e^{tQ}$$

A value of $t = 1.0$ represents the substitution rates for all amino acids over the course of an entire branch. The substitution rates have been shown to be exceptionally accurate for small values of t .

Algorithm’s 1 and 2 exemplify how the two methods of mutation differentiate between the geometric and P-Epitope simulation models. Algorithm 1 is the current geometric substitution model, a percentage of the infective population has their associated infection mutated approximately once per day based on the time step delta value. The mutation rate ψ was computed from in vivo sequence analysis, the direction of the mutations are controlled using sin and cosine waves (Cattoli et al., 2011).

Algorithm 2 is our proposed substitution model. Its assumed that amino acid substitutions are independent and generally time reversible. That is to say in one example mu-

tation there could be multiple amino acid substitutions that occur, and their substitution rates are mutually exclusive.

Algorithm 1 2dMutate(delta)

```

1:  $I = \Sigma \text{species}.I$ , for all species in model
2:  $I' = \text{poisson}(I * \text{delta})$ 
3: while  $I' > 0$  do
4:    $i = \text{getInfected}(\text{uniform}(I))$ 
5:    $v = i.\text{getVirus}()$ 
6:    $\theta = \text{uniform}(2\pi)$ 
7:    $v.\text{trait}_X = v.\text{trait}_X + \psi * \cos(\theta)$ 
8:    $v.\text{trait}_Y = v.\text{trait}_Y + \psi * \sin(\theta)$ 
9:    $I' = I' - 1$ 
10: end while

```

Algorithm 2 subMatrixMutate(delta)

```

1:  $I = \Sigma \text{species}.I$ , for all species in model
2:  $I' = \text{poisson}(I * \text{delta})$ 
3: while  $I' > 0$  do
4:    $i = \text{getInfected}(\text{uniform}(I))$ 
5:    $v = i.\text{getVirus}()$ 
6:   for  $\text{aminoAcid} \in v.HASequence$  do
7:      $\text{substitute}(\text{aminoAcid}, \text{uniform}(1))$ 
8:   end for
9: end while

```

This mutation model allows us to represent actual changes in protein structure over the course of the simulation. More importantly it enables the use of P-Epitope to measure antigenic distance between competing viral strains.

Algorithm 3 canInfect(virus v, susceptibleHost s)

```

1:  $\text{minRisk} = 1 - \text{homologousImmunity}$ 
2:  $\text{maxRisk} = \text{homologousImmunity}$ 
3:  $\text{risk} = 0.0$ 
4: for  $v_i \in s.\text{immuneHistory}$  do
5:    $\text{distance}(v, v_i)$ 
6:   if  $\text{distance} < \text{risk}$  then
7:      $\text{risk} = \text{distance}$ 
8:   end if
9: end for
10:  $\text{risk} = \min(\text{maxRisk}, \text{risk})$ 
11:  $\text{infectFlag} = \text{uniform}(1) < \text{risk}$ 
12: return  $\text{infectFlag}$ 

```

The method of determining if an infection occurs in a host is described in Algorithm 3. After contact has been established the distance between the virus and every virus in the hosts immune history is calculated. If a uniform random number generated at the time of contact is less than the risk associated with the immune history then the host is infected.

It is here that we propose to use P-Epitope to determine the risk factor of a potential infection. As shown in the research conducted by Gupta et al. there are various examples of past vaccination regimes failing due to strain selection relying on P-Sequence (Gupta et al., 2006). Had P-Epitope been used to compare strains for vaccine selection a more

successful vaccination regime could have been promoted in many of the examples cited. Gupta et al. showed that there is a higher correlation between P-Epitope and vaccine efficacy than other measures of antigenic distance when examining past vaccination regimes.

Algorithm 4 pEpitope(virus v1, virus v2)

```

1:  $pEpitope = 0$ 
2: for  $\text{epitope} \in \text{epitopeRegions}$  do
3:    $\text{localDifference} = 0$ 
4:   for  $\text{residue} \in \text{epitope}$  do
5:     if  $v1[\text{residue}] \neq v2[\text{residue}]$  then
6:        $\text{localDifference} = \text{localDifference} + 1$ 
7:     end if
8:   end for
9:    $\text{difference} = \text{localDifference} / \text{epitope.size}$ 
10:  if  $\text{difference} > pEpitope$  then
11:     $pEpitope = \text{difference}$ 
12:  end if
13: end for
14: return  $pEpitope * pEpConv$ 

```

P-Epitope is described in detail in Algorithm 4. The epitope regions to be compared can be fed into PhySim as a parameter, there are five major epitope regions in HA protein sequences A, B, C, D, & E where a recent survey identified which residues can be attributed to respective epitopes (Peng et al., 2014). A scalar value is attached to P-Epitope to obtain a parabolic risk function. The advantage of calculating antigenic distance using P-Epitope is that we are able to identify antigenic similarity between virus strains that other measures of antigenic distance would overlook. Figure 5 illustrates the area of overlap that other measures, such as P-Sequence, are unable to detect. As seen in the Figure there is significant overlap between the frequency distributions when using P-Epitope to compare strains from the same clade and strains from different clades which were grouped using P-Sequence. These are similarities that measures such as P-Sequence can not detect.

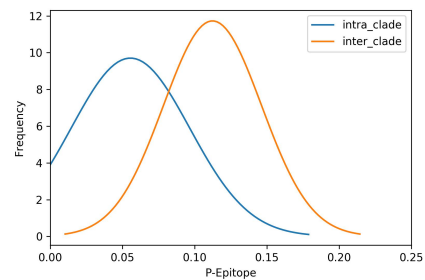


Figure 5: A sample of 100 H5N1 HA protein sequences from 10 different clades were compared using P-Epitope, frequency of P-Epitope values for inter and intra -clade distances is plotted

Experiments & Validation

We calibrated the model with respect to the following simulation outputs: ① Infective and susceptible populations for each time step, ② Number of clades in resulting phylogram, ③ antigenic diversity. The outputs of the simulation using the new model were compared against the previously calibrated geometric model.

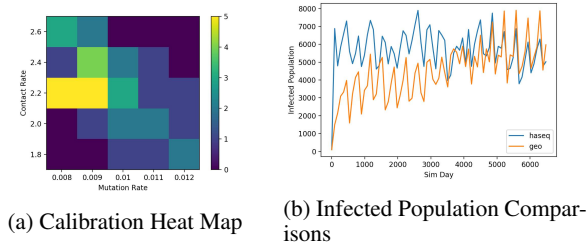


Figure 6: A window of the calibration efforts for PhySim with the antigenic model implemented

The starting point of the calibration effort consisted of estimating the average mutation rate for the new model based on the average nucleotide mutation rate used in the geometric model. The expected amino acid substitution rates were approximated to within 1% of the nucleotide substitution rates. This was done by adjusting the t parameter value for the substitution matrix, generating probabilities for each amino acid substitution and multiplying by the average number of each amino acid in a typical HA protein sequence.

Figure 6 illustrates an example of how the simulation model was validated for Turkey. The majority of parameter settings were kept consistent between the geometric model and our new model. In order to properly calibrate and test the new model only the mutation and contact rates were adjusted. Subfigure 6a shows a narrow window of calibration settings that produced consistent results, a more exhaustive calibration effort was conducted to find this window that required hundreds of different parameter combinations ranging the contact rate from 1.0 to 3.0 and the mutation rate from 0.002 to 0.10 in varying step sizes. The window illustrated represents the success rate of parameter combinations with a contact rate between 1.8 and 2.6 in steps of 0.2, and mutation rates between 0.008 and 0.012 in steps of 0.001.

Each simulation was seeded with slightly mutated root sequence variants equivalent to the number of initial infected individuals. Due to the discrete steps mutations take in the new model the initial propagation period is susceptible to low mutation rates, and can cause the number of infected individuals to zero out early. This was combated by spawning the slight variants, ensuring what is the equivalence of 100 simulation days of mutations. This results in the initial spike of infective individuals in subfigure 6b.

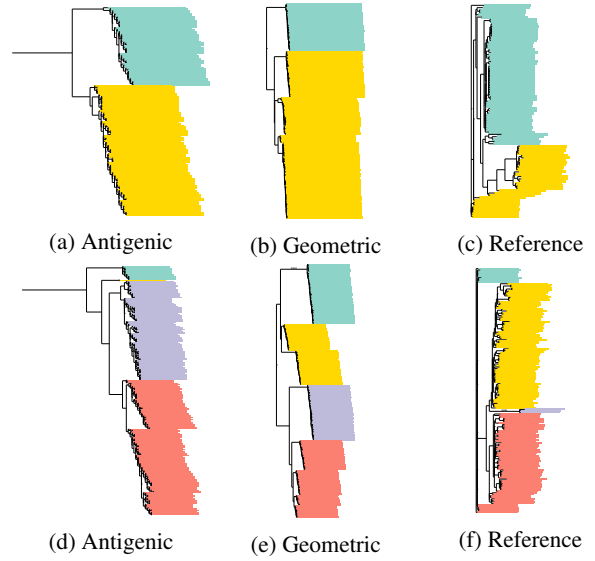


Figure 7: Antigenic phylograms were produced using our enhanced model (success is 2 clades for Turkey, 4 for Nigeria). Subfigures(a-c): Turkey. Subfigures(d-f): Nigeria

Parameter analyses

In this study we have used Generalized Sensitivity Analysis (GSA) (Güven and Howard, 2007) to assess the influence of parameter settings in our model. GSA is based on a two-sample Kolmogorov-Smirnov Test (KS-Test) and yields a $d_{m,n}$ statistic that is sensitive to differences in both central tendency and any differences in the distribution functions of parameters. The $d_{m,n}$ statistic ($0 \leq d_{m,n} \leq 1.0$) is the maximum separation between cumulative probability distributions observed in a two-sample KS-Test. The $d_{m,n}$ statistic is computed for each parameter by varying its value over a $\pm 25\%$ range, in steps of 10%, around its calibrated setting as shown in Figure 8. At each setting, 10 stochastic simulations are conducted and the number of successful (*i.e.*, simulation produces phylogram with same number of clades as reference *in vivo* phylogram) and unsuccessful outcomes are recorded. We have used the model for Turkey to conduct the sensitivity analysis.

The data is used to compute the cumulative probability of success and failure for each parameter as shown in Figure 8. The maximum difference between the cumulative success and failure probabilities is the $d_{m,n}$ statistic shown in red for each parameter. For example, from Figure 8, the $d_{m,n}$ statistic for Contact rate (β) is 0.256. Figure 9 shows a summary comparison of the influence of the parameters. The lightly shaded bands show the 95% Confidence Intervals (CI) computed using standard bootstrap approach using 5000 replications with 1000 samples in each.

As illustrated by the GSA $d_{m,n}$ statistic values, the following parameters do not have a strong influence on the model's characteristics – the initial population of birds (N),

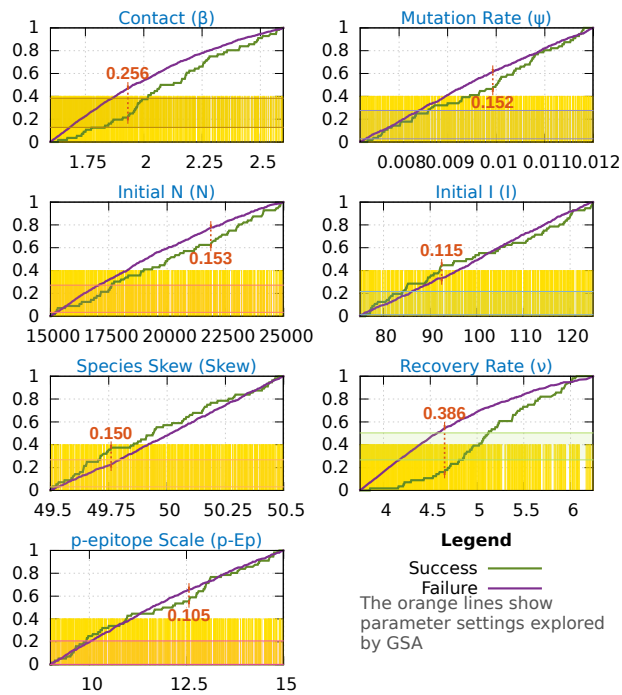


Figure 8: Detailed results from Generalized Sensitivity Analysis (GSA). The x-axis in each sub-chart indicates range of values for each parameter. In all sub-charts the y-axis is the $d_{m,n}$ statistic.

initial number of infected birds (I), variance in the abundance of different species (Skew), and the antigenic scaling parameter used with p-epitope. In other words, assumptions made about the values of these parameters do not have a significant impact on the validity and outcome of our analyses. Insensitivity to these assumed parameter values is an important aspect of our model. It enables us to draw inferences with sufficient confidence without requiring to have a good estimate of waterfowl populations, waterfowl species abundance, initial infections etc.

On the other hand, the most influential factors that primarily drive diversity of viral strains are: recovery rate (ν) at 0.386, contact rate (β) at 0.256, and mutation rate at 0.152. The recovery rate for H5N1 has been set to the putative value of 5 days. Accordingly, the two key parameters whose values have been determined via calibration are β and ψ , which are specific to each region being analyzed. These three influential parameters are also the primary targets for containment and prophylaxis efforts.

Correlation analysis: Consistent with interrelationships in nature, the parameters in the model have inherent correlations as illustrated by the correlogram in Figure 10. The correlogram has been plotted using results from successful configurations, *i.e.*, parameter settings that yield the correct number of clades, *i.e.*, the same number of clades as in the

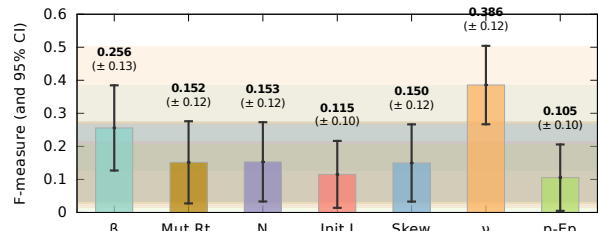


Figure 9: Parameter comparisons based on GSA

reference phylogram. The correlogram has been plotted using R and the PerformanceAnalytics package.

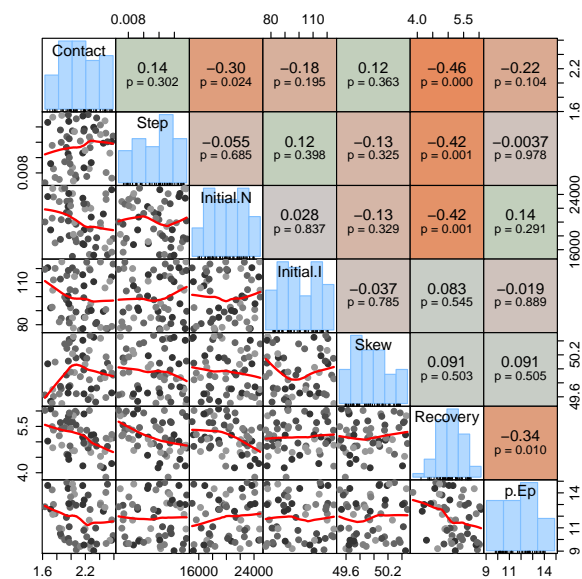


Figure 10: Correlation between parameters elicited by GSA

The correlogram shows that the recovery rate ($1/\nu$) is strongly, negatively correlated to contact rate (β), mutation rate (ψ), p-epitope scale (p-Ep), and initial population (N). These negative correlation are expected because of the nature of these parameters. For example, decreasing mutation rates (ψ) but increasing recovery time (ν) essentially maintains the antigenic diversity. Similarly, increasing contact rate (β) enables more infections to occur and hence, even with decreasing ν , overall antigenic diversity is maintained. The correlogram shows that the recovery rate ν plays a central role in anchoring other epidemiological and ecological parameters in the model. This observation also emphasizes the need for surveillance and assay-based identification of at least one of these four parameters in emergent epidemics and other parameters can be estimated via phylodynamic simulations.

Conclusions

Vaccinations are widely used to contain and mitigate epidemics caused by antigenic variants of Avian Influenza Viruses (AIVs), including the H5N1 serotype. However, vaccines need to be regularly updated to compensate for antigenic drift in AIVs. Currently, expensive *in vivo* assays are required to regularly update vaccines to compensate for antigenic drift. Furthermore, such *in vivo* assays and analyses do not provide insight into the underlying ecological processes that is necessary to inform containment and prophylaxis strategies. Consequently, *in vivo* methods are enhanced using computational or *in silico* approaches involving phylodynamic simulations. The antigenic models used for phylodynamic simulations play a critical role in overall effectiveness of *in silico* methods. Current state-of-the-art models merely use an abstract multidimensional space to approximate both genetic and antigenic changes.

This paper proposed and evaluated a novel antigenic model which is distinguished from current research by: ❶ use of actual Hemagglutinin (HA) protein sequences, ❷ simulating mutations occurring to the HA sequence(s) and further calibrating the mutation rates to mirror ecological niches, and ❸ and implementing an amino-acid level antigenic analysis algorithm. The paper discussed the motivation for the aforementioned enhancements and presented an algorithmic overview. The models were verified and validated using over 10,000s of simulations with 1,000s of different parameter settings and requiring over 2,500 hours of compute time. We assess the validity of our method using the current World Health Organization (WHO) H5N1 nomenclature for Turkey and Nigeria.

The influence and impact of parameters in our model has been explored via Generalized Sensitivity Analysis (GSA). Our GSA analysis showed that recovery rate (ν), contact rate (β), and mutation rate (ψ) strongly influence the antigenic diversity. Correlation analysis revealed a strong, negative correlation between recovery rate (ν) and contact rate (β), mutation rate (ψ), p-epitope scale (p-Ep), and initial population (N). This correlation emphasizes the need for surveillance and assay-based identification of at least one of these four parameters in emergent epidemics. Once a putative value for one of the parameters is identified, the other parameters can be estimated via phylodynamic simulations.

This study lays the groundwork for using detailed antigenic models in phylodynamic simulations. A key issue that we encountered was the high computational times for the simulations. Currently, we are exploring solutions to reduce the computational times.

Nevertheless, we contend that the benefits accrued from our methods offset the higher computational times. The significance of this research is that not only are we able to inform containment efforts similar to the current state-of-the-art, we also produce actual HA protein sequences that can be used in different methods of analysis in the future. As an example, with a fine-tuned model there is the possibility to

explore and monitor evolutionary characteristics and niches of avian influenza viruses. Unlike analysis done using current state-of-the-art models direct connections between clusters of viruses in our simulations to real world clades can be made, and the direct impact on containment efforts on the structure of real world avian influenza viruses will be able to be examined.

References

- Bedford, T., Rambaut, A., and Pascual, M. (2012). Canalization of the evolutionary trajectory of the human influenza virus. *BMC Biology*, 10(1):1–12.
- Cattoli, G., Fusarola, A., Monnea, I., Covenb, F., Joannisc, T., El-Hamidd, H. S. A., Husseine, A. A., Corneliust, C., Amaring, N. M., Mancina, M., Holmesh, E. C., and Capuaa, I. (2011). Evidence for differing evolutionary dynamics of A/H5N1 viruses among countries applying or not applying avian influenza vaccination in poultry. *Vaccine*, 29:9368–9375.
- Dang, C. C., Le, Q. S., Gascuel, O., and Le, V. S. (2010). Flu, an amino acid substitution model for influenza proteins. *BMC Evolutionary Biology*, 10(1):99.
- Giridharan, N. and Rao, D. M. (2016). Eliciting characteristics of h5n1 in high-risk regions using phylogeography and phylodynamic simulations. *Computing in Science Engineering*, 18(4):11–24.
- Gog, J. R. and Grenfell, B. T. (2002). Dynamics and selection of many-strain pathogens. *Proceedings of the National Academy of Sciences*, 99(26):17209–17214.
- Gupta, V., Earl, D. J., and Deem, M. W. (2006). Quantifying influenza vaccine efficacy and antigenic distance. *Vaccine*, 24(18):3881 – 3888. 3rd International Conference on Vaccines for Enteric Diseases.
- Guyen, B. and Howard, A. (2007). Identifying the critical parameters of a cyanobacterial growth and movement model by using generalised sensitivity analysis. *Ecological Modelling*, 207(1):11 – 21.
- Peng, Y., Zou, Y., Li, H., Li, K., and Jiang, T. (2014). Inferring the antigenic epitopes for highly pathogenic avian influenza h5n1 viruses. *Vaccine*, 32(6):671 – 676.
- Roche, B., Drake, J. M., Brown, J., Stallknecht, D. E., Bedford, T., and Rohani, P. (2014). Adaptive evolution and environmental durability jointly structure phylodynamic patterns in avian influenza viruses. *PLoS Biol*, 12(8):e1001931.
- Volz, E. M., Koelle, K., and Bedford, T. (2013). Viral phylodynamics. *PLoS Computational Biology*, 9(3):e1002947.
- WHO (2012). Continued evolution of highly pathogenic avian influenza A (H5N1): updated nomenclature. *Influenza and Other Respiratory Viruses*, 6(1):1–5.
- Wibawa, H., Bingham, J., Nuradji, H., Lowther, S., Payne, J., Harper, J., Junaidi, A., Middleton, D., and Meers, J. (2014). Experimentally infected domestic ducks show efficient transmission of indonesian h5n1 highly pathogenic avian influenza virus, but lack persistent viral shedding. *PLoS ONE*, 9:e383417.

Anomalies in the Behaviour of a Modularity Inducing Problem Domain

Zhenyue Qin, Tom Gedeon and RI (Bob) McKay

Australian National University, Canberra ACT 2600, Australia
rimanucs@gmail.com

Abstract

Espinosa-Soto and Wagner (2010) introduced a domain with weak assumptions on biology and environment, where modular structures emerge under simple evolutionary processes.

We found a number of anomalous behaviours: modularity emerged in this domain, but could not dominate populations as observed in biology. Highly fit, modular solutions exist in the search space, can be readily found by a simple deterministic procedure (and presumably could dominate populations if found), but evolutionary search never found them, despite mutation biases that appear to favour those solutions. Moreover, emergence of modularity was promoted by stochastic dynamicity in the fitness function: a stochastic but fixed fitness function generated much less modular solutions.

Introduction

Adaptability is an essential property of both biological and artificial evolutionary systems (Yang et al., 2013). Biological organisms have already evolved it through modularity (Gerhart and Kirschner, 2007), giving hope for artificial systems to also generate modular, adaptable systems (Pfeifer and Scheier, 2001).

Lack of modularity is a key factor limiting scaling of artificial biological systems to higher complexity (Kashtan and Alon, 2005; Pfeifer and Bongard, 2006). Artificial neural networks are usually densely connected (Jain et al., 1996), where brains have modules taking different responsibilities – hippocampus for novel situations, amygdala for emotional controls (Coward, 2013). Thus it is important to understand the conditions under which modularity spontaneously emerged through biological evolution. Engineers may leverage them to generate modular systems, while understanding may help to winnow the evolutionary theories of biology.

Formally, modularity is the division of structures or functions into sub-units that perform autonomously (Schlosser and Wagner, 2004). A module is a group of elements which associate preferentially within the group (Newman, 2006; Espinosa-Soto and Wagner, 2010). Many biological activities and structures can be modeled as networks – animal brains, signaling pathways, etc. (Barabasi and Oltvai, 2004).

A modular network can be partitioned into highly connected components, with only sparse connections between them (Freeman, 1977; Clune et al., 2013). Elements within a module preferentially undertake coherent functions independent of outside elements (Espinosa-Soto and Wagner, 2010; Larson et al., 2016). Such modules appear everywhere in biology (Coward, 2013), at multiple levels of organisation (Espinosa-Soto and Wagner, 2010; Coward, 2013). Modularity can promote the evolvability of organisms, i.e. the ability to rapidly adapt to novel environments (Pigliucci, 2008). Modular networks allow changes in one module without disturbing others; and modular structures can be reused and recombined to perform new functions (Espinosa-Soto and Wagner, 2010; Wagner and Altenberg, 1996).

Despite decades of research into modularity (Wagner et al., 2007), there is no consensus on its biological origin (Wagner and Mezey, 2004; Espinosa-Soto and Wagner, 2010). Three theories stand out, as their preconditions may commonly arise in nature (Wagner et al., 2007): modularly-varying evolutionary goals (Kashtan and Alon, 2005), biological parsimony pressures (Clune et al., 2013), and specialisations in gene activity patterns (Espinosa-Soto and Wagner, 2010). In the first, modular changes in environments generate an impetus toward modularity (Kashtan and Alon, 2005). Organisms whose environmental sub-components change repeatedly show more modularity than those from stable environments (Parter et al., 2007). Fluctuations are omnipresent in real environments (Espinosa-Soto and Wagner, 2010; Yachi and Loreau, 1999). However it is unclear to what extent these fluctuations are modular (Espinosa-Soto and Wagner, 2010). While links in networks often incur costs, as Clune et al. assert, it is less clear that the cost is so uniform across the many forms of biological networks as to fully account for modularity's ubiquity.

Gene regulatory networks (GRNs) commonly regulate to preserve specific gene activation patterns against external disturbance – the target pattern may differ over time or location (Jones and Taylor, 1980). Espinosa-Soto and Wagner (2010) suggested this may promote modularity, and defined a GRN abstraction to test it. In the model, there

was initially a single target; over evolutionary time, additional modularly-structured targets were added. They used a mutation-only (crossover-free) evolutionary algorithm. Modularity was observed to emerge, though in contrast to biological systems, it was not seen to dominate populations. Subsequently, this work was extended by Larson et al. (2016), who examined the effects of different recombination mechanisms. Among other results, they first demonstrated an evolutionary impact from crossover hotspots, a phenomenon in which we were already interested.

We originally intended to extend this work, concentrating on two widespread biological phenomena, diploidy and crossover hotspots. To establish a baseline, we experimented with variants of standard genetic algorithms, yielding anomalous and difficult-to-explain results. This paper details some further results of our explorations, revealing further anomalies and leading to deep questions about our intuitions on both the structure of this problem, and biological evolutionary landscapes in general.

Methods

We use genetic algorithms as our evolutionary simulation tools. The GRN domain was originally proposed by Wagner (1996) and customised by Espinosa-Soto and Wagner (2010) and Larson et al. (2016). All simulation code was implemented in Java 1.8.0 and Python 2.7.10. Modularity was evaluated using the NetworkX package with the community API (Hagberg et al., 2008).

Model

Cells in an organism display heterogeneity in functionalities and morphologies, yet generally contain the same genes. This heterogeneity primarily arises from differing gene expression profiles resulting from differing gene-gene regulation (repression or activation). Cells interpret the same genetic material in different ways so that their behaviors and structures vary. These distinct interpretations are due to regulation, among other mechanisms via the activation and repression of genes by other genes. Such a GRN may be abstracted as a weighted directed graph with the weights limited to +1 (activation) and -1 (repression) (Wagner, 1996).

We used the Espinosa-Soto and Wagner (2010) representation: a GRN with N genes is represented as an N^2 adjacency matrix $A = a_{ji}$ with $a_{ji} \in \{-1, 0, 1\}$ with 0 representing independence of gene i from gene j . The gene activity pattern of this network at time t is a Boolean row vector $s_t = [s_t^0, \dots, s_t^{N-1}]$. Gene i can either be active ($s_t^i = 1$) or inactive ($s_t^i = -1$). The state transition is modeled by:

$$s_{t+1} = \sigma\left[\sum_{j=1}^N a_{ji}s_t^j\right] \quad (1)$$

where $\sigma(x)$ equals 1 if $x > 0$ and -1 otherwise. For a more detailed explanation and justification of the model, please

refer to the above paper.

Fitness

Biological GRNs are commonly able to maintain specific activation states in the face of random external perturbations (Aderem, 2005). The abstraction by Espinosa-Soto and Wagner (2010) generated a set of P perturbations of the target, with each gene having 0.15 probability of mutating to the opposite state (they used $P = 500$, Larson et al. (2016) used $P = 300$). The GRN was recursively applied to each perturbation. Preliminary experiments indicated that it normally took fewer than 20 transitions to reach an attractor (Wagner, 1996). In that case, the Hamming Distance D between attractor and target state was returned; otherwise the maximum possible Hamming distance D_{max} was returned. In either case, the value $\gamma_i = (1 - D/D_{max})^5$ was computed for each perturbation i , with $1 \leq i \leq P$. Finally, the mean value $\bar{\gamma}$ over all γ_i was used to compute the fitness of the GRN g over a particular target t as:

$$f_t(g) = 1 - e^{-3\bar{\gamma}} \quad (2)$$

This process, of randomly sampling a set of perturbations of the target, and evaluating the GRN's ability to robustly return them to the target, was repeated each generation.

In stage one, the system evolved to regulate only the first target state. Subsequently, the fitness function rewarded regulation of newly introduced states, while maintaining pressure to regulate earlier ones, computing the overall fitness $f(g)$ as the arithmetic mean of $f_t(g)$ over all targets t .

We followed the strategy of Espinosa-Soto and Wagner (2010) (please see their paper for fuller detail), using only two targets: evolving for 500 generations with one target, then adding the second for a further 1500 generations. We based our choice of the number of perturbations (75) on a trade-off between the observation of Totten (2015) that 75–100 perturbations are sufficient for emergence of modularity, and the practical need to minimise runtime.

Larson et al. (2016) applied a different approach to evaluating the fitness of networks, by sampling a static (but larger) set of perturbations at the beginning of each run, and using this same set of perturbed targets whenever network fitness was calculated. This method has important computational cost advantages, since the fitness value of a given GRN on a given target remains fixed from generation to generation, so that caching and hashing methods can be used to give substantial speedups. However it converts the original stochastically dynamic fitness evaluation into a static, deterministic one. This has potential implications for search.

Evolutionary Simulations

Espinosa-Soto and Wagner (2010) imposed a mutation bias towards networks with a relatively low link density. A node u in the network has a probability $\mu = 0.05$ to mutate every

generation; if it does, it either loses or gains an interaction. The probability for u to lose an interaction is:

$$p(u) = \frac{4r_u}{4r_u + N - r_u} \quad (3)$$

where N is the number of genes in the target activation pattern, and r_u is the number of regulators of gene u (Espinosa-Soto and Wagner, 2010), i.e. the number of incoming edges. The probability an interaction is $1 - p(u)$. This bias acts to preserve the sparseness of the network, which computational biology research suggests is necessary for modularity to emerge.

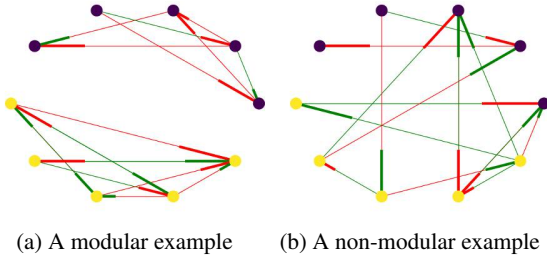


Figure 1: Modular and Non-modular networks. Different node colors represent distinct modules (based on the modular changes in target values). Green/red edges mean activation/repression.

Espinosa-Soto and Wagner (2010) defined modules as components of targets that followed similar activation histories. In the most-used example, with two targets of length ten, the activations of the first five locations in both targets were identical, while the activations of the second five were inverted between the targets. Thus the modules treated as the connected components in the GRN involving nodes 1–5 and nodes 6–10. Figure 1 shows typical examples.¹

Espinosa-Soto and Wagner (2010) used no crossover; Larson et al. (2016) used horizontal crossover, exchanging blocks of rows: when matrices A_1 and A_2 cross over at index i , the sub-matrices $A_1[0 : i - 1, :]$ and $A_2[0 : i - 1, :]$ remain unchanged, while the remainders are exchanged. However it ignores the diagonal symmetry of modules (if a_{ij} is in a module, then so is a_{ji}). We defined a ‘diagonal crossover’ using a diagonally symmetric interchange: given a random crossover point $[i, i]$, we preserve the sub-matrices $A_1[0 : i - 1, 0 : i - 1]$ and $A_1[i : 9, i : 9]$ (and also for A_2), while exchanging the rest. Compared with horizontal crossover, as Figure 2 illustrates, this should better preserve community structure.

¹These color conventions are used throughout this paper. While color in the images conveys additional information, the key distinctions are still observable in black and white.

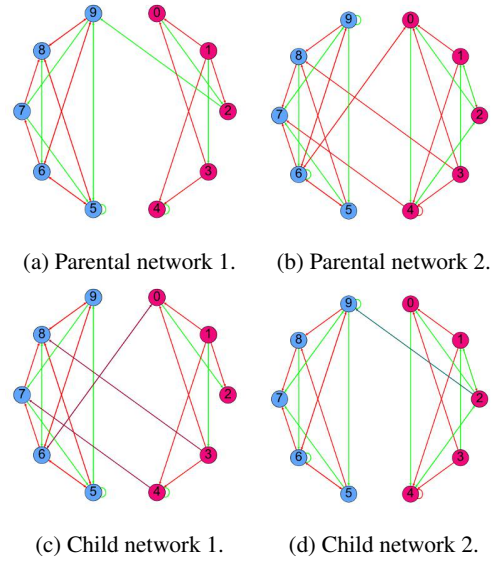


Figure 2: Illustration of diagonal crossover

Modularity Metric

In most of this paper, we used the Q modularity metric defined in Newman and Girvan (2004). This measures whether there are more inter-module edges than would be expected from the total number of edges. Formally:

$$Q = \sum_i^K \left[\frac{l_i}{L} - \left(\frac{d_i}{2L} \right)^2 \right] \quad (4)$$

Q falls in the range $[-\frac{1}{2}, 1)$, the upper bound depending on the number of modules ($\frac{1}{2}$ for two, $\frac{3}{4}$ for three).

Preliminary Experiments: Modularity Surprises

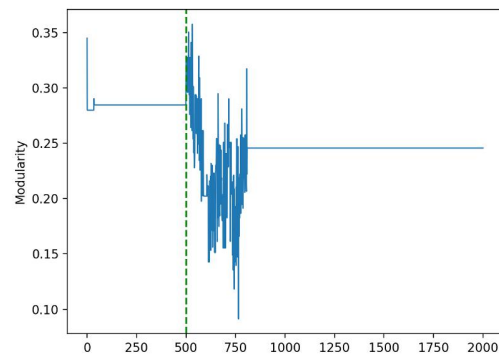


Figure 3: Modularity decreased after target change (marked by the vertical line).

Our baseline-setting experiments, using a standard genetic algorithm with an elite of 10 and tournaments of size 3, revealed surprising differences in the emergence of modularity from the results of Espinosa-Soto and Wagner (2010) and Larson et al. (2016). Recall that in their experiments, overall modularity increased after a second target was added to the fitness function. In our initial experiments (using the Louvain metric (Blondel et al., 2008) rather than the Q metric we use elsewhere), we instead observed a decrease immediately following the addition of the second target, with the overall modularity eventually stabilising to a level substantially below that of the first phase (see Figure 3).

Our settings differed from the previous work in the following ways:

1. Use of crossover (difference from the former only)
2. Tournament instead of proportional selection
3. Incorporation of elitism
4. Omission of the age–fitness Pareto mechanism (difference from the latter only, (Bongard, 2017))
5. Use of the Louvain metric

Of these differences, item 1 seems unlikely to explain our result since Larson et al. also used crossover, while items 3 and 4 both increase the relative eagerness of our search. The Louvain and Q metrics measure closely related properties, so seemed unlikely to be the cause. Item 2 is more complex, since a tournament of size 3 exerts relatively weak selection pressure, but the relative pressure of tournament and proportional selection varies with the stage of the algorithm. Proportional selection depends on relative differences in fitness, so it typically exerts fairly strong pressure in early stages of search, but as the population fitness converges and differences reduce, pressure weakens; by contrast, tournament selection, being dependent only on fitness rank order, exerts a relatively constant selection pressure throughout. In particular, when populations are relatively converged (as at the time of the target switch), we would expect even relatively small tournaments to be more selective than proportional.

Based on the above, we decided to test the joint effects of elitism and tournament selection. The results bore out this hypothesis: the same algorithm and settings, with elitism eliminated and proportional substituted for tournament selection, led to the emergence of modularity, see Figure 4.

Experiment Settings

Tables 1 and 2 show the gene activity patterns and the essential parameters of our evolutionary simulations. Unless otherwise specified, all experiments used the stochastic fitness evaluation of Espinosa-Soto and Wagner (2010). The only parameters that varied from the tables are the selection

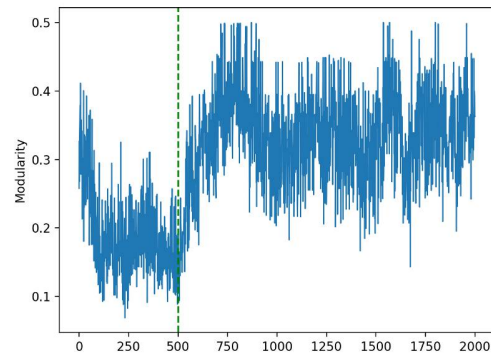


Figure 4: Without elitism, modularity increased after target change (marked by the vertical line).

Table 1: Gene Activity Patterns

1	
Target Pattern	Generation where Pattern is added
+1 -1 +1 -1 +1 -1 +1 -1 +1 -1	0
+1 -1 +1 -1 +1 +1 -1 +1 -1 +1	500

type (tournament) and size, and the elite size. The detailed explanations of these parameters are given in Table 3.

The evaluation metrics for experiments are the eventual fitness values and Q scores from the last generation. All are significance tested using Wilcoxon’s Signed-Rank Test.

Experiments and Results

Diagonal Crossover Promotes Modularity

We ran trials comparing horizontal, diagonal and no crossover, in all cases without elitism. As Tables 4 and 5 show, diagonal crossover generated significantly higher fitness and Q score than horizontal crossover, which in turn generated significantly higher Q score, though non-significantly lower fitness, than absence of crossover.

This Boolean model was proposed in Wagner (1996), who showed that random recombination made no difference to evolution of stability. Our experiments suggest that more structured forms of recombination (which occur in biology) can contribute to evolvability. Diagonal crossover can preserve underlying network modules. Although horizontal crossover did not preserve community structures as well as diagonal, its partitioning is still based on a modular structure, and thus partially preserves modularity.

Greediness Reduces Modularity

Elitism Reduces Modularity

Table 2: Parameters of the Evolutionary Simulations

Edge Size	Number of Perturbations	Per-location Perturbation Rate
20	75	0.15
Mutation Rate	Population Size	Selection Type
0.05	100	Proportional
Reproduction Rate	Maximum Generation	Elite Size
0.9	2000	0 or 10
Trials per Treatment	Significance Test	
40	Wilcoxon Signed Rank	

Elitism increases greediness of search by deterministically retaining the fittest individuals. We compared an elite of 10 with no elite, finding significantly lower fitness and modularity in the former case (Tables 6 and 7).

Comparing Best Fitness and Modularity between Proportional Selection and Tournaments of Various Sizes

Tables 8 shows final generation best fitness increased with decreasing tournament size, with proportional selection falling between tournaments of sizes 2 and 3, though none of these differences reached significance (Table 9). Modularity showed a similar pattern, although with proportional selection yielding lower modularity than tournaments of size 3; however only the differences with tournaments of size 10 were significant.

Further Analysis and Discussion

Emergence rather than Dominance

Modularity in biological networks is remarkably robust; specific GRN modules cross not merely species boundaries, but are shared across kingdoms, having survived billions of years. They do not merely appear in populations, they are generally common to all individuals in a species (Schlosser and Wagner, 2004). By contrast, modularity in this problem domain is a delicate flower. It does not appear at all if selection pressure is strong, and its emergence is heavily dependent on evolutionary details; not merely the dynamic fitness variation emphasised in the problem definition, but the details of whether stochasticity in the fitness evaluation is static or dynamic (see below).

What should we make of this? One possible reaction is to follow Clune et al. (2013) and ascribe biological modularity emergence to direct linkage costs, so that further inquiry is unnecessary. However our further confirmation that this

Table 3: Explanations of simulation parameters

Target Patterns	patterns that are perturbed, and towards which gene regulatory networks evolve
Target Addition Generations	the generations where new targets are introduced
Edge Size	the initial number of edges in the gene regulatory network
Perturbation Number	the number of perturbed versions of each gene activity pattern
Perturbation Rate	the expected proportion of corrupted genes in a pattern
Mutation Rate	the probability of a GRN node to gain or lose an interaction
Population Size	the number of individuals in the population
Selection Type	the type of selection used, and where tournament, the size of the tournament
Reproduction Rate	the proportion of the population reproduced without change, vacancies being filled by the selection mechanism
Maximum generation	the generation when the simulation will terminate

Table 4: Final Generation Best Fitness and Q Score with No, Horizontal and Diagonal Crossover

	No Crossover	Horizontal	Diagonal
Fitness	0.9476	0.9446	0.9488
Q Score	0.1961	0.2919	0.3386

domain does promote the emergence of modularity, albeit somewhat fitfully, deserves further investigation.

We looked more deeply at the diagonal crossover data of Table 4; we assumed that the data might reflect some fitness advantage from modularity (i.e. that over many runs, fit but highly modular individuals might be fitter than their non-modular cousins). So from each diagonal crossover run, we collected the fittest individuals among those that had the highest modularity value, and conversely, the least modular among those that had the highest fitness. Averaged over all runs, the latter were fitter than the former (Table 10) – the less modular networks could more robustly recover the unperturbed pattern. Thus the failure of the modular networks to dominate was less surprising: they could not do so because fitter, non-modular ones would take over.

We wondered whether this could arise from insufficient complexity in the targeted gene activity patterns. The num-

Table 5: Wilcoxon Ranked Sign Values for Table 4

	Fitness P	Q Score P
No < Horz		1.7090e-6
Horz ≤ No	0.0882	
Horz < Diag	0.0006	0.0019

Table 6: Final Generation Best Fitness and Q Score with and without Elites

	No Elite	Elite Size 10
Fitness	0.9488	0.9472
Q Score	0.3386	0.2735

ber of genes in patterns might be too simple, or the number of targets might be too few, to reflect natural environments. Perhaps modular networks might give great performance on complex tasks, but worse than non-modular ones for simple tasks. Using the basic set-up of subsection ‘Diagonal Crossover Promotes Modularity’, we ran more complicated evolutionary simulations using patterns comprising 15 nodes and three sub-patterns (for which the maximum modularity score is 0.75, rather than the 0.5 for two sub-patterns), encountering a sequence of seven different targets. Evolution was extended to 35,000 generations and during the final epoch from (26000 → 35000) generations, it was evolving to robustly recover all seven targets. We repeated the preceding analysis; the results in Table 11 resemble those of Table 10. Overall, the number and complexity of targets could not resolve the issue: less modular networks still recovered the target more robustly than more modular networks. However the differences in modularity were smaller, suggesting that modularity dominance might emerge with sufficiently complex problems – perhaps beyond the bounds of computational feasibility with current techniques.

Are Inter-Module Edges Critical to High Fitness?

Finally convinced that inter-module edges were essential to robust target recovery, we decided to validate this by an extreme measure. From the 40 runs in the first experiment of this section, we extracted the least modular individuals among those having the maximum fitness value found in the run, yielding 40 fairly non-modular but highly fit individuals. We then simply removed all inter-module edges. Naturally the resulting networks were perfectly modular, but we expected them to be highly unfit. This expectation was borne out in 16 cases. But in the majority (24) this crude

Table 7: Wilcoxon Ranked Sign Values for Table 6

	Fitness P	Q Score P
Elite 10 < No Elites	0.0003	0.0022

Table 8: Final Generation Best Fitness and Q Score for Proportional Selection and Different Sized Tournaments

	Proport	Tourn Size 2	Tourn Size 3	Tourn Size 10
Fitness	0.9488	0.9404	0.9404	0.9371
Q Score	0.3386	0.3697	0.3623	0.2783

Table 9: Wilcoxon Ranked Sign Values for Table 8

	Fitness P	Q Score P
Proportional > Tourn Size 2	0.7401	0.0467
Tourn Size 3 < Size 2	0.9313	0.7881
Tourn Size 10 < Size 3	0.0164	0.0015
Tourn Size 10 < Proportional	0.0227	0.0054

operation resulted in an *increase* in fitness. Figure 5 shows a real example of this process. The ‘Before removal’ GRN had a fitness of 0.9472. After removing $\approx 20\%$ of its edges to make it completely modular, its fitness rose to 0.9502. So the inter-module edges were not merely inessential to the GRN’s fitness, they were an impediment. Which leaves the question: since the resulting completely modular, very high fitness solutions were available to the search algorithm, why could it not find them?

Originally, we suspected this anomaly might result from the lower edge density of the ‘after removing’ networks – perhaps they were too much below the edge density targeted by the biased mutation operator, so that this soft constraint eliminated them from search. Further investigation revealed that on average, they still retained approximately 30 edges, which is above the targeted density of the mutation operator (equation 3), so far from being difficult to reach, the biased mutation operator favored moving toward them.²

In order to further understand this phenomenon, that

²We found this result so surprising that we suspected a bug in our implementation of the mutation bias. We turned off selection completely, giving every individual the same fitness. The average size stabilised at a little over 22 edges (because of asymmetry in the mutation distribution - a GRN must have at least zero edges, but has a finite probability of more than 40 - a small deviation above the mutation target of 20 is unsurprising).

Table 10: Modularity dominance for data underlying Table 4

Generation Range	Modularity	Fitness
(500, 2000)	0.5000	0.9482
	0.1736	0.9502

Table 11: Modularity dominance for extended runs from more complex environments

Generation Range	Modularity	Fitness
(26000, 35000)	0.5506	0.9100
	0.4151	0.9419

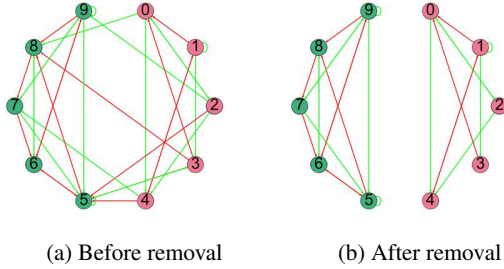


Figure 5: Illustration of inter-module connection removal.

our evolutionary simulations could not find a path to the trimmed networks, we recorded all fitness values that could be obtained by removing one inter-module edge in turn, until all have been deleted, and plotted them as graphs. Figure 6 is typical. We could usually see a steadily improving fitness as edges were deleted, along a path that was favored by the biased mutation operator, yet our genetic algorithm could not find these paths.

Fitness and Modularity benefit from Dynamic Fitness Evaluation

One important difference between the work of Larson et al. (2016) and of Espinosa-Soto and Wagner (2010) is the former’s sampling perturbations only once across a run. Thus there is a single change in the fitness function (the changed target after generation 500), while for the latter, the changing perturbation sample varies the fitness evaluation every generation. Preliminary experiments had already shown that static fitness evaluation substantially decreased both ultimate fitness and modularity (Qin et al., 2018); we decided to try to improve the performance of the static version by limiting diagonal crossover to location [5, 5] (i.e. the module boundary) – remarkably unsuccessfully, as lines 1 and 3 of Table 12 show. Unlike the fitness difference, the modularity difference was highly significant ($p = 2.6879 * 10^{-5}$).

To further understand this phenomenon, we collected the gene regulatory networks of the final generation, and mutated each network 499 times to generate neighbours. We measured the fitness values of these neighbours with the target perturbations from this generation, and determined their maximum. In this fashion, we would have 40 neighbourhood-maximum fitness values for both dynamic and static fitness evaluation. We repeated this process for

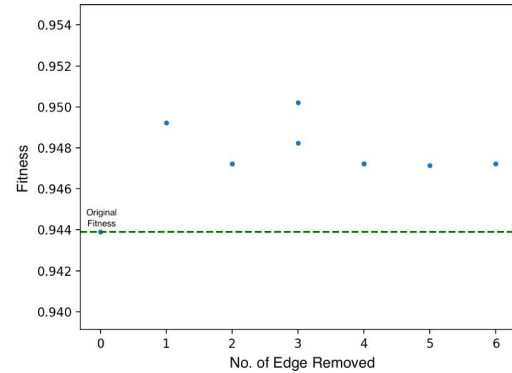


Figure 6: Fitness Values along all Inter-module Edge Removal Paths from a High Fitness, Low Modularity Network that results in Increased Final Fitness.

Table 12: Fitness and Q Scores for Neighbours of Final Generation Fittest Individuals in Static and Dynamic Environments

	Dynamic	Static
Final Fitness	0.9461	0.9323
Best Neighbour Fitness	0.9410	0.9323
Final Q	0.3374	0.1851
Best Neighbour Q	0.3791	0.2223

the modularity Q score. As Table 12 shows, the fittest individuals in the final generation for the dynamic problem were generally local optima, whereas for the static problem, they were generally on a fitness plateau, with equally fit neighbours, and a substantially lower fitness than found in the dynamic problem. In both scenarios, there were neighbours of substantially higher modularity than the original individual, but overall modularity, both of the final generation best individual and of its neighbours, were much higher in the dynamic problem than in the static. Again, fitness differences between dynamic and static were not significant, but modularity differences were ($p = 2.6956 * 10^{-5}$).

Conclusions

The modularity-inducing problem of Espinosa-Soto and Wagner (2010) is clearly important for understanding the evolution of network modularity, in that it relies on much weaker assumptions about the world that provides the evolutionary context (namely that some parts of the fitness target change independently of others) than do other explanations. It also provides potential insights into other aspects of biological evolution. The results of Larson et al. (2016) reveal

connections between modularity and recombination, and of particular importance, the first results known to us to suggest an evolutionary advantage to the ubiquitous phenomenon of recombination hotspots. Our own results suggest connections to homologous recombination, and to stochastic variation in fitness landscapes. They also reveal some puzzles: why don't modules dominate populations in this environment as they do in nature? Why can't the many different algorithms we tried find the very high fitness, completely modular solutions that we know to exist? Why does stochastic sampling generate fitter and more modular solutions than static sampling? We have ended up with more (but more detailed) questions than we started with. Clearly, further research and new analyses are needed...

Acknowledgements

Some of the outcomes / conclusions of the research in this paper were presented as a poster at GECCO 2018, and as a two-page summary in the proceedings (Qin et al., 2018). All underlying data is available at <https://github.com/ZhenyueQin/Project-Maotai-Modularity>.

References

- Aderem, A. (2005). Systems biology: its practice and challenges. *Cell*, 121(4):511–513.
- Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.
- Bongard, J. (2017). Personal communication.
- Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society B: Biological Sciences*, 280(1755):20122863.
- Coward, L. A. (2013). Towards a theoretical neuroscience. In *Towards a Theoretical Neuroscience: from Cell Chemistry to Cognition*, pages 389–393. Springer.
- Espinosa-Soto, C. and Wagner, A. (2010). Specialization can drive the evolution of modularity. *PLoS computational biology*, 6(3):e1000719.
- Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.
- Gerhart, J. and Kirschner, M. (2007). The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(suppl 1):8582–8589.
- Hagberg, A., Swart, P., and Schult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL).
- Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.
- Jones, P. A. and Taylor, S. M. (1980). Cellular differentiation, cytidine analogs and dna methylation. *Cell*, 20(1):85–93.
- Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(39):13773–13778.
- Larson, A., Bernatskiy, A., Cappelle, C., Livingston, K., Livingston, N., Long, J., Schwarz, J., Smith, M., and Bongard, J. (2016). Recombination hotspots promote the evolvability of modular systems. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 115–116. ACM.
- Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.
- Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.
- Parter, M., Kashtan, N., and Alon, U. (2007). Environmental variability and modularity of bacterial metabolic networks. *BMC evolutionary biology*, 7(1):169.
- Pfeifer, R. and Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Pfeifer, R. and Scheier, C. (2001). *Understanding intelligence*. MIT press.
- Pigliucci, M. (2008). Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82.
- Qin, Z., Gedeon, T., and McKay, R. I. B. (2018). Why don't the modules dominate? In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18*, pages 121–122, New York, NY, USA. ACM.
- Schlosser, G. and Wagner, G. P. (2004). *Modularity in development and evolution*. University of Chicago Press.
- Totten, M. (2015). Exploring the evolution of modularity in gene regulatory networks.
- Wagner, A. (1996). Does evolutionary plasticity evolve? *Evolution*, 50(3):1008–1023.
- Wagner, G. P. and Altenberg, L. (1996). Perspective: complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.
- Wagner, G. P. and Mezey, J. (2004). The role of genetic architecture constraints in the origin of variational modularity. *Modularity in development and evolution*, pages 338–358.
- Wagner, G. P., Pavlicev, M., and Cheverud, J. M. (2007). The road to modularity. *Nature Reviews Genetics*, 8(12):921–931.
- Yachi, S. and Loreau, M. (1999). Biodiversity and ecosystem productivity in a fluctuating environment: the insurance hypothesis. *Proceedings of the National Academy of Sciences*, 96(4):1463–1468.
- Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., and Karamanoglu, M. (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes.

Optimizing Radiation Therapy Treatments by Exploring Tumour Ecosystem Dynamics in – silico

Stephan Scheidegger¹ and Harold Fellermann²

¹ZHAW School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland

²Interdisciplinary Computing and Complex Biosystems Research Group, School of Computing, Newcastle University, Newcastle upon Tyne, UK
scst@zhaw.ch

Abstract

In this contribution, we propose a system-level compartmental population dynamics model of tumour cells that interact with the patient (innate) immune system under the impact of radiation therapy (RT). The resulting *in silico* - model enables us to analyse the system-level impact of radiation on the tumour ecosystem.

The Tumour Control Probability (TCP) was calculated for varying conditions concerning therapy fractionation schemes, radio-sensitivity of tumour sub-clones, tumour population doubling time, repair speed and immunological elimination parameters. The simulations exhibit a therapeutic benefit when applying the initial 3 fractions in an interval of 2 days instead of daily delivered fractions. This effect disappears for fast-growing tumours and in the case of incomplete repair. The results suggest some optimisation potential for combined hyperthermia-radiotherapy.

Regarding the sensitivity of the proposed model, cellular repair of radiation-induced damages is a key factor for tumour control. In contrast to this, the radio-sensitivity of immune cells does not influence the TCP as long as the radio-sensitivity is higher than those for tumour cells. The influence of the tumour sub-clone structure is small (if no competition is included). This work demonstrates the usefulness of *in silico* – modelling for identifying optimisation potentials.

Introduction

Cancer remains to be one of the most elusive widespread diseases – accounting for an estimated 16% of worldwide deaths.¹ In the last decades, many improvements concerning equipment and treatment planning tools have driven anti-cancer radiation therapy (RT) towards precise applications of radiation doses. Remarkable progress has been made regarding geometrical precision. In contrast to these more engineering – related aspects, the biological knowledge about growth dynamics and therapy response of tumours seems to remain behind technology development. This may be a reason for the upcoming discussion of a biologically adapted RT (Thorwarth, 2017).

One of the reasons for the comparably slow progress of biological understanding lies in the differences between the behaviour of tumour cells *in vitro*, *in vivo* (mouse model) and

in patient, which, on the one hand, prevent a direct transfer of knowledge gained by experiments to clinical treatment, and on the other hand, indicate that cancer is a systemic disease that can only be understood by treating tumours as complex systems that are intricately coupled to their host environment including the immune system.

Several studies consequently hypothesize that the major cause of radio-resistance observed during RT treatments may be related to the heterogeneity of tumour tissues (Horsman et al., 2012; Baumann et al., 2016). Under this systemic perspective, cancer might be regarded as an evolving ecosystem of diverse cell populations (different tumour sub-populations or sub-clones, host tissue, endothelial cells / vascular system, immune cells) with a dynamic behaviour influenced by the boundary conditions of RT. In such a framework, radiation-induced cell killing would constitute a selection pressure that leads to survival of radio-resistant sub-populations. Loss of competition between the different cancer sub-populations and host tissue might lead to an accelerated progress of disease. Evolutionary dynamics of cancer (Crespi & Summers, 2005) could be responsible for an adapted response of the tumour to anti-cancer therapy.

Ecological aspects of anti-cancer therapy have been discussed by different authors (Pienta et al., 2008; Basanta et al. 2015; Basanta et al., 2012). Merlo et al. (2006) considered cancer as an evolutionary and ecological process. Gatenby et al. pointed out the role of evolutionary dynamics for cancer prevention (Gatenby et al., 2010). Ecological principles have also motivated the investigation of the invasion of metastasizing cancer cells into bone marrow (Chen & Pienta, 2011). The connection between artificial life and cancer research has been pioneered by Maley and Forrest (2000) who developed an agent based model of precancerous cells that might develop into cancer by adopting mutations at atypically high rate. Population dynamics of tumours have been further studied by Gonzales-Garcia et al. (2002) and Sole (2003) who concluded through agent-based modelling that spatial genetic heterogeneity observed in tumours naturally follows from simple ecological competitor dynamics. Since then, many multi-agent models of tumour growth with increasing physical accuracy have been proposed (Abbott et al. 2006, Zhang et al. 2009, Bentley 2013, Ozik et al. 2018 and others). Yet, few of these approaches take into account the dynamics of and interaction with the tumour environment, particularly host

¹ <https://ourworldindata.org/causes-of-death>

tissue, immune system, and boundary conditions imposed by anti-cancer therapies.

Scheidegger et al. (2010) demonstrated by a combined population model *in silico* that - in certain situations - the tumour response to radiation is dominated by the radio-sensitivity of the endothelial cells. In the light of these results, there is certain evidence that tissue dynamics could play a pivotal role in development and therapy response of tumours. Scott et al. (2016) investigated the impact of spatial metrics onto the radiation efficacy using a hybrid cellular automaton model.

Despite this increased appreciation of ecological and evolutionary aspects of cancer dynamics, relatively little work has been performed that attempts to transfer well-established methodological approaches from theoretical ecosystem analysis to the domain of RT.

We here propose a high-level population dynamics model of tumour cell populations that interact with a simplified immune system under the impact of RT treatment (Fig. 1).

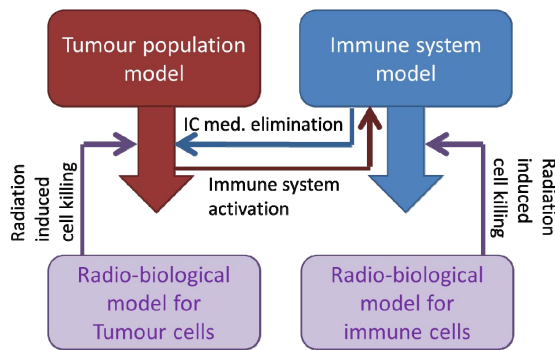


Figure 1: Structure with the different sub-models of a simplified tumour - immune system model. IC med. elimination means immune cell mediated cell killing.

Due to the sheer complexity of the biological tumour system and its response to both, the immune system and RT, we regard it as unpromising to bring the mechanistic details of the tumour ecosystem into a computer model: many of details are un-known and quantifiable experimental insight into most of the different dynamic processes is missing. Therefore, we propose a system dynamics-based phenomenological model, in order to support hypothesis generation and design of experiments *in vitro* or clinical trials. While our approach sacrifices many of the mechanistic details underlying tumour growth and treatment, the model has been developed with clinical applicability in mind: model parameters are in principle measurable and chosen to represent realistic parameter ranges, and model predictions are reported through clinically meaningful quantifiers.

By improving the systemic understanding of tumour ecosystems, quantitative *in silico* analysis of system-level cancer models could subsequently help to improve or optimize anti-

cancer treatment strategies. This broadened ecological and evolutionary view on the impact of RT onto tumours could be an important step for understanding the dynamics responsible for treatment response and may serve as a basis to optimize and improve RT treatments.

In this work, we will focus on the optimisation potential regarding fractionation schemes in External Beam Radiation Therapy (EBRT). In case of an EBRT, the radiation is applied by fractions with a dose per fraction of e.g. 2 Gy for prostatic adenocarcinoma (Fowler et al., 1995). Typically, the standard fractionation scheme for EBRT for adeno-carcinomas is a daily application of the fractions (5 days per week). There is a recent discussion about hypo-fractionation (less fractions with a higher radiation dose) for example using daily fraction doses of 2.75 Gy (e.g. non-small lung cancer; De Dios et al. 2017). Arguments for choosing a fractionated application are related to the 4 R's in RT: **R**epair (host tissue has faster repair compared to tumour cells, fractionation leads to a larger, so called "therapeutic window"), **R**epopulation (longer interval will allow tumours to repopulate), **R**e-oxygenation (reduction of tumour mass will enhance oxygenation and therefore increase radio-sensitivity of the tumour cells for subsequent fractions) and **R**edistribution (mitotic cells are more radio-sensitive and radiation-induced cell killing can cause a synchronisation of tumour cells regarding the cell cycle). Fraction size and time interval between fractions should be optimised: long intervals lead to unwanted tumour growth and complete repair of tumour cells, short intervals to less re-oxygenation and decreasing difference between cellular repair of tumour cells and host tissue (smaller therapeutic window). It has to be pointed out here that this is only a simplified explanation, not referring to the full ecosystem dynamics and excluding the aspect of immune system interaction. Exploring artificial life *in silico* could support modelling the immune system interaction and improve the understanding of the complex dynamics including pattern recognition.

Materials and Methods

The model describes tumour cell populations $T_k = T_k(t)$ interacting with an immune cell population $M = M(t)$. Both, immune cells and tumour cells are considered to be radio-sensitive. For modelling the radiation-induced elimination, a simplified Γ -LQ-model is applied (Scheidegger et al. 2011a). In this model, the elimination rate of tumour cell is determined by the radio-sensitivity constants α and β (which are considered to be population-specific) and the radiation dose rate R :

$$\left[\frac{dT_k}{dt} \right]_{Rad} = -(\alpha_k + 2\beta_k \Gamma_k) \cdot R \cdot T_k \quad (1)$$

and for the immune cells:

$$\left[\frac{dM}{dt} \right]_{Rad} = -(\alpha_M + 2\beta_M \Gamma_M) \cdot R \cdot M \quad (2)$$

The dose equivalent Γ (Transient Biological Dose Equivalent TBDE, the unit is Gray (Gy)) is rising with the dose rate R and decaying with cellular repair:

$$\frac{d\Gamma_{k,M}}{dt} = R - f(\Gamma_{k,M}) \quad (3)$$

In contrast to the full Γ -LQ-model (Scheidegger et al. 2011a), dose rate dependence is not considered (fixed dose rate) and repair is switched off during irradiation. Integration of Eq.3 leads to the well-established linear-quadratic model, where the logarithm of the surviving fraction S is given by a linear and a quadratic term of the radiation dose D : $\log S = -(\alpha D + \beta D^2)$. For repair, first order kinetics is assumed: $f(\Gamma_{k,M}) = \gamma_{k,M} \Gamma_{k,M}$. For the tumour cells (TBDE = Γ_k), two cases are examined: Complete repair - is assumed when the remaining dose equivalent (TBDE) is smaller than 1 mGy - and incomplete repair (TBDE ≥ 1 mGy).

The growth of the tumour cell populations is determined by a growth constant k_T which is here chosen to be constant for all populations (sub-clones). Tumour cells can mutate with the rate $k_{mut} T_k$ to form or join another tumour sub-clone T_{k+1} . In addition to the radiation-induced elimination, tumour cells can be eliminated by the immune cells M . The development of the population size of the tumour cell population T_k is given by:

$$\frac{dT_k}{dt} = k_T T_k + k_{mut}(T_{k-1} - T_k) - w_k k_{ine,k} T_k M - \left[\frac{dT_k}{dt} \right]_{Rad} \quad (4)$$

w_k is an immune-response specific weighting factor for the population k . Different weights can be explained by the fact that the immuno-biological sensitivity may be influenced by the mutation in two ways: (i) Higher sensitivity is gained by an increased genetic instability of tumour cells collecting more mutations and (ii) decreasing immuno-sensitivity can be achieved by an evolutionary process due to immune-system driven selection of tumour cells expressing less detectable antigens on their surface (immunoediting by immune system (Vesely et al. 2011); escape from the immune response by reduction of self-antigen presenting Class I Major Histocompatibility Complex MHC).

Tumour cells that have been eliminated from the population T_k are transformed to apoptotic or necrotic cells N_k . These cells - and with them, the amount of radiation induced Damage Associated Membrane Proteins DAMP's (Grimsley et al., 2003) - can be eliminated by apoptosis and phagocytosis:

$$\frac{dN_i}{dt} = \left[\frac{dT_i}{dt} \right]_{Rad} - k_{rie} M N_k \quad (5)$$

In this model, no explicit pathway (apoptotic or necrotic, s. discussion) is chosen. *In vivo*, apoptotic cells are removed by macrophages, whereas necrosis may be accompanied by inflammation, leading to a more complex immune response. Natural killer cells can respond in absence of self-antigen presenting MHC and may be activated by DAMP's. DAMP's such as Heat Shock Proteins HSP (Srivastava, 2002, Daugaard

et al., 2007) can be produced by ionizing radiation and are thus expressed in higher levels after RT (Nytko et al., 2019). The presented *in silico* model includes only one immune cell population. For simplicity, the activation of the (innate) immune system is assumed to be governed by the abundance of DAMP's in the tumour compartment, leading to an invasion of immune cells until an equilibrium level (or response level) M_{resp} is reached. Immune cells can be eliminated by radiation in the tumour compartment during RT. In consequence, the immune cell population in the tumour compartment is described by:

$$\frac{dM}{dt} = k_M \cdot (M_{resp} - M) - \left[\frac{dM}{dt} \right]_{Rad} \quad (6)$$

The DAMP-activation of the response level M_{resp} is assumed to be dependent on the sum of damaged cells by the following model:

$$M_{resp} = M_{min} + M_{max} \cdot \left(1 - e^{-r \sum_k N_k} \right) \quad (7)$$

M_{min} and M_{max} are limiting the amplitude of the response to a range between a minimal concentration (M_{min} per tumour compartment volume) of immune cells and a saturation level. Both levels may be dependent on patient specific immune response capability.

For clinical evaluation of the RT treatments, the Tumour Control Probability (TCP) is a well-established quantity. The TCP is calculated in the following way:

$$TCP = e^{-\sum_k T_k} \quad (8)$$

The concept behind Eq.8 is based on the fact that the description using differential equations delivers average population sizes. For small amounts of cells, statistical variation has to be taken into consideration.

In this study, EBRT was simulated by 32 fractions, each fraction with a radiation dose of 2.5 Gy. The fractionation schemes differ by the interval between the applications of RT fractions. A scheme with daily application was compared to fractionation with larger intervals of 2-4 days and a mixed scheme, where the initial 3 fractions have a spacing of 2 days, followed by daily application of subsequent fractions.

The simulations were carried out by numerical integration using a 4th order Runge-Kutta algorithm. The time increment was set to $\Delta t = 10^{-5} \text{ d}^{-1}$.

Results

Parameters used for the different simulations are summarized in Tab.1. The resulting TCP's are displayed in Fig. 2 and Fig. 3. In general, the selection of the parameter values is made in

regard to the resulting TCP: The parameter values are restricted to a range which results in a rising TCP ($0.01 < \text{TCP} < 0.99$) between a dose of 35 Gy and 80 Gy. This represents more or less the clinical observations for the selected radio-biological parameters.

Diagram No.	Parameter values	
	Fig.2	Fig.3
$\alpha_M / \text{Gy}^{-1}$	0.5	0.5
β_M / Gy^{-2}	0.2	0.2
$\alpha_k / \text{Gy}^{-1}$	varying	0.310 ($k=1$)
β_k / Gy^{-2}	varying	0.0625 ($k=1$)
γ_k / d^{-1}	10 (for all k)	varying
k_T / d^{-1}	$3.46 \cdot 10^{-2}$ except. Fig. 2d	$3.46 \cdot 10^{-2}$
k_{mut} / d^{-1}	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
k_{me} / d^{-1}	10^{-8}	10^{-8}
k_{ime} / d^{-1}	10^{-9} , varying for sub-clones	varying
k_M / d^{-1}	1	1
M_{\min}	10^6	10^6
M_{\max}	10^9	varying
r	10^{-3}	10^{-3}

Table 1: Parameter values used for the simulations: α and β for the tumour are in the upper range, especially when comparing to clinical studies but these values and the α/β -ratios are strongly varying across different patients and tumours (van Leeuwen et al., 2018); k_T is corresponding to an intrinsic tumour doubling time of 20 d (for clinical observed doubling times s. Mehrara et al., 2007).

The sensitivity of the model regarding the radio-biological parameters of the immune cells have been investigated by simulations using different α_M - and β_M - values: For simulations with the parameter set of Fig. 2, varying α_M - values do not have a significant impact on the TCP (less than 1%), as long as $\alpha_M > 0.3 \text{ Gy}^{-1}$. This is also the case for varying γ_M - values (set to 10 d^{-1} ; invading immune cells are considered as not pre-irradiated).

In a first step, the influence of different tumour cell populations on the dynamics of the system is investigated. In Fig. 2, different scenarios regarding the radio-sensitivity and immunological interactions of tumour sub-clones (5 different populations) are shown. The simulation included a phase of tumour evolution during 300 days before start of RT. The initial population size was set to 10^9 cells, leading to approx. $3 \cdot 10^{12}$ tumour cells at the beginning of the treatment. It was assumed that all populations have a doubling time of 20 days (accordingly $k_T = 3.46 \cdot 10^{-2} \text{ d}^{-1}$).

In Fig. 2a & 2b, the tumour sub-clones are considered to become more radio-sensitive with increasing mutations (due to higher genetic disorder). With every mutation, the value of α increases with an equidistant step: $\alpha_k = 0.3 + 2 \cdot 10^{-2} \cdot (k-1)$.

A similar relation was applied for the β -values: $\beta_k = 0.06 + 5 \cdot 10^{-3} \cdot (k-1)$. The values for $k=1$ represent the case of an adeno-carcinoma. From an immunological perspective, tumour sub-clones may develop in two different directions: Increasing mutations can lead to a better immunological elimination (due to the decreased expression of “don’t eat me”-signals such as CD47 or inappropriate expression of other membrane-bound signalling molecules).

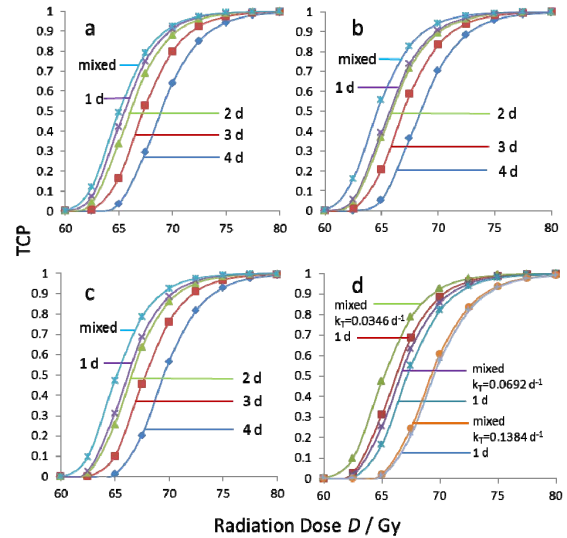


Figure 2: Tumour Control Probability TCP for a treatment of a tumour consisting on different sub-clones and different fractionation schemes (1 d = 1 day interval between fractions, 2 d = 2 days interval etc., mixed = 2 days interval between the initial 3 fractions, subsequent fractions in 1 day interval). (a) sub-clones with increasing weights w_k ; (b) sub-clones with decreasing w_k ; (c) single sub-clone scenario with average radio-sensitivity and (d) single sub-clone scenario with varying tumour doubling times.

The case with an enhanced immunological elimination for increasing k (with $k_{ime,k} = k \cdot k_{ime}$) is shown in Fig. 2a: Increased intervals between the fractions lead to a decreased tumour control (TCP-curves shifting to the right). The best tumour control can be achieved with the so called mixed protocol (first 3 fractions in an interval of 2 days, subsequent fractions daily). The explanation for this behaviour can be found by investigating the temporal development of the tumour cell population: After the first RT-fraction, a high amount of tumour cells are eliminated by radiation, leading to a high amount of apoptotic (or necrotic) cells. These cells are activating the immune response which in turn “co-eliminates” viable tumour cells. Depending on the selected invasion speed (governed by k_M), the elimination rate is low shortly after the end of irradiation, increases to a maximum after approxi-

mately 1 day and then decreases again. This effect can only be observed during the early (first to third) fractions. For the subsequent fractions, the immune system - mediated response becomes smaller due to the reduced amount of eliminated tumour cells and the radiation-induced elimination of immune cells during each irradiation. Therefore, repopulation (which is counteracting the immunological elimination) drives the outcome towards lower TCP-values, especially for larger intervals. This explains the lower tumour control for fractionation schemes having large intervals throughout the therapy course.

Decreasing immunological elimination (Fig. 2b) with increasing k ($k_{ime,k} = k^{-1}k_{ime}$) leads to similar results as in Fig. 2a. The effect of the mixed protocol is stronger compared to Fig. 2a and the difference between 1 day and 2 days interval is smaller, indicating the positive effect of immunological elimination during longer intervals at the beginning of the RT-course. It has to be pointed out here, that this result is influenced by the tumour evolution during the 300 days before therapy starts (initial conditions: $T_1(0) = 10^9$, $T_{k>1}(0) = 0$).

The comparison of the different cases (Fig. 2a & b) leads to the question how important the detailed modelling of different tumour sub-clones is. Fig. 2c shows a tumour where only one tumour population is taken into consideration. To achieve a comparable TCP, the radio-biological parameters are adapted to $\alpha_1 = 0.310 \text{ Gy}^{-1}$ and $\beta_1 = 0.0625 \text{ Gy}^{-2}$. The resulting TCP curve does not remarkably differ from the case exhibited in Fig. 2b, indicating a relatively small influence of the number of tumour cell populations considered in the model.

Since repopulation is counteracting the cell killing, simulations using different tumour doubling times (20 days, 10 days and 5 days) were carried out (Fig. 2d). The comparison of the mixed protocol with daily applied fractions reveals a disappearing therapeutic gain of the mixed protocol for fast repopulating tumours. This can be explained by the smaller decrease of the tumour cell population during the 2-day intervals compared to the case with slow repopulation. Especially during the second day of these intervals, faster repopulation compensates the immune-related elimination.

In Fig. 3, the response of one single tumour cell population onto the mixed protocol vs. daily fractions is shown for different conditions for repair and immunological elimination. The effect of incomplete repair leads to a strongly increased tumour control for both, mixed and daily fractionation. The difference between mixed and daily fractionation disappears at $\gamma = 2 \text{ d}^{-1}$. The remaining dose equivalent Γ after one day interval is 0.4 Gy, representing moderate incomplete repair. With $\gamma = 1 \text{ d}^{-1}$, the remaining dose equivalent Γ is 1.5 Gy. In this case, the mixed protocol exhibits a slightly lower TCP compared to the daily fractionation, indicating the strong influence of repair onto the outcome.

For an increased immunological elimination ($k_{ime} = 10^{-8}$) and incomplete repair (Fig. 3b), the mixed protocol reaches always higher TCP values, but the effect remains small. Fig. 3c shows the outcome for an increased immune cell density (governed by M_{max}). In this case, the effect of the mixed protocol becomes clearly larger for complete repair. For incomplete repair ($\gamma = 1 \text{ d}^{-1}$), the difference between the two fractionation schemes remains small. Interestingly, this behaviour does not change when M_{max} is increased to 10^{13} and

k_{ime} is set to 10^{-8} (Fig. 3d). The explanation for this small difference is the fact that after the first fraction of the treatment, the total number of tumour cells is reduced by over a factor 500 one day after irradiation. Therefore, subsequent fractions will not produce large amounts of apoptotic cells, resulting in a smaller activation of the immune system.

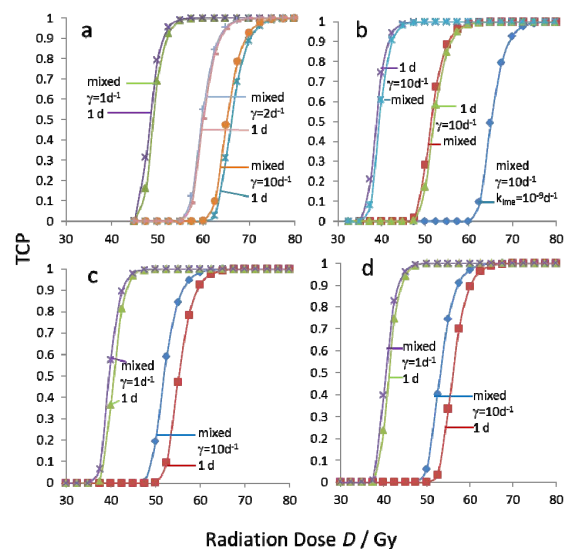


Figure 3: Tumour Control Probability TCP for a treatment of a tumour with varying repair speeds ($\gamma_k = \gamma = 1 \text{ d}^{-1}$ = complete vs. $\gamma = 2 \dots 10 \text{ d}^{-1}$ = incomplete repair) and varying immune-mediated elimination. Only mixed vs. daily fractionation is displayed. (a) $k_{ime} = 10^{-9} \text{ d}^{-1}$, $M_{max} = 10^9$; (b) $k_{ime} = 10^{-8} \text{ d}^{-1}$, $M_{max} = 10^9$; (c) $k_{ime} = 10^{-9} \text{ d}^{-1}$, $M_{max} = 10^{11}$; (d) $k_{ime} = 10^{-10} \text{ d}^{-1}$, $M_{max} = 10^{13}$.

Discussion and Conclusions

The presented model contains a simplistic approach to the tumour-immune system interaction. Therefore, the question is: What can we learn from such an artificial system *in silico*? Certainly, the presented model cannot be used as a predictive model, due to lack of any validation. Validation would require a comparison of model output to clinically observed TCP's and measuring immune cell densities during treatment.

The model includes an innate immune system type response and does not differentiate between the diverse populations of immune cells and their specific role in developing immune reaction. The interference with the adaptive immune response via antigen-presenting cells may strongly modify the tumour-specific response.

For assessing the kinetic constants, tumour-immune system interactions should be tracked during treatment to acquire time-resolved data. This is associated with big challenges even for a clinical trial and is not realistic in clinical routine. A

refinement of the immunological model would introduce a large number of additional parameters, many of them with large, patient-specific variations. To overcome these fundamental restrictions and difficulties, new approaches of modelling are needed. Looking to the immune system in its complexity, several network-structures involving signalling pathways and receptor signalling can be identified. Considering an adaptive immune system as a trainable network could lead to new insights to the dynamics of the tumour-immune-system interaction and – based on this - to new treatment concepts.

However, there are some concerns for modellers and clinicians as well. The process of modelling itself and simulations often helps to refine ideas and concepts about the investigated system. Regardless of the uncertainty concerning the parameter values, the range of parameter values was restricted to achieve more or less clinically realistic tumour control. In this range, the presented model depicts some aspects of the fundamental dynamics regarding innate immune system activation in combination with radiation-induced cell killing.

The model does not include competition between the tumour cell population and / or the host tissue, representing the situation of aggressive and fast growing tumours. In this regimen, considering different tumour sub-populations with varying radio-biological and immunological properties exhibits only a small influence on the tumour control, at least for the investigated parameter range. Detailed modelling of sub-clone – and host tissue interactions may become important when competition between the different populations reaches certain strength.

For description of the system, a compartmental model using ordinary differential equations is used. One may argue that some aspects of the ecosystem dynamics are related to the spatial distribution of cells / populations. Histological images from aggressive, highly malignant tumours often exhibit a more or less chaotic patchwork of host tissue, normoxic, hypoxic and necrotic areas, proliferating and apoptotic tumour cells etc.. Considering highly malignant tumours, the added value of spatio-temporal models is unclear. Therefore, the influence of spatial aspects in function of tumour malignancy on tumour evolution is an interesting and important research topic.

The general system behaviour exhibits the strong influence of repair on tumour control for the selected α – and β – values (representing a tumour with comparably high radio sensitivity). The reduced repair speed of tumour cells compared to the host tissue leads to a “therapeutic window” between TCP and Normal Tissue Complication Probability NTCP. No repair ($\gamma = 0 \text{ d}^{-1}$) will lead to survival which follows the linear-quadratic law, independent of fractionation. This results in a non-realistic TCP. Depending on the repair speed, immune system mediated response seems to be circumstantial compared to the cell-intrinsic radiation biology (incomplete repair). But for the most of the investigated cases (over a wide range of parameter values), the mixed protocol exhibits a slight and for stronger immune response a clear advantage compared to the daily applied fractions. In clinical practice, daily application of RT fraction is normally limited to Monday-Friday with an interruption every week end. This results in a “mixed” fractionation scheme, where the appearance of the larger intervals during treatment course is defined by the starting day of the treatment. It has to be

pointed out that there are many other fractionation schemes such as hypo-fractionation (e.g. de Dios et al., 2017), brachytherapy or stereotactic radio-surgery, which may support the idea of a biological treatment planning using *in silico* - models.

There is some evidence for decreased tumour doubling time or increased rate of repopulation at the end of a treatment (Steel, 1977; Kim et al., 2005). This may be related to a smaller amount of tumour cells and related to this, less hypoxic cells or less competition. Another factor may be the selection of faster repopulating sub-clones during treatment course. The presented results are based on a constant tumour doubling. An accelerated growth at the end of RT treatment may be an additional argument for introducing longer intervals at the start of the treatment course.

For some indications, moderate hyperthermia is applied in combination with RT (HT-RT). In such therapy schemes, typically 3-6 RT-Fractions are combined with a hyperthermic treatment by heating the tumour tissue up to 42-43°C for 60-90 minutes. The effect of moderate hyperthermia in combination with Radiotherapy (HT-RT) was demonstrated in clinical trials (e.g. Wust et al. 2002) and experimentally in vitro and in vivo. Several mechanisms which potentially contribute to the treatment response have been discussed (Streffler, 1995). The impact of heat in combination with radiation on the level of the tumour ecosystem has not yet been investigated. Beside many other effects, hyperthermia leads to an increased tissue perfusion. This may result in an improved accessibility of immune cells (and to a changed tumour micro-environment). In combination with the heat-induced expression of heat-shock proteins (HSP), hyperthermia may enhance the intratumour immune cell activity leading to additional cell killing. An indication for the importance of this process is the fact that HT-RT seems to have improved clinical outcome with only a small number of combined HT-RT fractions compared to the total number of applied RT-fractions. Regarding the results presented above, it could be an interesting option to combine heating and irradiation with longer intervals at the beginning of the HT-RT course. For avoiding thermotolerance induction, HT-RT treatments are applied with a spacing of 3-7 days. This has to be taken into consideration for optimising treatment schemes. Population-based models for combined HT-RT such as the Multi-Hit Repair (MHR) model (Scheid-egger et al. 2013) can be integrated in the proposed frame-work to support therapy optimisation.

References

- Abbott, R. G., Forrest, S., and Pienta, K. J. (2006). Simulating the Hallmarks of Cancer. *Artificial Life*, 12:617–634.
- Baumann, M., Krause, M., Overgaard, J., Debus, J., Bentzen S.M., Daartz J. et al. (2016). Radiation oncology in the era of precision medicine. *Nat Rev Cancer*, 16(4):234-249.
- Basanta, D., Anderson, A.R.A. (2015). Exploiting ecological principles to better understand cancer progression and treatment. *Interface Focus*, 3:20130020. <http://dx.doi.org/10.1098/rsfs.2013.0020>
- Basanta, D., Gatenby, R.A., and Anderson, A.R. (2012). An exploiting evolution to treat drug resistance: combination therapy and the double bind. *Mol. Pharmaceut...* 9:914–921.
- Bentley, K. (2013). Artificial life in the fight against cancer: Extended abstract of invited keynote lecture. In 2013 *IEEE Symposium on Artificial Life (ALife)*, 145–147. doi:10.1109/ALIFE.2013.6602444

- Chen, K.W., and Pienta, K.J. (2011). Modeling invasion of metastasizing cancer cells to bone marrow utilizing ecological principles. *Theoretical Biology and Medical Modelling*, 8:36.
- Crespi, B., and Summers, K. (2005). Evolutionary biology of cancer. *Trends Ecol Evol*, 20:545–552. (doi:10.1016/j.tree.2005.07.007)
- Daugaard, M., Rohde, and M., Jaattela, M. (2007). The heat shock protein 70 family: Highly homologous proteins with overlapping and distinct functions. *FEBS letters*. 2007, Jul 31;581(19):3702-10.
- De Dios, N.R., Sanz, X., Foro, P., Membrive, I., Reig, A., Ortiz, A., Jiménez, R., and Algara, M. (2017). Accelerated hypofractionated radiation therapy (AHR) for non-small-cell lung cancer: can we leave standard fractionation? *Clin Transl Oncol*, 19(4):440-447.
- Fowler, J.F., and Ritter, M.A. (1995). A rationale for fractionation for slowly proliferating tumors such as prostatic adenocarcinoma. *Int J Radiation Oncology Biol. Phys.*, 32(2):521-529.
- Gatenby, R.A., Gillies, R.J., and Brown, J.S. (2010). The evolutionary dynamics of cancer prevention. *Nat. Rev. Cancer*, 10:526–527. (doi:10.1038/nrc2892)
- González-García, I., Solé, R. V., and Costa, J. (2002). Metapopulation dynamics and spatial heterogeneity in cancer. *Proc. Natl. Acad. Sci., U.S.A.* 99:13085–13089.
- Grimsley, C., and Ravichandran K.S. (2003). Cues for apoptotic cell engulfment: Eat-me, don't-eat-me, and come-get-me signals. *Trends in Cellular Biology*, 9:465.
- Horsman, M.R., Mortensen, L.S., Petersen, J.B., Busk, M., and Overgaard J. (2012). Imaging hypoxia to improve radiotherapy outcome. *Nat. Rev. Clin. Oncol.*, 9(12):674-687.
- Kim, J.J., and Tannock, I.F. (2005). Repopulation of cancer cells during therapy: an important cause of treatment failure. *Nature Reviews Cancer*, 5:517-525.
- Maley, C. C. and Forrest, S. (2000). Exploring the Relationship between Neutral and Selective Mutations in Cancer. *Artificial Life*, 6:325–345.
- Mehrara, E., Forssell-Aronsson, E., Ahlman, H., and Bernhardt, P. (2007). Specific growth rate versus doubling time for quantitative characterization of tumor growth rate. *Cancer Res.*, 67(8):3970-3975.
- Merlo, L.M.F., Pepper, J.W., Reid, B.J., and Maley, C.C. (2006). Cancer as an evolutionary and ecological process. *Nat. Rev. Cancer*, 6:924–935. (doi:10.1038/nrc2013)
- Nytko, K.J., Thumser-Henner, P., Weyland, M.S., Scheidegger, S., and Rohrer Bley, C. (in press). Sensitivity towards thermoradiotherapy treatment in canine and human cancer cell lines. *PLOS ONE*.
- Ozik, J. et al. (2018). High-throughput cancer hypothesis testing with an integrated PhysiCell-EMEWS workflow. *BMC Bioinformatics*, 19:483.
- Pienta, K.J., McGregor, N., Axelrod, R., and Axelrod, D.E. (2008). Ecological Therapy for Cancer: Defining Tumors Using an Ecosystem Paradigm Suggests New Opportunities for Novel Cancer Treatments. *Translational Oncology*, 1:158–164.
- Scheidegger, S., Fuchs, H.U., Zaugg, K., Bodis, S., and Fuchsli, R.M. (2013). Using State Variables to Model the Response of Tumour Cells to Radiation and Heat: A Novel Multi-Hit-Repair (MHR-) Approach. *Computational and Mathematical Methods in Medicine*, 2013, <http://dx.doi.org/10.1155/2013/587543>
- Scheidegger, S., Lutters, G., and Bodis, S. (2011a). A LQ-based kinetic model formulation for exploring dynamics of treatment response of tumours in patients. *Z. Med. Phys.*, 21:164–173.
- Scheidegger, S., and Fuchsli, R.M. (2011b). Kinetic model for dose equivalent – an efficient way to predict systems response of irradiated cells. *Proc. of ASIM 2011*, (full papers, ISBN 978-3-905745-44-3).
- Scheidegger, S., Lutters, G., and Bodis, S. (2010). Computer simulation of oxygenation and growth inhibition of tumours during fractionated radiotherapy: Exploration of the dynamic system behaviour. *Radiother. Oncol.*, 95, Suppl.1:512.
- Scott, J.G., Fletcher, A.G., Anderson, A.R.A., and Maini P.K. (2016). Spatial metrics of tumour vascular organisation predict radiation efficacy in a computational model. *PLOS Comput Biol*, 12(1):e1004712. doi:10.1371/journal.pcbi.1004712
- Steel, G. G. (1977). *The Growth Kinetics of Tumours*. Clarendon, Oxford.
- Srivastava, S. (2002). Roles of heat-shock proteins in innate and adaptive immunity. *Nature Reviews Immunology*, 2:185-194.
- Streffler, C (1995). Molecular and cellular mechanisms of hyperthermia. In: Seegenschmiedt, M.H., Fessenden, P., Vernon, C.C. (eds.), *Thermoradiotherapy and Thermochemotherapy*, pages 47-74, Springer Berlin, Heidelberg, New York.
- Thorwarth, D. (2017). Biologically adapted radiation therapy. *Z Med Phys*, 28:177-183.
- Van Leeuwen, C.M., Oei, A.L., Crezee, J., Bel, A., Franken, N.A.P, Stalpers, L.J.A., and Kok, H.P. (2018). The alpha and beta of tumours: a review of parameters of the linear-quadratic model derived from clinical radiotherapy studies. *Radiation Oncology*, 13:96, doi:10.1186/s13014-018-1040-z
- Vesely, M.D., Kershaw, M.H., Schreiber, R.D., and Smyth, M.J. (2011). Natural innate and adaptive immunity to cancer. *Annual Review of Immunology*, 29:235-271.
- Wust, P, Hildebrandt, B, Sreenivasa, G, et al. (2002). Hyperthermia in combined treatment of cancer. *THE LANCET Oncology*, 3(8):487-497.
- Zhang, L., Wang, Z., Sagotsky, J. A., and Deisboeck, T. S. (2009). Multiscale agent-based cancer modeling. *J. Math. Biol.*, 58:545–559.

Horizontal Gene Transfer Leads to Increased Task Acquisition and Genomic Modularity in Digital Organisms

Michael J Wisner^{1,2}, Rosangela Canino-Koning¹ and Charles Ofria^{1,2}

¹BEACON Center, Michigan State University

²Ecology, Evolutionary Biology, and Behavior Program, Michigan State University
mwiser@msu.edu

Introduction

Biological organisms exhibit a tremendous range of diversity across many different scales. At a genetic level, one of the starkest distinctions is between how genetic material is arranged in prokaryotes (such as bacteria) and eukaryotes (such as plants and animals). When a given biological process, such as enzymatic degradation of an energy source, requires several steps, it is common in eukaryotes for the genes encoding these steps to be widely dispersed through the organism's genome, though perhaps not as widely as would be expected from a purely random distribution (Lee and Sonnhammer, 2003). In contrast, bacterial genomes are typically arranged in operons, where some or all of the components needed for a particular function are found together in one section of the genome (Land et al., 2015). What might account for this difference in genome organization?

One intriguing possibility is that the greater extent of genome modularity in bacteria could be driven by horizontal gene transfer (HGT). HGT refers to a set of ways in which organisms can acquire genetic material from a source other than their direct parent(s), whether through ingesting nucleic acids as energy sources, accepting a plasmid from another cell, or even having nucleic acids injected by a virus (Soucy et al., 2015). HGT is generally seen as pervasive in bacteria, while considered less common in eukaryotes (Madamsetti and Lenski, 2018; Koonin, 2016). Because HGT can insert genes into new host organisms, it provides a potential selective benefit toward genomic arrangements where the instructions needed for a particular function are arranged compactly. An organism with all of the information needed for a complete function is more likely to manage to spread this genetic material via HGT if that material is localized to one genomic segment, instead of being diffuse throughout the genome (Lawrence, 1999; Lawrence and Roth, 1996). This very spread of information via HGT could even act to make bacterial genomes more modular (Kreimer et al., 2008). These processes can reinforce each other – a more modular genome makes for more effective HGT, which then causes the resulting genomes to be even more modular.

Testing the evolutionary impacts of either allowing or dis-

allowing particular types of mutations is exceptionally difficult in physical organisms. Therefore, we chose to address these questions by harnessing the power of digital evolution.

Study System and Experimental Design

Avida (Ofria et al., 2009) is a digital evolution software platform in which self-replicating computer programs compete for digital resources. Within this system, we evolved populations in two fundamental treatments. Both treatments involved cycling between two environments, which differed in whether performing particular Boolean logic tasks was rewarded or punished, a setup we have previously shown to promote evolvability (Canino-Koning et al., 2018). We evolved populations for 200,000 updates, alternating between our changing environments every 500 updates. Each evolving population was placed into one of two treatments. In one treatment, we included in the set of possible instructions one that, when executed, produced an HGT event; in the other treatment, no HGT was possible. At the end of each run, we took the numerically most abundant genotype and analyzed both its genotype (sequence of instructions) and phenotype (which tasks it did and did not perform). We ran 50 replicates of each treatment. 5 of these 100 runs crashed and thus do not contribute to the current study; we are rerunning those seeds for a complete data set.

Results and Discussion

The inclusion of HGT leads to the evolution of more tasks (Fig. 1). Among the final organisms, more tasks were performed by individuals in the treatment where HGT was possible than the one where it was not (mean task number 2.956 v 2.149, Wilcoxon rank sum test $W = 1642$, two-tailed $p \ll 0.0001$). This difference is not only significant, but also substantial; on average, HGT leads to more than a 35% increase in the number of different tasks being performed by the organism. Even within the context of a changing environment the addition of HGT further increases the number of tasks that the most abundant organisms perform.

The organisms that evolved in the presence of HGT also have more modular genomes. The number of tasks that a

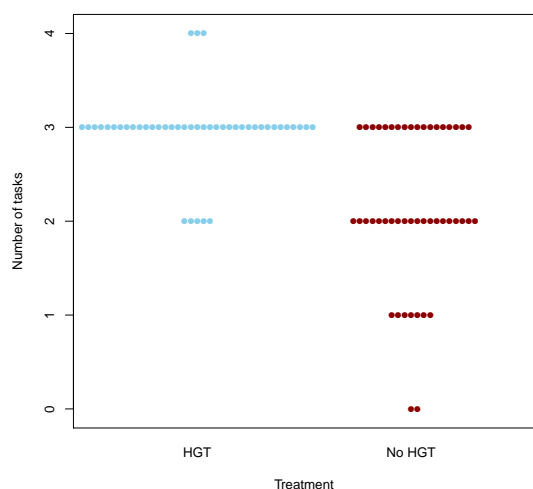


Figure 1: Number of tasks in final dominant organism. Labels indicate the evolutionary environment

given site in the genome contributes to is slightly higher in the HGT-enabled treatment (mean number of tasks per site: 1.474 vs 1.350, Wilcoxon rank sum test $W = 1396$, two-tailed $p = 0.0082$), but this difference is of a much smaller relative magnitude (approximately 13.8%) than the difference in the number of tasks performed (approximately 37.6%). When we normalize the number of tasks per site by the total number of tasks in an organism, there are fewer in the populations that have experienced HGT (mean 0.5077 v 0.6293, Wilcoxon rank sum test $W = 592.5$, $p < 0.0003$), a reduction of approximately 19.3%.

We recognize that the number of tasks that a given site is involved in performing is not the only way to measure the modularity of a genome. See, for example, (Misevic et al., 2006) for methods of measuring both physical and functional modularity in digital genomes. Work is ongoing to more fully characterize the evolved genomes under various measurements of modularity.

Conclusions

In a changing environment, Horizontal Gene Transfer (HGT) leads to the evolution of more tasks than are found in a corresponding treatment without HGT. The resultant genomes are also more modular, with fewer tasks encoded per site when accounting for the total number of tasks performed by the organism. This evidence supports the hypothesis that HGT itself may explain some of the different patterns in genome organization found in different types of biological organisms; those that undergo extensive HGT may have more modular genomes as a direct consequence of this parasexual process.

Scripts and Reproducibility

The configuration settings for Avida used in this experiment, as well as the R script used to analyze the results, are available at <http://github.com/mjwiser/ALife2019HGT>.

Acknowledgements

This research was supported in part by the BEACON Center for the Study of Evolution in Action (NSF Cooperative Agreement No. DBI-0939454) and by Michigan State University through the computational resources provided by the Institute for Cyber-Enabled Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Canino-Koning, R., Ofria, C., and Wisner, M. (2018). Fluctuating environments select for short-term phenotypic variation leading to long-term exploration. *bioRxiv*, page 394676.
- Koonin, E. V. (2016). Horizontal gene transfer: essentiality and evolvability in prokaryotes, and roles in evolutionary transitions. *F1000Research*, 5:F1000 Faculty Rev-1805.
- Kreimer, A., Borenstein, E., Gophna, U., and Ruppim, E. (2008). The evolution of modularity in bacterial metabolic networks. *Proceedings of the National Academy of Sciences*, 105(19):6976.
- Land, M., Hauser, L., Jun, S.-R., Nookaew, I., Leuze, M. R., Ahn, T.-H., Karpinet, T., Lund, O., Kora, G., Wassenaar, T., Poudel, S., and Ussery, D. W. (2015). Insights from 20 years of bacterial genome sequencing. *Functional & Integrative Genomics*, 15(2):141–161.
- Lawrence, J. (1999). Selfish operons: the evolutionary impact of gene clustering in prokaryotes and eukaryotes. *Current Opinion in Genetics & Development*, 9(6):642–648.
- Lawrence, J. G. and Roth, J. R. (1996). Selfish operons: horizontal transfer may drive the evolution of gene clusters. *Genetics*, 143(4):1843.
- Lee, J. M. and Sonnhammer, E. L. (2003). Genomic Gene Clustering Analysis of Pathways in Eukaryotes. *Genome Research*, 13(5):875–882.
- Maddamsetti, R. and Lenski, R. E. (2018). Analysis of bacterial genomes from an evolution experiment with horizontal gene transfer shows that recombination can sometimes overwhelm selection. *PLOS Genetics*, 14(1):e1007199.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proceedings of the Royal Society of London B: Biological Sciences*, 273(1585):457–464.
- Ofria, C., Bryson, D. M., and Wilke, C. O. (2009). Avida. In *Artificial Life Models in Software*, pages 3–35. Springer London.
- Soucy, S. M., Huang, J., and Gogarten, J. P. (2015). Horizontal gene transfer: building the web of life. *Nature Reviews Genetics*, 16:472.

Attractor Landscapes and Information Processing by Convective Obstacle Flows

Stuart Bartlett^{1,2}, Yuk L Yung¹

¹California Institute of Technology, Pasadena, United States

²Earth Life Science Institute, Tokyo, Japan
sjb@gps.caltech.edu

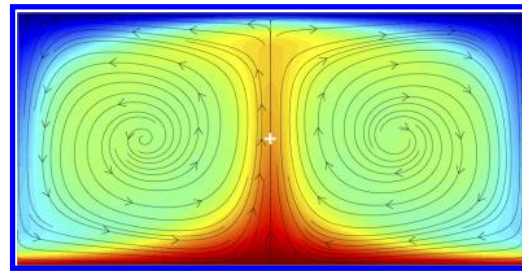
Abstract

We present recent results concerning the attractor landscape, memory, hysteresis and computation that can emerge in simple convective obstacle flows. In these systems a single phase fluid is heated from below and cooled from above. Small obstacles (one or two) are placed on the horizontal mid plane of the system and extract some fraction of the fluid's horizontal or vertical momentum. Horizontal momentum sinks tend to attract convection plumes. Vertical momentum sinks are bistable; the obstacle will either align with a convection cell centre or convection plume depending on initial conditions and the history of the system. The resulting attractor landscape can be exploited to produce a single bit memory or even elementary Boolean logic.

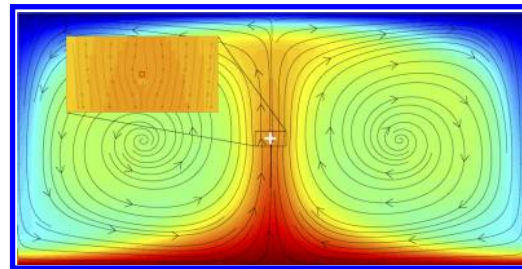
In this work we use numerical simulations of 2D convection flows to illuminate the rich dynamical behaviours that thermal fluids can exhibit. Natural convection has been studied for centuries, but many aspects of this non-linear phenomenon are still a matter of active research (e.g., Ahlers et al., 2009; Manneville, 2006). In this study we explored the effects of small obstacles placed in a driven thermal fluid system. The boundary temperatures are held constant and adjusted such that the dimensionless driving force is also constant.

Obstacles placed in such flows cause a break in their horizontal translational symmetry. When such obstacles, or momentum sinks, extract only horizontal momentum, the flow aligns plumes (maxima of vertical momentum) with the sink, as shown in Fig. 1(a). A similar attractor occurs when both horizontal and vertical momentum are extracted, as illustrated in Fig. 1(b).

These two attractors are identical except for the fact that the total momentum sink (Fig. 1(b)) is removing a significant quantity of kinetic energy from the flow, whereas the horizontal sink (Fig. 1(a)) is not extracting any energy since the flow



(a)



(b)

Figure 1: Typical flow structure of a convective fluid driven at a Rayleigh number of $Ra = 10^5$ in the presence of a small sink of momentum placed at $[x, y] = [W/2, H/2]$ (the white crosshairs show its location, its size is only $\sim 0.7\%$ of the system height). a) Horizontal momentum extraction, b) Total momentum extraction.

field at that point (plume centre) is purely vertical (no horizontal momentum to extract).

When only vertical momentum is extracted by the sink, there are two types of attractor that occur: one in which either vortex centre aligns with the sink (Fig. 2(a)), and another in which either convection plume aligns with the sink (Fig. 2(b)). A detailed analysis of the fluid dynamical basis of these effects is given in Bartlett and Yung (2019).

The bistability and attractive/repulsive be-

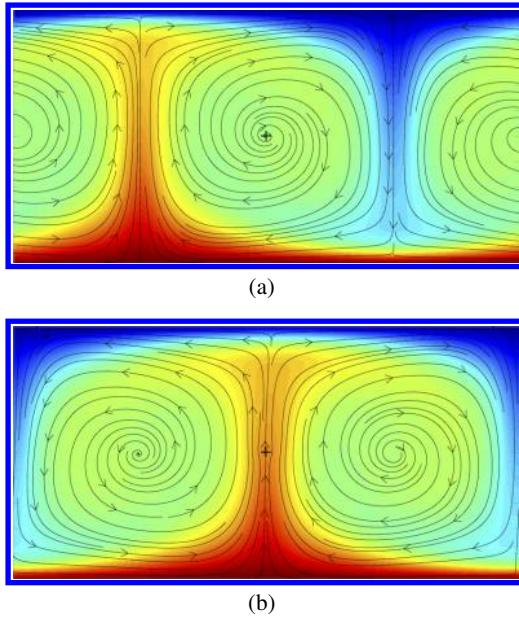


Figure 2: The two different types of attractor of a convective fluid driven at a Rayleigh number of $Ra = 10^5$ in the presence of a small sink of vertical momentum placed at $[x, y] = [W/2, H/2]$. a) Clockwise vortex attractor, b) Upwelling plume attractor.

haviour outlined above can be exploited to confer information storage and processing capabilities. Fig. 3 shows the history-dependence of the state of a flow system in which vertical momentum is extracted by the sink, but horizontal momentum extraction is switched on and then back off. Applying horizontal momentum extraction shifts the state of the system to the plume attractor, which is an absorbing state. Thus, when the parameters are returned to their original values, the system does not return to its original configuration. Convection flows can therefore store at least a single bit of information using this state history dependence.

Furthermore, simple Boolean logic, such as the OR gate, can be implemented using this simple system. In fact, several logic gates can be instantiated when two obstacles are placed in the flow Bartlett and Yung (view). These results suggest a new direction for the field of fluid computation (Adamatzky, 2018; Foster and Parker, 1970; Katsikis et al., 2015; Prakash and Gershenfeld, 2007), which has so far not explored the use of convection flows for simple computational logic.

We are now exploring the integration of convective logic gates such that entire computational

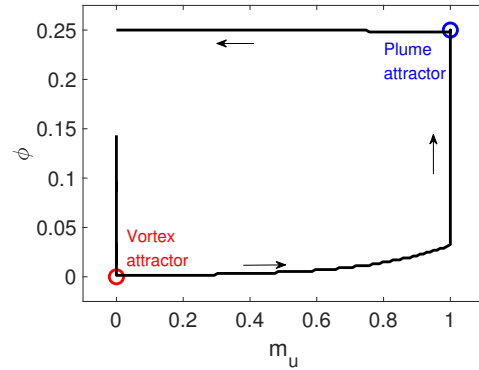


Figure 3: Phase space behaviour for a convective obstacle simulation with time-varying horizontal momentum extraction.

circuits can be constructed using only thermal fluid units. Results such as these suggest the use of information theory and computer science to understand a wide range of non-linear and non-equilibrium systems that traditionally lie outside of these fields (Feldman et al., 2008).

References

- Adamatzky, A. (2018). The dry history of liquid computers. *arXiv preprint arXiv:1811.09989*.
- Ahlers, G., Grossmann, S., and Lohse, D. (2009). Heat transfer and large scale dynamics in turbulent rayleigh-bénard convection. *Rev. Mod. Phys.*, 81:503–537.
- Bartlett, S. J. and Yung, Y. L. (2019). Convective flow in the presence of a small obstacle: Symmetry breaking, attractors, hysteresis, and information. *Phys. Rev. E*, 99:033103.
- Bartlett, S. J. and Yung, Y. L. (Under review). Computation by convective obstacle flows.
- Feldman, D. P., McTague, C. S., and Crutchfield, J. P. (2008). The organization of intrinsic computation: Complexity-entropy diagrams and the diversity of natural information processing. *Chaos*, 18(4):043106.
- Foster, K. and Parker, G. (1970). *Fluidics: components and circuits*. Wiley-Interscience.
- Katsikis, G., Cybulski, J. S., and Prakash, M. (2015). Synchronous universal droplet logic and control. *Nature Physics*, 11(7):588.
- Manneville, P. (2006). Rayleigh-bénard convection: thirty years of experimental, theoretical, and modeling work. In *Dynamics of Spatio-Temporal Cellular Structures*, pages 41–65. Springer.
- Prakash, M. and Gershenfeld, N. (2007). Microfluidic bubble logic. *Science*, 315(5813):832–835.

On Sexual Selection in the Presence of Multiple Costly Displays

Clifford Bohm^{1,3,4}, Acacia L. Ackles^{1,3,4}, Charles Ofria^{2,3,4} and Arend Hintze^{1,2,3,4}

¹Department of Integrative Biology

²Department of Computer Science and Engineering

³BEACON Center for the Study of Evolution in Action

⁴Michigan State University, East Lansing, MI 48824

cliff@msu.edu

Abstract

Sexual selection is a powerful yet poorly understood evolutionary force. Research into sexual selection, whether biological, computational, or mathematical, has tended to take a top-down approach studying complex natural systems. Many simplifying assumptions must be made in order to make these systems tractable, but it is unclear if these simplifications result in a system which still represents natural ecological and evolutionary dynamics. Here, we take a bottom-up approach in which we construct simple computational systems from subsets of biologically plausible components and focus on examining the underlying dynamics resulting from the interactions of those components. We use this method to investigate sexual selection in general and the sexy sons theory in particular. The minimally necessary components are therefore genomes, genome-determined displays and preferences, and a process capable of overseeing parent selection and mating. We demonstrate the efficacy of our approach (i.e we observe the evolution of female preference) and provide support for sexy sons theory, including illustrating the oscillatory behavior that developed in the presence of multiple costly display traits.

Introduction

Sexual selection occurs when a member of one sex selects mates based on traits displayed in individuals of another sex. The effects of sexual selection are clearly evident in natural systems, from the bright coloration of male cichlid fishes (Payne and Krakauer, 1997) to the sweeping antlers of cervids (deer, elk and moose) (Clutton-Brock, 1982) to the brilliant tail feathers of peacocks (Petrie et al., 1991). In addition to being attractive to mates, such display traits can be costly both to produce (Hunt et al., 2004) and to select for (Pomiankowski, 1987; Head et al., 2005). Despite an abundance of literature, a unified understanding of the mechanisms that drive and maintain sexual selection in the face of costs has not yet been formulated.

A number of theories have been developed to explain some mechanisms of sexual selection based upon observations of natural and theoretical systems. Primary among these theories are runaway selection (Grier, 1930), good genes (Rowe and Houle, 1996), sexy sons (Weatherhead and

Robertson, 1979), sexual conflict (Chapman et al., 2003) and the handicap principle (Zahavi, 1975). While these are often viewed as competing theories, a growing community is developing that argues that they are actually complementary, each explaining some aspect of sexual selection dynamics (Kokko et al., 2002).

Here we study sexual selection using a dynamic agent-based system composed of many parts that can each be designed independently. Our approach is not to attempt to model any particular natural system, but rather, to define system parts that each simulate different aspects of naturally occurring sexual selection. This ‘system of parts’ will allow for modeling of both naturally occurring as well as plausible, but not currently existing phenomena. As each part of the model is validated, we will develop new parts to increase the complexity of the phenomena that can be studied and improve the systems applicability to nature. Since the system of parts will be built up rather than designed to mimic specific natural systems, we can also remove parts to determine if they are critical to a particular phenomenon.

We have chosen to focus on the sexy sons theory (see below) because of its simplicity. In order to observe sexy sons behavior we needed only a small number of elements: (1) mutable and heritable genomes, (2) genetically determined preferences and displays (with assignable costs), and (3) a process capable of overseeing parent selection and mating. In particular to investigate sexy sons, we did not need to consider significantly more complex system parts that would be needed to allow for phenomena such as condition dependent displays, developmental processes or, complex behaviors related to life histories, nuptial gifts, or parental care.

Sexy Sons Theory

Ronald Fisher first proposed the idea of runaway sexual selection, a condition in which a trait that signals fitness becomes preferred by sexual selection and subsequently is exaggerated to such an extent that the trait becomes costly. The sexy sons theory extends Fisherian selection with the idea

that a female’s preference for a potentially costly male trait¹ could increase the offspring production of males with that trait. It logically follows from sexy sons theory that it would be beneficial for a female to select these preferred males so that her male offspring can inherit their father’s attractiveness (sexy sons) and those sons’ increased offspring count would improve her long term fitness.

While sexy sons is generally accepted as theory, few empirical studies have tested its critical hypothesis (Huk and Winkel, 2008) and significant doubts remain (Kirkpatrick, 1985). This lack of empirical evidence is due primarily to challenges that limit testing sexy sons in biological contexts. For example, how can an experimenter ensure that a male display is a sexy sons trait? (i.e., that the trait provides no beneficial information). Even if such a trait can be identified, assessing the reproductive success of organisms would require long-term, highly controlled experiments, extensive genotyping, and accurate phylogeny reconstruction. A computational approach, however, will be faster and provide perfect data collection, allowing for clearer insight into the dynamics of sexual selection.

In this paper we turn to computational agent-based evolution to investigate sexy sons. Previous research has employed theoretical methods, both mathematical and computational, to attempt to understand the sexy sons hypothesis (see Kokko et al. (2006) for a survey of the topic), but prior work has tended to model only a single sexy sons trait regulated by a single genetic loci and then investigate under what conditions that trait will be selected. Pomiankowski and Iwasa (1998) did examine multiple trait sexual selection using a system of equations to model two traits, but their formulation was not agent based and did not included either inheritance or mutation. Interestingly, even though Pomiankowski’s system is quite different from ours, it did produce cyclic display and preference behavior that aligns with our results. We and others (Mead and Arnold, 2004) argue that the two loci approach and other simplifications that are often employed in generating models lack the complexity needed to model sexual selection properly.

Here, we ask: **What effect does the number of costly traits have on the dynamics of sexual selection?** Using this system we show that the behavior generated by sexual selection when only a single costly display trait is available creates a illusion of stability. We then show that when multiple equally costly display traits are available, female preference will oscillate between the available traits.

Methods

We used the MABE (Modular Agent Based Evolver) framework (Bohm et al., 2017) to implement and evaluate populations of evolving digital organisms with genetically deter-

¹In this paper we will use the convention that females act as selectors and males are selected. This division is common in nature due to differential offspring investment costs, but is not universal.

mined sex, sexual preferences, and sexual displays. MABE is a general-purpose evolutionary computation and artificial life research tool that allows researchers to construct experiments by combining different modules. Modules in MABE have five different types: Brains (neural/cognitive digital architectures), Genomes (sources of mutable and heritable information), Worlds (problem descriptions/environments), Optimizers (parent selection and population management) and Archivists (data tracking and recording). For this work we developed an optimizer called Two Sexes Optimizer that manages gene detection, sex determination, parent selection and reproduction. Aside from the Two Sexes Optimizer we were able to rely on existing modules. We used the Circular Genome and Default Archivist modules. Due to the simple nature of our experiments we did not need to use either a brain or world module.

The digital organisms in this work each had a haploid genome (a list of 1000 integers valued 0 to 255). An organism’s sex, sexual preference, and sexual display level were determined by scanning its genomes for specific genes, each indicated by a different sub-sequence (start codon). Sex genes used a 4-digit start codon (the values 101,102,103,104) with the parity of next site determining the sex. Preference genes also used a 4-digit start codon (105,106,107,108) with the value of the next site determining which trait is preferred in a mate (using the modulus operator and the number of available display traits). Organisms were required to have one and only one sex gene and one and only one preference gene or they were removed from the population without having an opportunity to reproduce. Display genes each had a unique 2-digit start codon and the number of each display gene start codon found within the genome was used to determine the level of the associated display. While an organism could have any number of genes for each display trait in their genome, the display values were capped at 50 (i.e. an organism with 51 or more genes of a particular display appears to females as though they had 50 genes of that display). Random selection was implemented by adding a display trait with a fixed value of 0, which was not associated with any gene, to all males. This constant display trait allowed for random selection; as all males had the same display value for this trait, females selecting based on this trait were selecting randomly.

Each organism received a score, S , which determined the organism’s number of mating opportunities relative to the rest of the organisms of the same sex in the population. In all of the experiments in this manuscript, we allowed one beneficial display trait, and zero or more costly display traits. Females scores (S_F) were calculated by

$$S_F = \min(t_b, 60) \quad (1)$$

where t_b is the number of beneficial display trait genes in their genome.

Male scores (S_M) were calculated by

$$S_M = \min(t_b, 60) - C \sum_{t \in T_c} \max(0, t - 10) \quad (2)$$

where t_b is as above, C is the cost coefficient (which sets to cost of costly traits), and T_c is the number of display trait genes in their genome for each display trait in the set of costly traits in the current experiment. In other words, both males and females receive a base score equal to the level of their beneficial trait (to a maximum of 60). Females pay no costs, and males pay a cost relative to the number and level of their costly displays but only for those displays that are above 10.

While exploring parameter values (such as cost coefficient and population size) there were a number of essentially arbitrary decisions we needed to make. Two of these decisions, while not necessary to generate the effects observed, did significantly improve the consistency of the results.

First, the score cap for the beneficial display trait was set at 60 but the value for display traits where females can no longer detect differences was set at 50. This difference means that the beneficial display trait continued to provide benefits even after changes to the trait were undetectable to females. As a result, if the beneficial display trait were to drift away from its optimal value (60) it would not immediately create a signal that was detectable to females, which might become a focus for selection. Similar beneficial traits are likely found in nature; perhaps fur color, where predators have more sensitivity to the color range than females, or a vocalization that acts both as a warning and a mating call where the auditory range of the target of the warning is greater than that of the females.

Secondly, costs were not applied on male display traits when they were less than or equal to 10 because we found that if these displays were costly at any level, then the associated genes generally would be purged from the genome when not being selected for. Once a gene had been entirely purged a rare mutation would be required for it re-appear. The costly display traits describe traits that at low levels are not significantly costly. This type of costly display can be thought of as tail length in a species where short to medium tail length has no significant fitness effect, but beyond some limit, tails become costly either because of upkeep, predation risk, or some other factor.

To initialize each run, we generated a population with 1000 organisms with randomized genomes. We seeded each organism's genome with one sex gene at genome location 250 and one preference gene at location 750. The spacing between sex and preference genes ensured that these genes would be able to be inherited separately (i.e. low linkage). Since the sex and preference of each organism was based on the site following the start codons, the sex and preference of each initial organism were random. The genomes were also seeded with five randomly positioned copies of each of five

display trait genes, one beneficial, and four costly. For consistency, we seeded genomes with the five different display trait genes in every condition even though not all five were used in all experiments.

We evolved each population for 10,000 generations. Every generation each organism's sex, sexual preference and display trait levels were read from that organism's genome. Organisms that did not have exactly one sex gene and one preference gene were considered to have experienced a lethal mutation and were discarded. We then divided the remaining population into females and males. A female was selected using roulette selection (fitness proportional) and then a collection (a lek) of 20 males were selected, also using roulette selection, for that female to choose from. The female mated with the male in the collection with the highest display level matching the female's preference (or a random male from the collection if the female had no preference) and produced an offspring. This process was repeated many times to produce the next generation of 1000 organisms. The parents were then discarded; that is, organisms lived for exactly one generation.

Mating consisted of crossing the selected parents genomes and applying mutations to the resulting genome. We setup genomes to have to have 19 equally-spaced crossover points, generating 20 genome segments. For each pair of segments, the system selected a random parent from the mating pair and used that parent's genome for the first genome segment of the offspring genome. Then the second genome segment was contributed by the other parent. This process was repeated for the remaining segments. This process is comparable to a 10-chromosome genome where each parent randomly contributed $\frac{1}{2}$ of each of their chromosome to each offspring chromosome.

Two types of mutations were possible: point mutations and *indel* mutations. Point mutations changed the value of one site from its current value to a random number in the range of 0 to 255. Indel mutations copied a section of the genome (from 2 to 12 sites) and replaced another section of the genome of the same length with the copied values. Note that these mutation types were selected because they maintain genome length. We found that if insertion mutations were allowed the system would produce extremely large (and therefore slow to convert) genomes. Point mutations occurred at a rate of 0.001 per genome site (on average 1 per genome). Indel mutations occurred at a rate of .0001 per genome site (on average 1 per 10 genomes).

We chose to define display traits with 2-digit start codons so that there would be a non-trivial chance for them to spontaneously emerge as the result of mutation. Specifically, there is a $\frac{1}{65,536}$ chance that two randomly selected numbers will generate a specific pair. With a population size of 1000 organisms evaluated for 10,000 generations the creation of display start codons by mutation, while rare, becomes likely. Of course, once one or more genes exist, indel mutations can

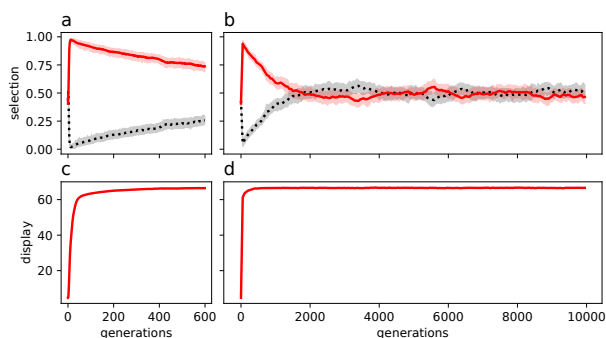


Figure 1: Average results of 100 replicates of Condition 1. 1.a and 1.b show percent of females selecting either for a beneficial trait (red) or randomly (black dotted) over 10,000 generations (1.a provides a detail of the first 600 generations). 1.c and 1.d show the display level of the beneficial trait over the same time scales. Initially selection for the beneficial trait is high, but as the trait passes 50 (the greatest value at which females are able to detect trait variation) selection drifts, eventually to a level equal to random selection. The trait eventually achieves a value slightly above 60 (the greatest value at which trait increases are rewarded). Shaded areas display 95% confidence.

copy these and the likelihood of an indel mutation resulting in a copied gene increases as there are more of that gene present. On the other hand, we did not want a high rate of creation of sex and sex preference genes. The 4-digit sex and preference start codons are far less likely to arise by mutation ($\frac{1}{4,294,967,296}$) and so additional copies of these genes would most likely be the result of indel mutations.

We tested five conditions.

Table 1: Experimental Conditions.

Condition	Num. Costly Traits	Trait Cost
1	0	NA
2	1	1.0
3	1	0.1
4	4	1.0
5	4	0.1

In all five conditions, one beneficial display trait was visible to selection, as was the option for females to choose a mate randomly. Condition 1 served as a control, with no costly traits. Conditions 2 and 3 examine when a single costly trait was also available at a high or low cost, respectively. Conditions 4 and 5 expanded this test to four costly traits, again with high or low cost, respectively.

Costs were manipulated by altering the cost coefficient (C from equ. 2).

We ran 100 replicates of Condition 1 and 300 replicates each of Conditions 2,3,4, and 5. Each replicate was run with a different random seed for 10,000 generations with a population size 1000.

Readers wishing to replicate the results from this paper are directed to the supplemental materials which include

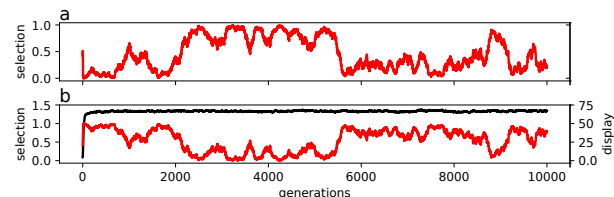


Figure 2: The results of a typical replicate from Condition 1. 2.a, shows the % of random selection. 2.b, shows the level of the beneficial display trait (black) and the % of selection for that trait (red). Selection for the beneficial trait is high until it reaches 50, after which selection drifts between random selection and the beneficial display trait.

files and instructions for generating the data presented in this paper (see: <http://github.com/cliff-bohm/ALIFE-2019-On-Sexual-Selection>).

Results

The results of Condition 1 (one beneficial trait) are shown in Fig.1. We observed that selection tended to fix on the beneficial display trait and remain fixed until that trait reached 50, the level over which females are unable to detect variation. Female preference then appeared to drift. Fig 1.b illustrates that random selection and selection for the positive trait appear to stabilize at approximately 50% each for remainder of the 10,000 generations. Condition 1 demonstrates that sexual selection in our system can target displays and will select for a beneficial display when there is detectable variation in the display and furthermore that if the beneficial trait's variation is undetectable to females, then the trait will be as attractive as random selection. Fig 2 shows behavior of a single replicate of Condition 1 and illustrates that the apparent selection stability between random selection and selection for the beneficial trait is really the result of averaging drift with higher variance across 100 replicates.

The results of Condition 2 (one beneficial trait and one costly trait at high cost) are shown in Fig 3. We observed that like Condition 1, sexual selection tended to select for the beneficial display trait early, but where Condition 1 resulted in drift once the beneficial display trait has reached 50, we see selection for the costly display trait develop. Once this trait reaches 50, we did not see the system tending to drift as in Condition 1. Rather, we observed persistent selection for the costly display trait and low occurrences of selection for the beneficial display trait or random selection. Fig. 5.a through c. show the behavior of an arbitrary replicate of Condition 2.

In Condition 3 (one beneficial trait and one costly trait at low cost), shown in Fig. 4, the reduced cost coefficient resulted in lower interest on the part of females for the costly display trait. As opposed to Condition 2 where selection for the costly display trait was consistently at or near 100%, in Condition 3, selection for the costly display trait hovered

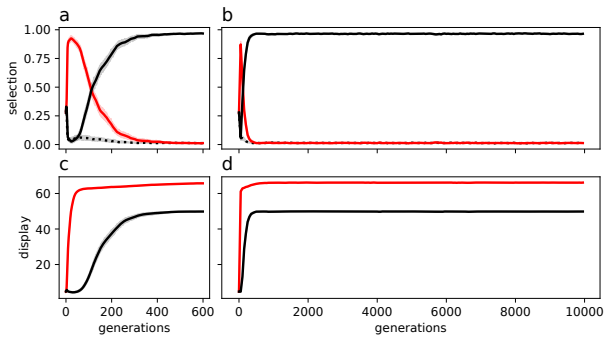


Figure 3: Average results of 300 replicates of Condition 2. 3.a and 3.b show the percent of females selecting either for a beneficial trait (red), a costly trait (black solid), or randomly (black dotted) over 10,000 generations (3.a provides a detail of the first 600 generations). 3.c and 3.d show the display level for the beneficial trait (red) and the costly trait (black) over the same time scales. Initially selection for the beneficial trait is high, but as the trait passes 50 (the greatest value at which females are able to detect trait variation) selection shift to the costly trait. The beneficial trait eventually achieves a value slightly above 60 (the greatest value at which trait increases are rewarded) while the costly trait stabilizes around 50. Shaded areas display 95% confidence.

around 75% after the beneficial display trait had exceeded 50. Random selection and selection for the beneficial display trait both maintain values near 12%. Interestingly, selection for the beneficial trait seemed to behave in the same manner as random selection supporting the idea that selection for the beneficial display trait provides the same benefit as random selection once the beneficial trait exceeded the level of female detection. 5.d though f. show the behavior of arbitrary replicate of Condition 3 and illustrate that the selection levels are not in fact stable and that the apparent stability in Fig. 4 is a the result of averaging replicates.

The results of Condition 4 (one beneficial trait and four costly traits at high cost) are shown in Fig 6. We observed the same early preference for the beneficial display trait that was seen in Conditions 1 and 2. Once the beneficial display trait reached 50 though, we did not observe a single costly trait being selected for, but rather we observed that all four of the 4 costly display traits are being selected for at comparable levels, about 25% and that the display levels all seem to be approximately the same, at a level of about 12. Fig. 8.a through f. show the behavior of an arbitrary replicate of Condition 4. Looking at the single replicate it is evident that the 4 costly traits are not at equilibrium as Fig. 6 suggests, but rather preference and trait values are constantly shifting. Note, that the averaged lines of the four traits overlap in Fig. 6, suggesting that each trait spent similar time under selection when compared across replicates.

In Condition 5 (one beneficial trait and four costly traits at low cost), shown in Fig. 7, the reduced cost coefficient resulted in comparable rates of selection among the four costly

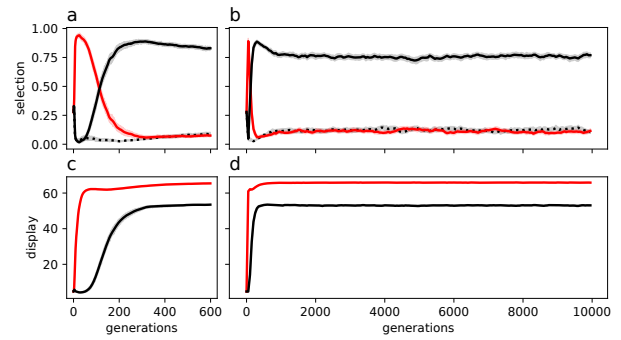


Figure 4: Average results of 300 replicates of Condition 3. This condition is identical to Condition 2 except that in Condition 2 the cost coefficient was 1.0 and here it is 0.1. See Figure 3 for description.

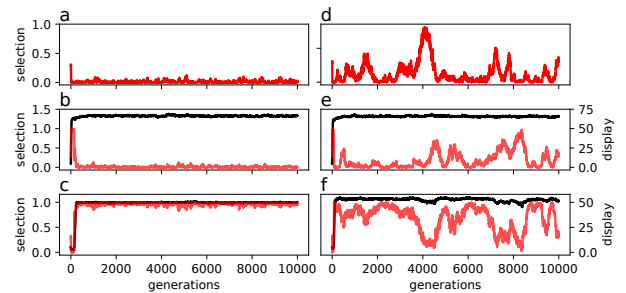


Figure 5: The results of two typical replicates, one from Condition 2 and one from Condition 3. 5.a, 5.b, and 5.c show the results of a Condition 2 replicate (one beneficial display trait and one costly display trait with cost coefficient = 1.0). 5.a, shows the % of random selection, 5.b, the level of the beneficial display trait (black) and the % of selection for that trait (red), and 5.c, the level of the costly display trait (black) and the % of selection for that trait (red). 5.d, 5.e, and 5.f show the same data for a replicate from Condition 3 (cost coefficient = 0.1). In both conditions, selection for the beneficial trait is high until it the beneficial trait reaches 50, after which selection is predominately focused on the costly display trait. The primary differences between conditions is the consistency of selection for the costly display trait (more focused when the cost is high) and, to a lesser extent, the consistency for the level of the costly trait itself. It can be observed in the low costly condition both random selection and selection for the beneficial trait appear to be targeted frequently when selection is not focused on the costly trait.

display traits as was observed in Condition 4, but higher levels in the costly display traits themselves. Perhaps these higher levels resulted from a slower decay rate due to the reduced selection pressure exerted by the decreased cost coefficient. 8.g though 1. show the behavior of an arbitrary replicate of Condition 5. Here again like in Condition 4, oscillations between different preferences and traits were seen.

Discussion

Costly display traits can be a preferred target for sexual selection. More surprisingly, they can even be preferred over neutral or even beneficial traits particularly when those

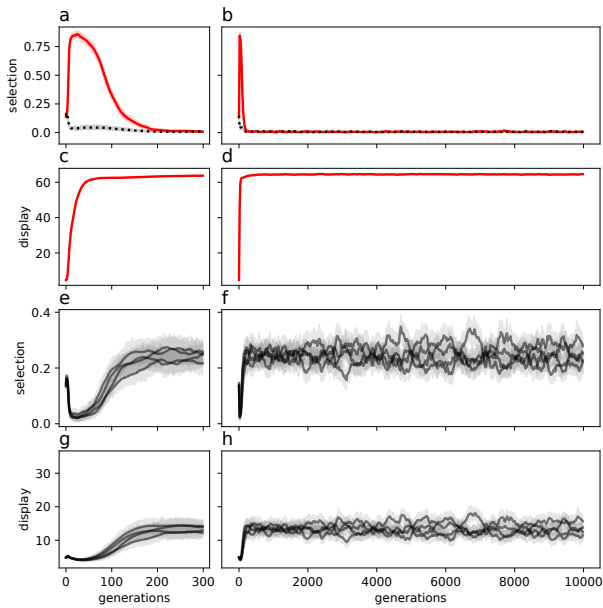


Figure 6: Average results of 300 replicates of Condition 4. 6.a and 6.b show the percent of females selecting either for a beneficial trait (red) or randomly (black dotted) over 10,000 generations (6.a provides a detail of the first 600 generations). 6.c and 6.d show the display level for the beneficial trait over the same time scales. 6.e and 6.f show the percent of females selecting for each of 4 costly traits. 6.g and 6.h show the levels of these four traits. The four costly traits all have the same cost and so are not individually labeled. Initially selection for the beneficial trait is high, but as the trait passes 50 (the greatest value at which females are able to detect trait variation) selection drifts and appears to be evenly divided among the four costly traits (each approximately 25 percent). The beneficial trait eventually achieves a value slightly above 60 (the greatest value at which trait increases are rewarded) while the costly traits each stabilize near 12. Shaded areas display 95% confidence.

traits fail to provide discriminatory information. This work demonstrates this result conclusively. The costly display traits were designed to provide no benefit, but clearly they must, and the most logical explanation for that benefit is the sexy sons hypothesis.

In the introduction we posed the question, **What effect do the number and cost of costly traits have on the dynamics of sexual selection?** We designed a series of experiments to demonstrate both the efficacy of our system and then to incrementally add complexity in the form of costly displays to explore system dynamics. We will now provide conjecture for what we believe are the driving factors that explain the observed behavior.

In Condition 1 we observe that selection for the beneficial display is high only when the beneficial display has detectable variation. Moreover, we observe that female preference for the beneficial trait behaves in the same manner as random selection, once that beneficial trait no longer has detectable variation. The fact that the beneficial trait maintains

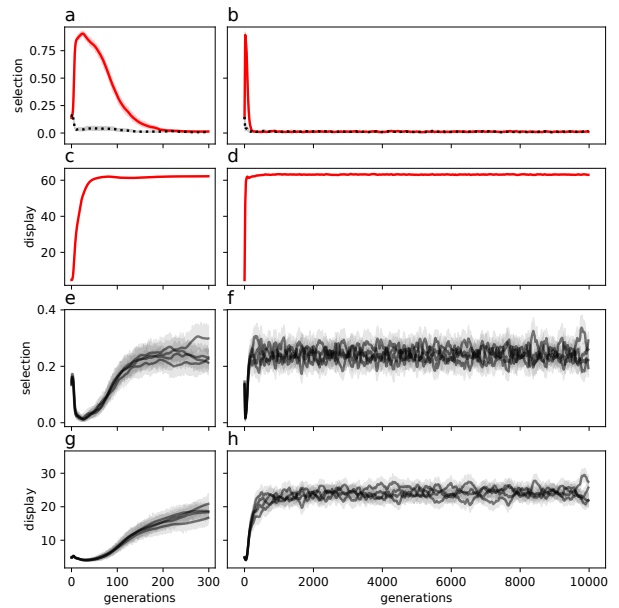


Figure 7: Average results of 300 replicates of Condition 5. This condition is identical to Condition 4 except that in Condition 4 the cost coefficient was 1.0 and here it is 0.1. See Figure 6 for description.

a high level (slightly above 60) is reasonable as those organisms who have this display at a higher than average level are provided more mating opportunities. Fig. 2 shows a typical replicate of Condition 1 and illustrates that preference is not actually balanced between random selection and the beneficial trait, but rather drifts between the two.

In Conditions 2 and 3 we observe the system with a single beneficial display and a single costly display. These conditions demonstrate that a sexy sons trait can (and in this system will) become the target of selection, once it is the only available target for selection. Females could have selected for either the beneficial display trait or accepted random mates. Instead, when presented with a collection of males, females overwhelmingly chose to mate with the male that had the lowest probability of having been selected to be part of that collection.

The increased rate of selection for the costly trait in Condition 2, versus Condition 3, may seem counter intuitive; Should we not observe lower levels of selection for the costly display when the display is more costly? We hypothesize that when the costly display value reaches 50 or above, selection for that trait begins to drift. At high cost, as female preference drifts there is greater pressure for males to reduce the level of their costly display which creates a greater variation and this creates a stronger signal and thus provides a larger sexy sons benefit.

In Condition 4 and 5 we considered the system with a single beneficial display and four costly displays. Fig. 6, suggests that the system stabilizes at around 25% selection for

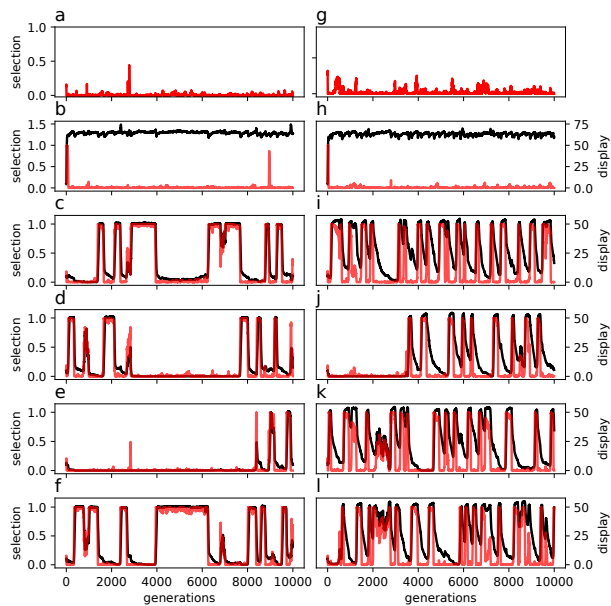


Figure 8: The results of two typical replicates, one from Condition 4 and one from Condition 5. 8.a through 8.f show the results of a Condition 4 replicate (one beneficial display trait and four costly display traits with cost coefficient = 1.0). 8.a, shows the % of random selection, 8.b, the level of the beneficial display trait (black) and the % of selection for that trait (red), and 8.c through 8.f, the level of each of the costly display traits (black) and the % of selection for that trait (red). 8.g through 8.l show the same data for a replicate from Condition 5 (cost coefficient = 0.1). In both conditions, selection for the beneficial trait is high until it reaches 50, after which selection oscillates among the four costly traits. The primary differences between conditions is the duration of the periods of selection for each costly display trait (more focused when the cost is high) and the slower rate at which the costly display traits fall when not under selection. Unlike Fig. 8, random selection or selection for the beneficial trait are rare. Data from additional replicates can be found in the supplemental material (<http://github.com/cliff-bohm/ALIFE-2019-On-Sexual-Selection>).

each of the four costly traits. This apparent result is not the case though. Fig. 8 shows the behavior of typical replicates of Conditions 4 and 5 and reveals that rather than stability, we observe oscillations among the four costly display traits. In Conditions 2 and 3, once the costly trait maxed out and began to decay, there was no other viable option for female preference (neither the random option nor the beneficial trait had detectable variation), so any small variation in the costly trait was the only signal that selection could target). In Conditions 4 and 5 when one costly display trait maxes out, there are always three other, equally viable, options. This is the primary result in this paper: when there are multiple costly traits *the stable state of the system is a state of constant change*.

How does the difference in the cost coefficient explain the differences between the results in Conditions 4 and 5? Fig.

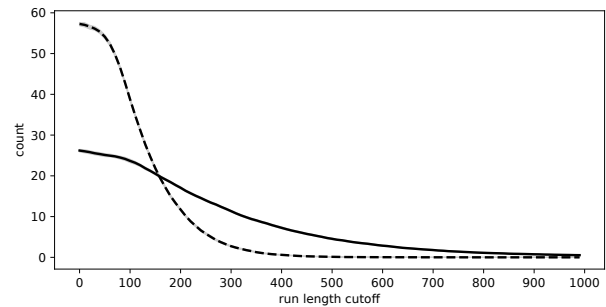


Figure 9: Compares the lengths of periods where a significant majority of females are selecting for a single display trait in Conditions 4 and 5 - one beneficial trait and four costly traits at high cost (4) and low cost (5). A period begins when the % of selection for a trait exceeds 90% and ends when it drops below 50%. All replicates from each condition were averaged, so values show average counts per replicate. The x-axis lists length of the periods from 1 generation to 1000 generations and the y-axis shows the average number of runs which were at least this long among all replicates in the higher cost Condition 4 (solid line) and lower cost Condition 5 (dashed line). The results illustrate that while Condition 4 had less overall runs they tended to be longer. Shaded areas are 95% confidence.

9 shows that at higher cost (in Condition 4) there are fewer total runs (defined as a period when a single display trait is being selected for by a majority of females), but these runs tend to last longer. We argued that in Condition 2, the relatively high cost of display causes a rapid decay in display once selection for that display begins to drift and this faster rate of decay, in turn, results in a stronger signal. We expect that the same phenomena may explain the relationship between cost and run length here. The stronger signal generated by higher cost results in more focused selection on the trait that is currently preferred.

What about the Handicap Principle and the Good Genes Theory?

Two models often cited to explain the existence of costly displays are the Handicap Principle and the good genes theory. We would be errant not to address these directly.

The Handicap Principle, argues that evolution may result in some costly displays that "serve as marks of quality" since an organism would need to be fit in order to survive with such handicaps (Zahavi, 1975). The handicap principle can not apply here since it requires the existence of condition dependant displays and the displays here are entirely genetically based.

The Good Genes theory, posits that an arbitrary display may be linked with some beneficial hidden gene. Essentially, the good gene provides an honest signal that a male contains a particular beneficial gene. Clearly the beneficial display trait in our experiments is a good gene. It is directly observable and is directly associated with mating opportunity. On the other hand, the costly display traits are clearly not. Good genes theory can not be used to explain the se-

lection of costly displays that we observe. It may be reasonable to argue that costly display traits are sometimes good genes. If females find a particular costly display trait attractive and select for that display to such an extent that it outweighs the cost then the presence of the gene in a male does (for a time) communicate that the male's offspring are likely to have higher than average reproductive success. Is there room in good genes theory for transient good genes? For a different approach to the same argument, see Kokko (2001).

Conclusion

In this work we demonstrated that sexual selection on costly traits can occur and that sexy sons provides a sound explanation. We also demonstrated that multi-trait sexual selection can result in a semi-stable state of constant change. This work investigated only a subset of parameters possible in the system used. From here we will continue to investigate the sexy sons theory, examining how this system responds to different costs, including female search costs and developing methods to investigate genetics effects such as how sexy sons and trait oscillation affects genomic rates of change. In the longer term, we will extend our system to include more complex features of sexual selection such as threshold selection, condition dependant traits, parental care, and sexual display signal fidelity.

Understanding sexual selection is not only important to understanding biology's history and predicting its future, but also may provide dynamics that could support open-ended evolutionary processes. Deeper understanding of the oscillatory behaviors generated by costly selection and how this phenomena alters genomes and phenotypes may explain how some species have been able to navigate away from local optima or across seemingly impossible fitness valleys. Finally, if we were able to control sexual selection it could even be used in engineering and machine learning contexts as a more naturalistic form of diversity search.

Acknowledgements

We wish to thank Vincent Ragusa, Jory Schossau, Alexander Lalejini, and Emily Dolson who all provided insights that helped develop this work. CB would like to particularly thank Jason Bundy for sharing his enthusiasm on the topic and for acting as the initial catalyst in a process that eventually resulted in this document. This work was supported by the National Science Foundation (NSF) through the BEACON Center (Cooperative Agreement DBI-0939454). Computational resources critical to this work were provided by Michigan State University's Institute for Cyber-Enabled Research.

References

Bohm, C., C G, N., and Hintze, A. (2017). Mabe (modular agent based evolver): A framework for digital evolution research.

In *Artificial Life Conference Proceedings 14*, pages 76–83. MIT Press.

Chapman, T., Arnqvist, G., Bangham, J., and Rowe, L. (2003). Sexual conflict. *Trends in Ecology & Evolution*, 18(1):41–47.

Clutton-Brock, T. (1982). The functions of antlers. *Behaviour*, 79(2-4):108–124.

Grier, N. (1930). Fisher. the genetical theory of natural selection (book review). *Social Forces*, 9(1):295.

Head, M. L., Hunt, J., Jennions, M. D., and Brooks, R. (2005). The indirect benefits of mating with attractive males outweigh the direct costs. *PLoS biology*, 3(2):e33.

Huk, T. and Winkel, W. (2008). Testing the sexy son hypothesis research framework for empirical approaches. *Behavioral Ecology*, 19(2):456–461.

Hunt, J., Brooks, R., Jennions, M. D., Smith, M. J., Bentsen, C. L., and Bussiere, L. F. (2004). High-quality male field crickets invest heavily in sexual display but die young. *Nature*, 432(7020):1024.

Kirkpatrick, M. (1985). Evolution of female choice and male parental investment in polygynous species: The demise of the "sexy son". *The American Naturalist*, 125(6):788–810.

Kokko, H. (2001). Fisherian and good genes benefits of mate choice: how (not) to distinguish between them. *Ecology Letters*, 4(4):322–326.

Kokko, H., Brooks, R., McNamara, J. M., and Houston, A. I. (2002). The sexual selection continuum. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 269(1498):1331–1340.

Kokko, H., Jennions, M. D., and Brooks, R. (2006). Unifying and testing models of sexual selection. *Annu. Rev. Ecol. Evol. Syst.*, 37:43–66.

Mead, L. S. and Arnold, S. J. (2004). Quantitative genetic models of sexual selection. *Trends in ecology & evolution*, 19(5):264–271.

Payne, R. J. and Krakauer, D. C. (1997). Sexual selection, space, and speciation. *Evolution*, 51(1):1–9.

Petrie, M., Tim, H., and Carolyn, S. (1991). Peahens prefer peacocks with elaborate trains. *Animal Behaviour*, 41(2):323–331.

Pomiankowski, A. (1987). The costs of choice in sexual selection. *Journal of theoretical Biology*, 128(2):195–218.

Pomiankowski, A. and Iwasa, Y. (1998). Runaway ornament diversity caused by fisherian sexual selection. *Proceedings of the National Academy of Sciences*, 95(9):5106–5111.

Rowe, L. and Houle, D. (1996). The lek paradox and the capture of genetic variance by condition dependent traits. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1375):1415–1421.

Weatherhead, P. J. and Robertson, R. J. (1979). Offspring quality and the polygyny threshold: "the sexy son hypothesis". *The American Naturalist*, 113(2):201–208.

Zahavi, A. (1975). Mate selection - a selection for a handicap. *Journal of theoretical Biology*, 53(1):205–214.

Steering the Growth of Adaptive Self-Preserving Dissipative Structures

Matthew Egbert^{1,2,*}, Yan Kolezhitskiy¹ and Nathaniel Virgo³

¹ School of Computer Science, University of Auckland, Auckland, NZ

² Te Ao Mārama — Centre for Fundamental Inquiry, University of Auckland, Auckland, NZ

³ Earth-Life Science Institute, Tokyo Institute of Technology, Tokyo, Japan

* m.egbert@auckland.ac.nz

Abstract

In the 1950s, the famous cyberneticists Gordon Pask and Stafford Beer conducted a series of remarkable electrochemical deposition experiments. By applying an electric potential across electrodes submerged in an acidic solution of ferrous sulfate, they could bias the growth of electrochemical deposition so as to form functional structures including sensory structures capable of distinguishing between different sounds. Unfortunately, the details of their apparatus and methods are unavailable. As a consequence, their experiment has not been replicated, and the precise mechanisms underlying their results remain unknown. As preliminary steps toward recreating their remarkable results, this paper presents a new computational model that simulates the growth and decay of dendritic structures similar to those investigated by Beer & Pask. We use this model to demonstrate a plausible mechanism through which an electrochemical system of this kind could respond to a reinforcement signal. More specifically, we investigate three strategies for varying the applied electrical current so as to guide the formation of structures into target forms. Each presented strategy succeeds at influencing the growth of the structure, with the most successful strategy involving a ‘constant-current’ feedback mechanism combined with an externally driven oscillation. In the discussion, we compare the adaptation of these structures with various biological adaptive processes, including evolution and metabolism-based adaptive behaviour.

Introduction

The research presented below is inspired by the electrochemical deposition experiments undertaken by Gordon Pask and Stafford Beer in the 1950s. By applying a current across electrodes placed in an acidic solution of ferrous sulfate, these researchers induced the electrochemical deposition of iron onto the negatively charged electrode(s) (see Fig. 1). By varying the applied voltage, they could choose when iron was deposited and when it dissolved back into solution, and using this technique to reward (i. e., stabilize) desired growth and punish (i. e., dissolve) less desirable growth, they grew an iron ‘ear’ that was capable of distinguishing between two different frequencies of sound.

“We have made an ear and we have made a magnetic receptor. The ear can discriminate two frequencies, one

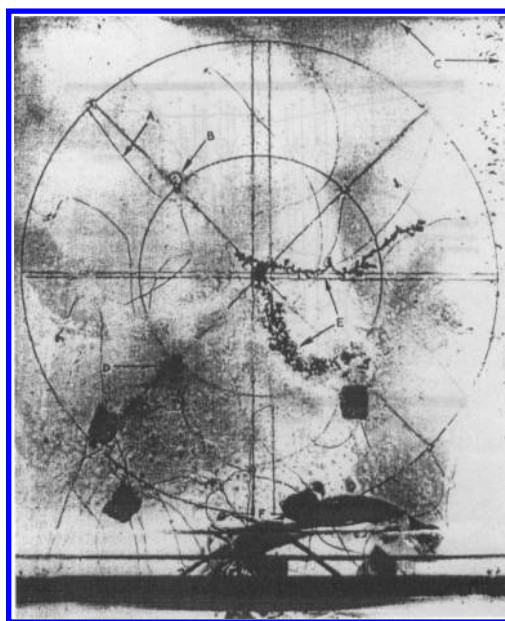


Figure 1: Photograph of electrochemical deposition experiments conducted by Gordon Pask and Stafford Beer in the 1950s. From (Pask, 1958) and (Cariani, 1993).

of the order of fifty cycles per second and the other of the order of one hundred cycles per second. The ‘training’ procedure takes approximately half a day and once having got the ability to recognize sound at all, the ability to recognize and discriminate two sounds comes more rapidly ... The ear, incidentally, looks rather like an ear. It is a gap in the thread structure in which you have fibrils which resonate at the excitation frequency.” (Pask, 1959, p. 261)

From their description, Pask and Beer’s system seems to exhibit a kind of reinforcement learning: it develops specific structures (such as fibrils with a particular resonant frequency) in response to a reward signal (current), without being given precise instructions on what form the structures should take. This is remarkable because it is a relatively sim-

ple physical system and lacks any obvious system for assigning credit or propagating the reward signal. Our goal is to understand what makes this possible, and whether the principles underlying it will generalise to other kinds of physical or dynamical system.

In biological evolution, a directed process of selection biases undirected (i. e., random) genetic variation, resulting in the formation of adaptive structures. Pask and Beer's experiments might be described in similar terms: iron accrues rather randomly, providing a kind of stochastic growth—and direction is given to this change via the selective variation of the applied current. This evolution-inspired description given above seems plausible, but growing an 'ear' merely by changing a voltage applied to a solution of ferrous sulfate seems to almost border on magic. How did this system work and what are the limits of this technique for growing functional structures, using only a reward/punishment-like feedback? If it is a kind of physical instantiation of a search algorithm, what is the search space like—are there lots of local minima or is it better captured as a very high-dimensional space full of neutral networks (Huynen, 1996) that facilitate adaptation to a wide variety of selection pressures? Does the evolutionary metaphor completely describe or explain how these kinds of systems might adapt to different 'selection pressures' (i. e., reward schemes), or are there other dynamics that don't fit so nicely into an evolutionary metaphor.

We do not yet know the answer these questions. Unfortunately, Beer and Pask failed to publish this area of their research in sufficient detail that others might repeat it. In the next section, we present our first steps toward recreating Beer and Pask's results, in the form of a computational model of electrochemical deposition. We use this model to evaluate the possibility of using dynamically modulated voltage so as to steer the growth of the structure in desired forms. But before we delve into the details of our model, we first review some related research which highlights some reasons that we find this system interesting.

Related research

There have been a number of efforts to recreate Pask's work—see e. g. the projects listed in (Boden, 2010, p. 136). In the cases that we are familiar with, there has been success in growing dendritic structures, but nothing as remarkable as the development of a new sensor as reported by Pask and Beer. There have, on the other hand, been a number of interesting results in the investigation of a comparable adaptive self-organising dissipative structure (Nicolis and Prigogine, 1977) known as a ramified charge-transportation networks.

A ramified-charge-transportation-network (RCTN) consists of a number of small steel spheres, placed in a petri dish, and partially submerged in castor oil. A circular grounded electrode runs around the periphery of the dish and a high-voltage ($\approx 20kV$) electrode sits above it. These simple systems demonstrate remarkable self-organizing dy-

namics. Specifically, the spheres self-organize into tree-like structures with topology that depends upon initial configuration of the beads, and that can be radically different based on minor changes in the initial setup (Jun and Hübler, 2005). Once grown these structures display statistically robust network features. For example, when the number of spheres is kept roughly the same, the number of termini and branching points will remain similar despite any topologically different structures (Jun and Hübler, 2005).

Kondepudi et al. (2015) describe the dynamics of these systems in terms of 'energy-seeking' and 'self-healing' behaviours. If a branch is broken, then the system will restore it. Further, the tree will continue moving its branches around the available space to maximize the current conducted by the structure. This can be considered a form of self-preservation, as the flow of electricity is what allows the system to persist in spite of its ordered, low-entropy state. From these and other observations, Kondepudi et al. argue that their overall behaviour can be considered as an end-directed (i. e., purposeful) process (Kondepudi et al., 2015).

The enactive approach (Stewart et al., 2010; Thompson, 2007) takes these kinds of precarious, self-maintaining systems as a conceptual starting point for defining agents (Barandiaran et al., 2009) and related phenomena, such as intrinsic normativity and teleology (Barandiaran and Egbert, 2013). But even if one does not subscribe to these approaches, the ability of these systems to adapt under seemingly arbitrary requirements (e. g., detecting the difference between these two frequencies of sound) makes these systems fascinating models for understanding the adaptability of biological organisms.

It is worth emphasizing the open-endedness of the adaptability of these systems. It would be difficult to argue that an acidic solution of dissolved ferrous sulfate has the inherent propensity to self-organize into a sound-discriminating ear, and yet by applying an electric potential across such a solution in a particular way, Pask & Beer were able to cause it to form into such a functional structure. This is rather remarkable. Does it hint at a not yet fully understood mechanism that might help us to understanding the remarkable open-ended adaptation demonstrated in nature? Cariani (1993, p. 20–21) suggests that by understanding the mechanism underlying Pask's result, we may come to understand how to create systems that can autonomously identify which features of the environment they respond to in a way that is more open-ended than that of conventional neural networks.

“[Conventional learning machines such as NN] improve on their (initial) designs by altering their decision functions contingent upon evaluation of past performance. But even with these machines, the designer must foresee the basic categories of percepts (i.e. primitive features) and actions which will be adequate to solve the problem at hand [...] For real world tasks, however, there is no such set of basic categories that is

defined beforehand, so that in addition to finding appropriate mappings there is also the problem of deciding what the basic categories will be. Essentially, contemporary trainable machines have the freedom to adapt within a set of percept and action categories, but they do not have the freedom to modify those categories. [...] Pask was specifically looking for a machine that would create its own “relevance criteria”, one which would find the observables that it needed to perform a given task. The device [would develop] sensors to choose, independent of the designer, those aspects of its external environment to which it would react. Not only would particular input-output combinations be chosen but the categories of input and of output would be selected by the device itself.”(Cariani, 1993, p. 21)

Some might argue that modern neural networks are capable of selecting their own categorisation schemes, but even if this is granted, they do not (yet) innovate a sensor that wasn't there before.

To summarize, the system studied by Pask and Beer (and the related RCTN structures) are worthy of further study as they (i) demonstrate unusual dynamics; (ii) are comparable to the precarious adaptation of individual biological organisms; and (iii) demonstrate an apparent open-ended ability to adapt. In the next section, we introduce a computational model of Pask and Beer's system. In the following section, we describe our efforts to use a reward-like variation of the applied voltage so as to steer the growth of the dendritic structures.

The reward function that our model optimises is considerably simpler than the task that Pask and Beer set for their system, but our model nevertheless demonstrates a plausible mechanism by which a physical system of this kind can respond to reinforcement signal at all. Our model builds to some extent on ideas presented in [cite Virgo and Harvey 2008], but the mechanism is much more physically realistic.

Model

We now present a model of electrochemical deposition. Using finite difference methods, we use a rectangular 256×128 lattice to simulate a two dimensional space 2 units wide by 1 unit tall. Each position on the lattice is considered to be either a negatively-charged highly conductive solid or an insulating liquid, $M_{i,j} \in \{S, L\}$. The electric potential, ϕ , is calculated across this lattice by fixing the conductive solids (i. e., treating them as boundary conditions) and then solving the Dirichlet problem for the Laplace equation,

$$\nabla^2 \phi = 0 \quad (1)$$

by numerically integrating (using the forward-time centered space method —see e. g., Recktenwald, 2004) the heat equation,

$$\frac{\partial \phi}{\partial t} = \nabla^2 \phi \quad (2)$$

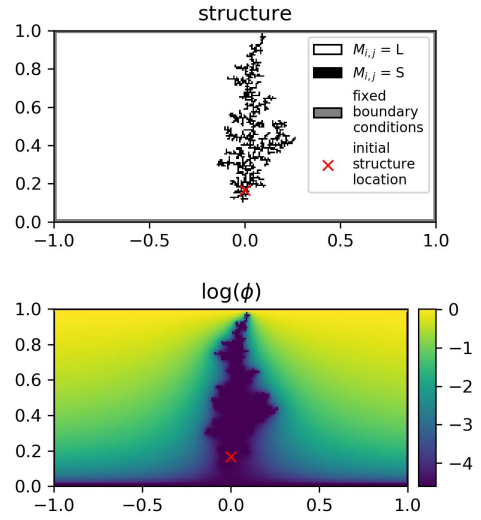


Figure 2: Example final state of a simulation trial, showing the grown structure (top) and its surrounding electric potential field (bottom).

until the system has come (close) to equilibrium i. e., until $\forall \phi : |\phi_n - \phi_{n-1}| < 10^{-3}$, where the subscript n is the current iteration index. When approximating the initial equilibrium for any given run of the simulation, we increase accuracy by reducing the tolerance by an order of magnitude, i. e., $\forall \phi : |\phi_n - \phi_{n-1}| < 10^{-4}$.

The boundary conditions for the Dirichlet problem vary between experiments. We describe this variation in detail below, but in every case, there is a negatively charged conducting solid with a fixed relative potential of 0. We call this ‘the structure’ and it grows and decays via simulated electrochemical deposition and dissolution as described below. In addition, each experiment also always includes a positive boundary condition, that corresponds to an electrode with a fixed relative positive charge.

Each iteration of our simulation begins by approximating the electric-potential field equilibrium as described above. We then determine how the conducting structure will grow or decay. To do so, we identify I , a set of ‘interface cells’: liquid locations with one or more solid locations in their von Neumann neighbourhood.

$$I = \{(i, j) : M_{i,j} = L \wedge \exists S \in \{M_{i-1,j}, M_{i+1,j}, M_{i,j-1}, M_{i,j+1}\}\}. \quad (3)$$

For each interface cell, we calculate the probability that it will become part of the conducting structure. The probability of these ‘constructive’ changes are proportional to ϕ of the interface cell (as ϕ is the potential relative to the structure, which is proportional to electron flow at the interface).

To calculate this probability, use the following equation,

$$P_c = \left\{ \frac{1}{Z} (\phi_{i,j} + \zeta) : (i, j) \in I \right\}, \quad (4)$$

where $\frac{1}{Z}$ is a normalization factor selected such that the sum of all of the probabilities is 1 and ζ is a parameter that scales the relative influence of the voltage compared to entirely random process—as ζ approaches infinity, the probabilities become equal across the interface cells. The results of varying ζ can be seen in Figure 3. Essentially, as ζ is increased, the structure that grows loses its filamentous structure and becomes less sensitive to ϕ -gradients, i. e., stops growing toward high values of ϕ .

Destructive events, where one of the solid neighbours of the interface cell becomes part of the liquid insulating material are also possible. The probabilities of the destructive events are a function of $a(i, j)$, defined as the mean age (time since creation) of the solid cells in the Moore neighbourhood of the interface cell. The assumption underlying this distribution is that over time, the existing structure become smoother and thus less likely to dissolve. This age-based probabilities are calculated according to the following equation.

$$P_d = \left\{ \frac{1}{(1 + a(i, j))^2} : (i, j) \in I \right\}. \quad (5)$$

In Pask and Beer’s experiments voltage was varied so as to reward (i. e., stabilize / cause to grow) or punish (destabilize / dissolve) the structure. In our model we similarly have a reward parameter, $r \in [0, 1]$ that biases the relative likelihood of constructive vs. destructive events. The next event thus selected from the following set:

$$P = \left\{ \frac{rp}{\sum P_c} : p \in P_c \right\} \cup \left\{ \frac{(1-r)p}{\sum P_d} : p \in P_d \right\}. \quad (6)$$

It is important to note that the reward function varies over time (as a function of system state) but not over space. As we shall see the structure tends to grow rather directly toward regions of high ϕ , but it possible to counterdict this energy-seeking behaviour by selectively rewarding certain types of growth and punishing others by varying r .

Once the event is identified, the structure grows or decays as appropriate, the electric potential equilibrium is recalculated, I is updated, the probabilities for the next event are calculated etc., in a repeating iterative manner. Because of the probabilistic selection of events, each iteration corresponds to a different amount of time passage, specifically: Δt is taken from an exponential distribution with the rate parameter, λ , is the sum of the scaled but not normalized probabilities, i. e., $\lambda = r \sum P_c + (r - 1) \sum P_d$.

The stochastic degradation of the structure means that it is not uncommon for sections of the structure to become disconnected. When this happens, sections that are not contiguous with the initial starting point of the structure (which

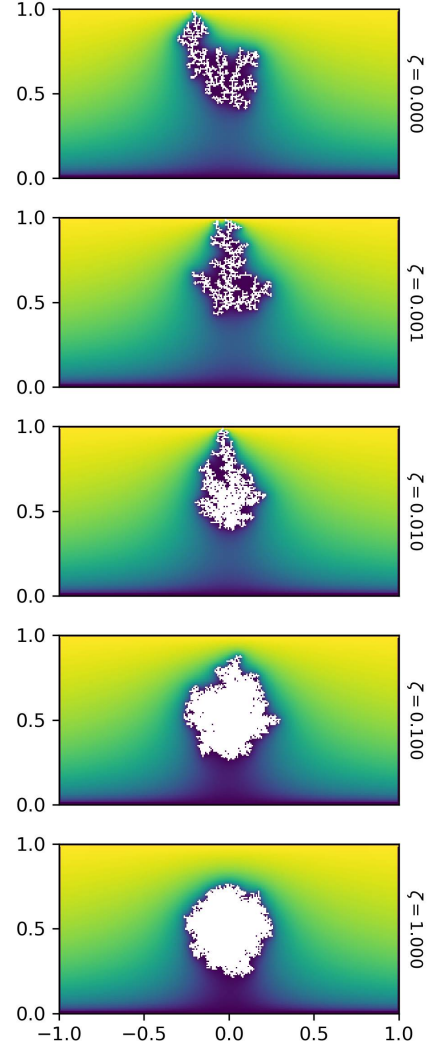


Figure 3: Increasing ζ results in fewer dendrites and reduced electrotaxis. A seed structure at $(0, 0.5)$ responds to a voltage gradient ($\phi_{y=0} = 0, \phi_{y=1} = 1$) in different ways depending upon the relative influence of randomness and voltage as described by simulation parameter ζ .

is interpretable as the negative electrode) are assumed to fall to a neutral potential and rapidly dissolve. This is simulated by removing any solid structure cells that are not connected to the initial starting position of the structure. This is not a physically realistic aspect of our simulation, but rather a simplification. In future work we may model disconnected conducting elements in a more realistic manner. Finally, we make it impossible for the first ‘seed’ cell of the structure to dissolve.

Experiments & Results

We now evaluate different strategies for modulating the reward signal, r . Each experimental reward strategy is a function of x^* , the current mean horizontal position of the top-most part of the structure. To calculate this value, we identify the top-most row of M that contains structure $i^* \equiv \max_i |M_{i,j} = S$, and calculate the mean x of those positions within this row that contain structure, $x^* \equiv \bar{J}$ where $J \equiv \{ \frac{2j}{256} - 1 | M_{i^*,j} = S \}$.

In each case the structure is seeded at the middle of the space, close to the bottom $(x, y) = (0, \frac{1}{6})$. A fixed voltage $\phi = 1$ electrode is simulated as spanning the top edge of the area. To impose a gradient, the bottom edge is clamped to a value of $\phi = 0$ and the left and right boundaries are also clamped as a linear gradient between the top and bottom boundary conditions. Formally: $\phi_{y=1} = 1$; $\phi_{y=0} = 0$ and $\phi_{x=-1} = \phi_{x=1} = x$. In all of the following experiments, we simulate the growth of these structures until either the structure touches the top electrode, or 25,000 iterations have passed. For each of the following experiments, we fix $\zeta = 0$.

Control Conditions. We will first describe the control strategy where reward is fixed at $r = 1$. An example of the type of structure that grows in the control condition can be seen in Figure 2. The initial seed, located at $(0, \frac{1}{6})$ grows rapidly and rather directly to the positive electrode. The bottom frame of Figure 4 shows the average density of 25 structures grown in these conditions. In every case, the trial ends when the structure has grown to the top of the simulation, and the horizontal location of the top of the structure is distributed approximately evenly around the centre of the arena (see the bottom row of Figure 6).

Strategy 1: Simple Reward. The first experimental reward strategy we consider is the modulation of r according to the following simple linear function of x^* . Our goal here is to encourage the structure to grow to the right.

$$r = mx^* + b \quad (7)$$

It is not self-evident which values of parameters m and b will maximise our influence of the structure. Figure 5 shows the conducted a systematic survey to investigate the influence of these parameters. Of the values tested, the parameters that maximised the mean rightward growth of the structures were $(m, b) = 5.06, 0.5$ and it is these parameters that were used to generate the ‘Simple Reward’ portions of Figures 4 and 6.

This simple strategy succeeds at influencing the growth of the structure. Once the structure has grown a little bit to the right of its initial location, r increases, and provided the structure does not grow back to the left, r will remain high enough for the structure to continue to grow. In other words, after an initial growth to the right, further rightward growth

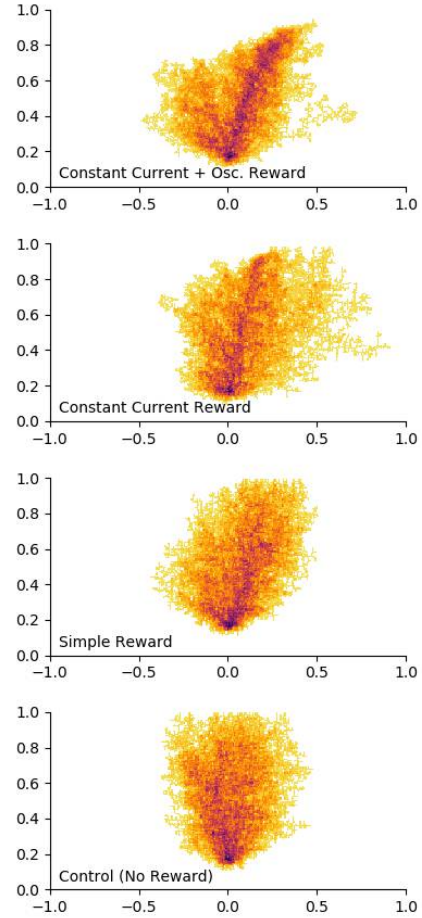


Figure 4: Superposition of the final states of 25 trials for the best found parameterization of each indicated strategy. Darker areas indicate locations where structure existed at the end of a greater number of trials.

is unnecessary for the system to grow, and so it grows, attracted by the higher values of ϕ close to the top of the arena. Is it possible to do better?

Strategy 2: Constant Current. Pask generally refers to current rather than voltage when describing this experiment, and so there has been some speculation that they used a constant current device that regulates voltage so that the total current flowing between the two electrodes is constant (or kept below some maximum)—see e.g. the description in (Bird and Di Paolo, 2008, p. 201). As conductive structure grows between the electrodes, resistance decreases. If the applied voltage were fixed, the current flowing through the system would thus increase as the resistance dropped. The constant current regulator is a simple feedback control device that regulates the applied voltage so that current is con-

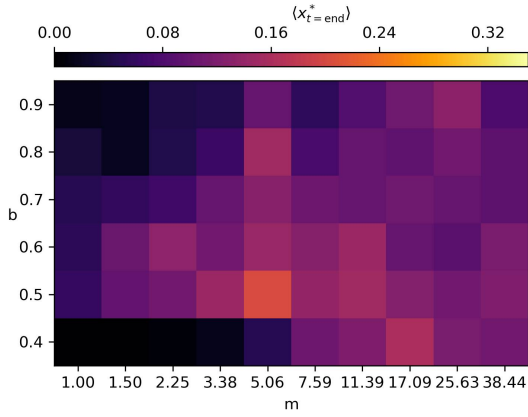


Figure 5: Systematic survey of simple-reward parameters. Values indicate the mean x^* found for the indicated parameter values, with brighter values corresponding to greater influence of growth.

stant. Essentially if the current is greater than a target value, the applied voltage is decreased and vice-versa according to negative feedback-like relationship similar to $\frac{dV}{dt} = \theta - A$, where V is the voltage, θ is the target current (selected by Pask) and A is the current.

For our second experimental reward strategy we simulate a constant-current feedback mechanism so that current is maintained at a target value (θ),

$$r = \frac{1}{2} + \theta - A, \quad (8)$$

where A is the total current between the positive electrode and the structure, which is calculated by summing the ϕ values at the interface,

$$A = \sum_{(i,j) \in I} \phi_{i,j}. \quad (9)$$

The target current (θ) is then dynamically modulated according to how far right the structure has grown (again approximated by x^*), thus:

$$\theta = \frac{1}{2} + m \left(x^* - \frac{1}{2} \right). \quad (10)$$

Once again it is not clear how to select a value for parameter m in this strategy. We experimented with $m \in \{10, 20, 30, 40, 50, 60\}$. The most effective value at maximising the mean x^* was $m = 40$ maximised, and so we used that parameter value to generate the data plotted for this strategy in Figures 4 and 6. In these figures we can see that the constant current strategy produces a distribution of x^* with a mean that is greater than the simple strategy, but note that this is not a statistically significant increase (see Table 1).

	Simple	Const. Curr.	CC + Osc.
Control	$p < \mathbf{0.001}$	$p < \mathbf{0.001}$	$p < \mathbf{0.001}$
Simple		$p < 0.141$	$p < \mathbf{0.006}$
Const. Curr.			$p < 0.609$

Table 1: Tukey’s test. This table indicates chance that variation between means of the data plotted in Figure 6 is due to chance. Bold entries are considered statistically significant.

Strategy 3. Constant Current with Exploratory Oscillations. While observing the simulations of the constant current strategy, there was often a feeling of wishing that the structure would ‘experiment’ more—i. e., try out different random configurations and keep those that increase the reward signal. To encourage this kind of exploration, we added an externally driven oscillation to the reward signal to produce our final strategy.

The reward function is the same as in the Constant Current strategy, except that we update Equation 8 to include a sinusoidal function of the current iteration of the simulation, τ . It would be more appropriate to have this be a function of time rather than iteration, and this will be an improvement that we make in future work.

$$r = \frac{1}{2} + \theta - A + n \cos(2\pi\tau/p), \quad (11)$$

Once again, the control strategy includes free parameters, and we used a systematic survey to search for those that are more effective. Figure 7 show the results of this survey. There is no clear trend among these parameters, but they all perform well compared to the previous reward strategies. We selected the best performing parameters $(n, p) = (0.1, 100)$ to generate the data plotted in Figures 4 and 6. The distribution of x^* generated by this strategy now significantly outperforms the simple reward mechanism ($p < 0.001$ —see Table 1).

Observations & Discussion

We have presented a new model for exploring the electrochemical deposition system investigated by Pask & Beer in the 1950s. The model has helped us to understand how by the selective rewarding of particular patterns of growth, it is possible to influence or ‘steer’ the dissipative structures that grow in these conditions. This is a potentially significant result, because it suggests a novel, and simple, mechanism through which physical systems can respond to reinforcement signals, potentially producing complex, organised structures as a consequence.

Our ability to guide the form of these structures in our model is not absolute and it is interesting to consider the source of any limitations and thus how they might be overcome. One limitation may come from the energetic gradients inherent in our simulation whereby the conducting structure naturally grows up ϕ gradients. Each reward strategy rewarded growth orthogonal to the ϕ gradients, but the

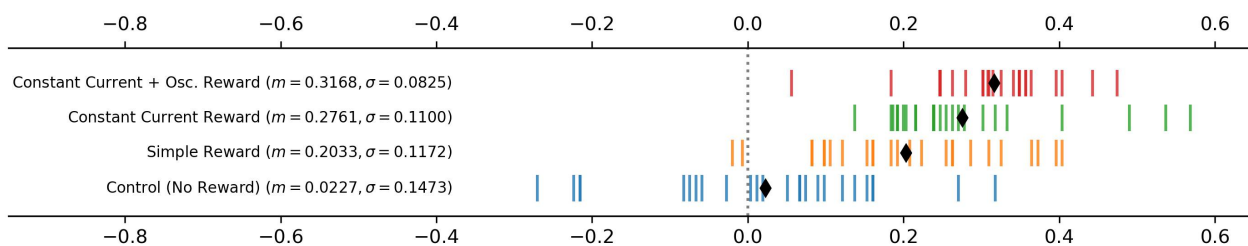


Figure 6: Comparison of best parameterization for each evaluated strategy. Each row indicates the x position of the top-most structure element for 25 trials of indicated reward strategy.

structures all also (unsurprisingly) responded primarily to the ϕ gradient by growing upwards. Decreasing the influence of the ϕ gradient might be expected to improve the steer-ability. One way to do this would be to increase ζ . In the physical experiment, this would correspond with decreasing the relative voltage between the electrodes. But decreasing the voltage excessively would mean that no deposition would occur. A constant current mechanism might allow the voltage to remain high, while decreasing the ‘attractive force’ of the positive electrode. To speculate: the constant current mechanism partially neutralizes the attractiveness of the ϕ gradient, as growth toward the positive electrode reduces resistance, which increases the current, which would cause the constant current mechanism to decrease the applied voltage. If properly tuned, such a mechanism might mean that the growth of the structure responds *only* to the reward function (and not also to the ϕ gradient as is currently the case). In this way, a well-designed reward mechanism would ‘flatten’ the landscape of possible structures, facilitating the growth of those structures that maximise the reward signal.

It is also interesting to consider this ‘flattening’ of the search space in the context of genetic evolution, where the search space of nucleotide sequences is essentially flat (i. e.,

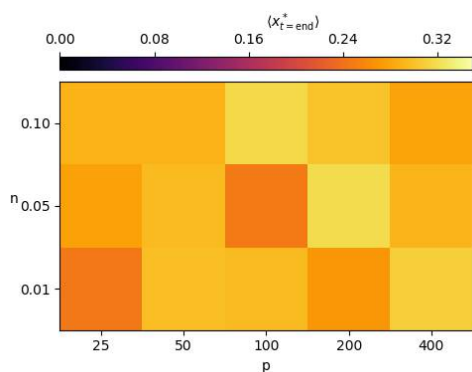


Figure 7: Systematic survey of simple-reward parameters. Values indicate the mean x^* found for the indicated parameter values, with brighter values corresponding to greater influence of growth.

there is little inherent cost for choosing a adenine or a guanine), facilitating the ability of evolution to search the space of polypeptide sequences unabated.

Comparison to evolution. In the introduction, we compared random but selected growth of the structure to the selection of random mutations in Darwinian evolution. Metaphors like this are useful both for identifying similarities between systems, and for highlighting differences. One such difference that we noticed in the simulation is that when a new branch begins to grow, it tends to grow in that same direction for some time. This inertia-like effect may be due to the tendency of new structures to grow into areas where they are more exposed, and thus subject to higher voltage and thus more likely to grow further—a kind of autocatalytic growth. It may be that increasing the reward signal during one of these may further accelerate this tendency allowing for a more instructive or directive kind of reward mechanism along the lines of “do more of that” rather than the post-hoc reward, “what you just did was good, keep it”. Darwinian evolution has no explicit inertia mechanism such as that just described, but it is interesting to reverse the metaphor and to consider that there are occasions when a new mutation opens up a set of possible environmental interaction which encourages further mutations.

Comparison to metabolism and biological individuality. It is also interesting to consider one of these structures as a model of a biological individual performing a metabolism-based behaviour. The dendritic structure is a dissipative structure that relies upon the dissipation of energy (the flow of electricity) to persist. It reconfigures itself to amplify or stabilize this flow of energy, and when conditions are right, this adaptation can respond, not just to physical energy gradients (control case), but to more complex requirements (shown here in simulation and in Pask and Beer’s original experiments). A number of other dissipative structures similarly act so as to satisfy their own needs —see e. g. RCTN discussed in introduction and motile oil-droplets (Hanczyc, 2011). These physical systems, like some bacteria (Egbert et al., 2010) are responding essentially to their own rates of self-construction in what is referred to as metabolism-based

behaviour, which can facilitate adaptation and evolution in a number of ways (Egbert et al., 2012; Egbert and Pérez-Mercader, 2016).

Origins of life. Pask’s electrochemical experiments seem to demonstrate the emergence of functional components (for example, the vibrating filaments in the ‘ear’), without them needing to be ‘designed in’ by a human engineer. One can find examples of this in other places, such as the emergence of new traits in evolution, or the emergence neurons that perform specific filtering operations when training a deep neural network, for example. But Pask’s case is remarkable because it occurs in such a simple physical system, purely as the result of the physical processes of fluctuating growth and decay of filaments, in response to a reward function.

Although there are some parallels with evolution, this simple physical mechanism differs in that there is no need for large, complex polymer molecules to be produced. If evolution is not the only mechanism by which complex functional structures can arise in the natural world, it becomes possible that evolution as we know it is the *result* of a dynamical process, and not just its starting point. The emergence of Paskian growth seems much easier than the emergence of complex biomolecules, and perhaps mechanisms resembling it played a role in steering the abiotic world on its way to the emergence of biology.

Self-organizing steerable self-organizing systems In our model there are a number of parameters that would influence the extent to which the self-organization is controllable or ‘steerable’. That Pask and Beer were capable of finding conditions suitable for steering the self-organization of an ‘ear’ is remarkable. Instead of trying to identify effective parameter regimes directly, it may be more effective to identify negative-feedback-like mechanisms that automatically regulate parameter regimes so as to produce steerable self-organization. One example of such a mechanism may be the constant-current mechanism as described above.

Future work We have proposed a mechanism through which a simple electrochemical system could plausibly respond to a reward signal. However, the task we set our system, of growing in a particular direction, was very simple in comparison to the task of “growing an ear” that was achieved by Pask and Beer. It will be important in future work to show that this kind of system can solve more difficult tasks. It will be equally important to build a better theoretical understanding, in order to understand whether other kinds of physical system can exhibit similar reinforcement learning behaviour.

Acknowledgements

Preparation of this manuscript was generously supported by a Royal Society Te Apārangi, Marsden Fund Te Pūtea Rangahau A Marsden grant awarded to Matthew Egbert in 2017 (Project ID: 17-UOA-196).

References

- Barandiaran, X. E., Di Paolo, E., and Rohde, M. (2009). Defining Agency: Individuality, Normativity, Asymmetry, and Spatio-temporality in Action. *Adaptive Behavior*, 17(5):367–386.
- Barandiaran, X. E. and Egbert, M. D. (2013). Norm-Establishing and Norm-Following in Autonomous Agency. *Artificial Life*, 20(1):5–28.
- Bird, J. and Di Paolo, E. (2008). Gordon Pask and His Maverick Machines. In Husbands, P., Holland, O., and Wheeler, M., editors, *The Mechanical Mind in History*, pages 185–211. The MIT Press.
- Boden, M. A. (2010). *Creativity and Art: Three Roads to Surprise*. Oxford University Press. Google-Books-ID: MS0UDAAAQBAJ.
- Cariani, P. (1993). To evolve an ear. Epistemological implications of Gordon Pask’s electrochemical devices. *Systems Research*, 10(3):19–33.
- Egbert, M. D., Barandiaran, X. E., and Di Paolo, E. A. (2010). A Minimal Model of Metabolism-Based Chemotaxis. *PLoS Computational Biology*, 6(12):e1001004.
- Egbert, M. D., Barandiaran, X. E., and Di Paolo, E. A. (2012). Behavioral Metabolism: The adaptive and evolutionary potential of metabolism-based chemotaxis. *Artificial Life*, 18(1).
- Egbert, M. D. and Pérez-Mercader, J. (2016). Adapting to Adaptations: Behavioural Strategies that are Robust to Mutations and Other Organisational-Transformations. *Scientific Reports*, 6:18963.
- Hanczyc, M. M. (2011). Metabolism and Motility in Prebiotic Structures. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1580):2885–2893.
- Huynen, M. A. (1996). Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution*, 43(3):165–169.
- Jun, J. K. and Hübler, A. H. (2005). Formation and structure of ramified charge transportation networks in an electromechanical system. *Proceedings of the National Academy of Sciences of the United States of America*, 102(3):536–540.
- Kondepudi, D., Kay, B., and Dixon, J. (2015). End-directed evolution and the emergence of energy-seeking behavior in a complex system. *Physical Review E*, 91(5):050902.
- Nicolis, G. and Prigogine, I. (1977). *Self-organization in nonequilibrium systems: from dissipative structures to order through fluctuations*. Wiley, New York, NY.
- Pask, G. (1958). Physical analogues to the growth of a concept. pages 765–794, National Physical Laboratory, London. H.M.S.O.
- Pask, G. (1959). The Natural History of Networks. In Marshall C. Yovits and Scott Cameron, editors, *Self-Organizing Systems: Proceedings of an Interdisciplinary Conference*, volume 2, pages 232–263. Pergamon Press.
- Recktenwald, G. W. (2004). Finite-difference approximations to the heat equation. *Mechanical Engineering*, 10:1–27.
- Stewart, J., Gapenne, O., and Di Paolo, E. A. (2010). *Enaction: Toward a New Paradigm for Cognitive Science*. MIT Press.
- Thompson, E. (2007). *Mind in Life*. Harvard University Press, London, England.

Complex Systems and Artificial Life: A Decade's Overview

Thomas McAtee¹ and Claudia Szabo¹

¹The University of Adelaide, Adelaide, South Australia 5005
thomas.mcatee@adelaide.edu.au, claudia.szabo@adelaide.edu.au

Abstract

Artificial Life models and algorithms are informed by natural and biological processes and phenomena. Artificial Life finds particular use in simulating large, complex systems such as large scale ecosystems or social networks, where the interaction between system entities may give rise to emergent behaviours. Despite the increasing popularity and ubiquitous nature of complex systems, the extent of which artificial life approaches are considered in complex systems modelling and their application across complex systems domains is still unclear. To better understand the overlap between artificial life and complex systems, we conducted a systematic literature review of last decade's artificial life research that had a complex system focus. We performed an automated search of all relevant databases and identified 538 initial papers, with 194 in the candidate set, resulting in 115 primary studies. Our results show that the three most frequent application domains are simulation, followed by social modelling, and biological modelling. We find a plethora of paradigms that can be broadly classified into three main categories, namely, biological, social, and hybrid. We identify the artificial life paradigms that are used to generate the most common complex systems properties as well as a number of research challenges that are critical for the growth of both artificial life and complex systems modelling.

Introduction

Artificial Life is a set of models and approaches adapted and inspired by naturally occurring phenomena and processes (Langton et al., 1989). These paradigms can be applied to a diverse range of problems, such as ecological modelling (Punithan et al., 2011), evolving artificial creatures (Loula et al., 2010), combat simulation (Yu and Zhao, 2010), and modelling application performance in proprietary app stores (Cocco et al., 2014). Complex systems are comprised of autonomous entities with complex behaviours, whose interactions can lead to unexpected and emergent properties (Szabo et al., 2014; Mittal, 2013). Complex adaptive systems (CAS) are a type of complex system where entities and the environment are encouraged to adapt and interact with each other in order to achieve desired properties (Holland, 2006) and provide a more realistic abstraction of real-life scenarios (North et al., 2013). Such systems have become ubiq-

uitous in domains such as social networks, supply chains, health-care networks, smart-cities and smart-grids, the "Internet of Things", and the Internet itself (North et al., 2013).

Artificial life approaches and paradigms can be easily studied and analysed with a complex systems lens, thus allowing a focus on important properties such as self-organisation (Holland, 2006), emergence (Szabo et al., 2014), adaptation (Holland, 2006), modularity (Holland, 2006) and criticality among others. *Self-organization* occurs when entities interact to achieve a specific goal or to move the system in a different state (Holland, 2006; Mittal, 2013). *Emergence* occurs when entities organize to behave collectively, leading to the creation of an unpredictable *macro* state that cannot be decomposed into its *micro* components (Szabo et al., 2014). Some systems can exhibit emergent behavior without self-organization, such as a stationary gas (Mittal, 2013). Emergence has been observed in a plethora of systems, such as communities forming in social networks, formation of ant colonies, and rigid cellular structures (Birdsey et al., 2015). *Autonomy* is exhibited when entities within a system selectively act upon their environment without external control and is fundamental to the demonstration of emergence and self-organisation (Froese et al., 2007). *Adaptation* refers to the individual adaptive processes of system entities and environments as well as to the adaptive ability of the system as a whole (Holland, 2006). When *modularity* is employed, entities or the environment must be comprised of sub-entities that determine the behaviors and actions of the parent entity (Holland, 2006). *Criticality* refers to the time period before the system enters a stable, unstable, or emergent state. In many systems, criticality is observed at the edge of chaos or as a decision point.

Employing a complex systems perspective and considering the above properties explicitly would further the applicability of artificial life paradigms to a variety of domains and would test their use under complex, large-scale scenarios, thus potentially significantly developing the field. Conversely, a better understanding of the artificial life paradigms that would facilitate the appearance of specific complex systems properties will aid significantly in their design, such as,

for example, when designing systems with a specific desired emergent property (Mittal, 2013). While a large number of works have looked at modelling complex systems including artificial life paradigms, the extent of their use and applicability is yet unknown. To address this gap, we perform a systematic literature review of artificial life research that had a complex system or complex adaptive system focus.

Related Work

Despite existing literature reviews regarding individual models (Santé et al., 2010), complex system properties (Froese et al., 2007), and artificial life paradigms (Emmeche, 1998), to the best of our knowledge, there has been no review focused on the use of artificial life paradigms in modelling complex systems. In the following we discuss several literature reviews focused on specific artificial life modelling paradigms or applications.

Work by Bedau (Bedau, 2003) analyses established artificial life advances and paradigms up to 2003, providing a rough timeline of artificial life developments starting with Langton's initial seminal work (Langton et al., 1989) and progressing to the then-state of the art advances in evolutionary robotics, swarm intelligence, and evolutionary language modelling. Bedau identifies 13 paradigms, including two that are strongly based on complex adaptive system properties, namely, self-replication and self-organisation, with a discussion that considers the potential applicability of adaptation to future artificial life research, however the study fails to give an in-depth analysis of how complex systems are modelled using artificial life paradigms and approaches.

A study by Bousquet et al. (Bousquet and Le Page, 2004) reviews the application of multi-agent simulations (MAS) to the modelling of ecosystem management. They identify that scientists working in ecosystem management need to examine the interactions between ecological and social dynamics, and that MAS provides a useful model for examining the effects of the convergence of these dynamics.

A later work by Froese et al. (Froese et al., 2007) analyses the use and definitions of autonomy within artificial life applications. The study notes that there is no consensus regarding the definition of 'autonomy' and proposes the introduction of a conceptual distinction between the classes of behavioural and constitutive autonomy. The provided definitions assert that behavioural autonomy relates to the capacity of a system for stable and/or flexible interaction with its environment, whereas constitutive autonomy relates to a system's capacity for autopoiesis, which is considered by the authors to have the undesirable quality of restricting the concept of autonomy to organisms. This classification scheme is used to demonstrate that systems at the date of the study publication (2007) had increased in autonomy over the systems published ten years prior. A review by Sante et al. (Santé et al., 2010) focused on the application of cellular automata models to the simulation of real-world urban pro-

cesses such as urban planning or modelling urban evolution and described the strengths, flaws, and challenges of using each model for different application domains.

Methodology

Our work aims to identify the overlap between artificial life and complex systems, specifically to better understand how artificial life approaches are considered in complex systems design, modelling, or analysis and conversely whether artificial life approaches consider complex systems perspectives or properties. In the following, we use the term complex systems broadly, to cover both complex systems as well as complex adaptive systems.

Identification of Research

We conducted a systematic literature review by adapting the guidelines proposed by Kitchenham (Kitchenham, 2004) and following a highly structured process that involved (i) an initial trial search to determine selection and exclusion criteria as well as the search string, (ii) relevant database search, (iii) selection of studies, (iv) filtering the studies by evaluating their pertinence, (v) extracting data using our data extraction form and (vi) synthesising the results.

The search terms were applied to the title, abstract, and keyword fields of the ACM Full Text Collection, IEEE Explore, ScienceDirect, SpringerLink and Scopus databases, identifying 68, 19, 433 and 18 papers respectively (538 total). Following our established selection/exclusion criteria discussed below, we read through a target set of 10 papers to determine their inclusion in the set and to calibrate our process. The inter-rater reliability of this process was measured with the Fleiss-Davies kappa (Davies and Fleiss, 1982), which measures the agreement when a fixed set of raters classify a number of items into a fixed set of categories. The Fleiss-Davies kappa for individual classification was 80%, which is considered excellent (Banerjee et al., 1999). In the next stage, we selected the papers that formed the basis for the review. The search results were divided among the authors, who examined each title and abstract, and the corresponding full paper if required, to determine its relevance.

Search String and Inclusion/Exclusion Criteria

The search string used boolean operators to refine the search and was: (**'Artificial Life'** AND (**'Complex Systems'** OR **'Complex Adaptive Systems'**)) AND (**model** OR **analysis**) and was adapted to the specific database. Only papers published since 2008 were considered. The inclusion/exclusion criteria were:

- *Topic* - the paper must design or use, implement, and evaluate at least one artificial life paradigm, and the system under study must be a complex system. Surveys, reviews, and position papers were excluded.
- *Length* ≥ 5 pages - short papers were excluded.

- *Language* - only English papers were included.
- *Peer-reviewed* - only papers that have been peer-reviewed were included.
- *Exclusions* - papers that contained only wet- or hardware-based Artificial Life models (without accompanying software-based models) were excluded.

The above search string was formulated to find studies that utilise both Complex Systems and Artificial Life in their design and implementation. The focus on the overlap between complex systems and artificial life means that a range of paradigms that may not have been utilised in conjunction with complex systems were not reviewed.

Data Items

An overview of the data items extracted from the papers is presented below.

Modelling approach - Captures the modelling approached employed.

Analysis approach - Captures the type of analysis used to evaluate the model.

ALife paradigm employed - Captures the artificial life paradigms employed in each paper. These paradigms may include natural or biological behaviours (such as foraging, predation), naturally occurring phenomena (such as ecosystems, evolution, or protein-folding), or social behaviours (such as elections, economic exchange, or semiosis).

Complex Adaptive Systems Properties - Captures any CAS properties that are identified by the paper authors.

Further Applications - Captures whether the paper authors identified any further domains within which the paper topic could be applied.

Scalability Considered - Captures whether the paper authors considered scalability in their analysis.

Challenges & limitations - Captures the challenges or limitations faced by the approach as stated by the paper authors.

Challenge type - Captures the type of challenges identified by paper authors.

Results

We present an overview of the identified data items from the 115 primary studies as well as discuss answers to our research questions.

Application Domains

Our analysis identified 32 unique application domains, 19 of which occurred more than once, as shown in Table 1. 16 application domains occurred only once, such as data mining (de Buitelir et al., 2012), and logistic networks design (Otto and Bannenberg, 2010) among others. The most frequently identified application domain was Simulation (representing nearly 13% of all primary studies), where papers

Application Domain	Frequency
Simulation	15
Social Modelling	13
Biological Modelling	12
Robotics	9
Linguistics	7
Optimisation	6
Complex System Analysis	5
Artificial Life Modelling	5
Music Modelling	4
Disease Modelling	4
Automated Design	4
Markets	3
Ecosystem Modelling	3
Routing	2
Video Games	2
Genetics	2
Military/Tactical Modelling	2
Life History Modelling	2
Pattern Recognition and Generation	2
Other	13

Table 1: Application Domains (N=115)

demonstrated the use of artificial life paradigms to optimise and improve existing simulation and simulation-related practices. For example, Seth's work explored the use of Granger Causality to detect the autonomy or emergence of a complex system (Seth, 2010), while the work by Kirshenbaum et al. demonstrated the use of simulation to educate students on Swarm Intelligence (Kirshenbaum et al., 2008). Social Modelling (11.3%) was the second most frequently utilised application domain, where social structures, networks, and situations were investigated using artificial life paradigms. The third most frequently investigated application domain is Biological Modelling (10.4%), where biological processes and phenomena were modelled and evaluated. For example, researchers modelled cancer growth using cellular automata (Monteagudo and Santos Reyes, 2013) and several primary studies modelled the phenomena of protein folding using neural networks and L-Systems (Varela and Santos, 2018).

Modelling & Analysis Approaches

We identified 23 unique modeling approaches in the primary study set. 57.4% of the primary studies employed *Agent-based Modelling (ABM)* as a modelling paradigm. ABM models simulate the actions and interactions of autonomous agents with the intent to assess the system-wide results of these interactions. *Cellular Automata* was the second most frequently used modelling approach with 45.2% of primary studies using it. Cellular Automata are composed of a (usually two dimensional) lattice of cells, each of which is configured to be a particular state and can affect the states

Model Type	Frequency
Agent-based Modelling	64
Cellular Automata	21
Ant Colony Systems	5
Evolutionary Algorithm	3
Neural Network	2
Robotics	2
P-Systems	2
Analytical	2
Graphs	2
Swarm Intelligence	2
Other	10

Table 2: Modelling Approaches Frequencies (N=115)

of neighbouring cells. The third most commonly utilised modelling approach was *Ant Colony* (AC), which was used employed significantly less than either of the previous two models at 4.3%. The AC model uses ant colony-inspired methods to optimise problems that can be simplified to graph representations.

The models were analysed using simulation (88.7%), analytical methods or proofs (11.3%), and once through the installation of a physical prototype (0.87%).

Artificial Life Paradigms Used

Artificial Life derives many of its paradigms from biological and social phenomena. To better understand the spread of paradigms across the papers in our primary study set, we classified the 93 identified paradigms into three main categories, namely, **Social**, **Biological**, and **Hybrid**¹. A significant number of these (60, or 64.52%) were only identified once, and have been aggregated into the *Other* categorisation at the bottom of each Paradigm table.

Social - Paradigms that are based on social processes or phenomena, for example *Social structures*, *Artificial Societies*, and *Communication*.

Biological - Paradigms that are based on biological & physiological processes or phenomena, such as *Pheromones*, *Genetics*, and *Metabolism*.

Hybrid - Hybrid paradigms are syntheses of social and biological paradigms. This category also contains paradigms that do not neatly fit into biological or social classifications, such as *Swarm intelligence*, which has both a social and biological basis or *Pathfinding* which, while a natural process, has a clear basis in neither.

Biological Paradigms The three most commonly employed biological paradigms are Evolution, Predation, and Pheromones as shown in Table 3. Evolution was utilised as

¹Space constraints prevent us from providing the full list of papers that include a specific paradigm. We include the full tables and reference list here: <https://tinyurl.com/alife-paradigms>

Biological Paradigm	Frequency
Evolution	39
Predation	11
Pheromones	9
Reproduction	8
Foraging	7
Artificial chemistry	4
Bacterial-based algorithms	2
Energy flow	2
Genetics	2
Metabolism	2
Morphogenesis	2
Protein Folding	2
Starvation	2
Other	21

Table 3: Biological Paradigm Frequencies

a catch-all class for paradigms that relate to modelling evolution, or utilise evolutionary computation or evolutionary concepts (such as evolutionary dynamics or morphological evolution) (Joachimczak et al., 2013). Predation refers to a predatory relationship between at least two agent classes within a complex system; one agent, the predator, consumes agents from the prey class (Seth, 2010). The pheromone paradigm relates to the use of trails left by agents in the style of biological pheromones released by ants and other creatures.

Evolution is the most commonly employed biological paradigm with 33.9% of the primary studies utilising it in some form. The frequency of *Evolution* paradigms is disproportionate when compared with all other paradigms, as the next most commonly employed paradigm across all paradigms is *Swarm Intelligence*, which was only employed in 14.8% of the primary studies, and the third most common paradigm (*Predation*) being utilised in 9.6% of the primary studies. The Other category captures a wide set of infrequently used paradigms, such as Apoptosis, Biological Growth, Exaptation and Inheritance among others. 69.57% of the primary set papers employed a biological paradigm.

Social Paradigms Social paradigms relate to phenomena that occur through agents directly interacting with each other. The most commonly employed social paradigms in the primary set are Cooperation, Semiosis, and Economic exchange as shown in Table 4. 21.74% of the primary set papers employed a social paradigm. Cooperation refers to agents helping each in order to reach a mutually beneficent outcome such as (Oswald and Schmickl, 2017). Semiosis refers to the emergence of linguistic constructs through the interaction between agents and their environment (Shibuya et al., 2018). Economic exchange refers to the exchange of resources between agents.

Hybrid Paradigms Hybrid paradigms utilise qualities from multiple other paradigm classes. For example, while Swarm

Social Paradigm	Frequency
Cooperation	5
Semiosis	4
Ant colony	3
Economic exchange	3
Flocking	3
Competition	2
Crowd movement	2
Imitation	2
Social behaviours	2
Social networks	2
Other	19

Table 4: Social Paradigm Frequencies

Intelligence is a social phenomenon due to its mechanical reliance on interactions between a swarm of agents, the modelling of swarm intelligence tends to utilise biological mechanics such as pheromones for interaction.

The most frequently utilised hybrid paradigms listed in Table 5 are Swarm Intelligence, Learning, and Population Dynamics. Swarm Intelligence is the collective behaviour of a decentralised group of self-organised agents, exemplified in nature by the group activities of ants and bees (von Mammen and Jacob, 2009). Learning is a phenomena by which an agent or creature acquires knowledge about themselves or their environment through experience (Azumagakito et al., 2011). Population dynamics refers to the examination of populations in dynamical systems, with regards to how particular sub-populations are affected or affect the greater system (Bornhofen and Lattaud, 2009). 59.13% of papers used a hybrid paradigm.

Hybrid Paradigm	Frequency
Swarm Intelligence	17
Learning	11
Population Dynamics	9
Co-Evolution	5
Disease Model	3
Neural Network	2
Migration	2
Multiple Particle Interaction	2
Pathfinding	2
Stigmergy	2
Other	20

Table 5: Hybrid Paradigms Frequencies

Paradigm Frequency By Year

Figure 1 shows how often artificial life paradigms were used to model complex systems in each year of the research window. We include only paradigms with an aggregated frequency greater than four. We observe that *evolution* has been frequently employed in the past decade with a peak of use

occurring in 2009. In addition, *pheromones* were used with increasing frequency until 2012.

Limitations

34.78% (40 out of 115) of the primary studies reported a form of challenge or limitation when discussing the outcome of their research. The most frequently reported limitation is related to modelling (18.96%), with paper authors citing challenges in merging Artificial Life paradigms with conventional techniques such as manufacturing methodologies with biological paradigms (Leitao, 2009; Monteagudo and Santos Reyes, 2013), challenges typical to Complex Systems such as the lack of reliability in the occurrence of emergence (Lopez, 2010), and challenges in accurately and efficiently modelling simulation environments (Azumagakito et al., 2011; Isidoro et al., 2011; Bornhofen and Lattaud, 2009). The second most frequently reported limitation is in the area of analysis (7.76% of papers), where authors referenced difficulties in the visualisation of results or models (Punithan et al., 2011), limited analytical scale (Janecek et al., 2013; Niazi, 2014), difficulty in developing analytical metrics (Taylor and Cody, 2015), and limited analysis leading to ambiguities regarding how models operate (Oswald and Schmickl, 2017). *Implementation* and *Validation* were equally reported to be challenging (4.34%). The challenges of implementation led to limitations in scalability (Krol and Popiela, 2009) and parameterisation of complex models (Yamamoto and Miorandi, 2010). The challenges of validation led to difficulties with ensuring that ensuring model behaviour is correct.

Scalability

Large-scale systems are becoming ubiquitous and as such there is a need for artificial life approaches to consider scale both in terms of the number of entities considered in the model or system, but also in terms of the number of attributes and the complexities of entity behaviours and interactions. 19 papers (16.38% of the total number of papers), evaluated the scalability of the proposed approach.

Complex Adaptive Systems Properties

22 complex adaptive systems (CAS) properties were identified in the primary study set, as shown in Table 6. Our analysis aims also to identify the artificial life paradigms that are used to facilitate specific complex systems properties, and Table 6 shows the most frequent paradigms per property. *Emergence* was the most frequently considered property with 57.39% of the papers analysing emergent properties or being designed to achieve emergence. Papers reported using 90 paradigms to generate emergence, mostly from the Biological and Hybrid paradigm categories (50 (55.56%) and 28 (31.11%), respectively). The most common paradigms for generating emergence were evolution (Otto and Bannenber, 2010; von Mammen and Jacob, 2009), predation

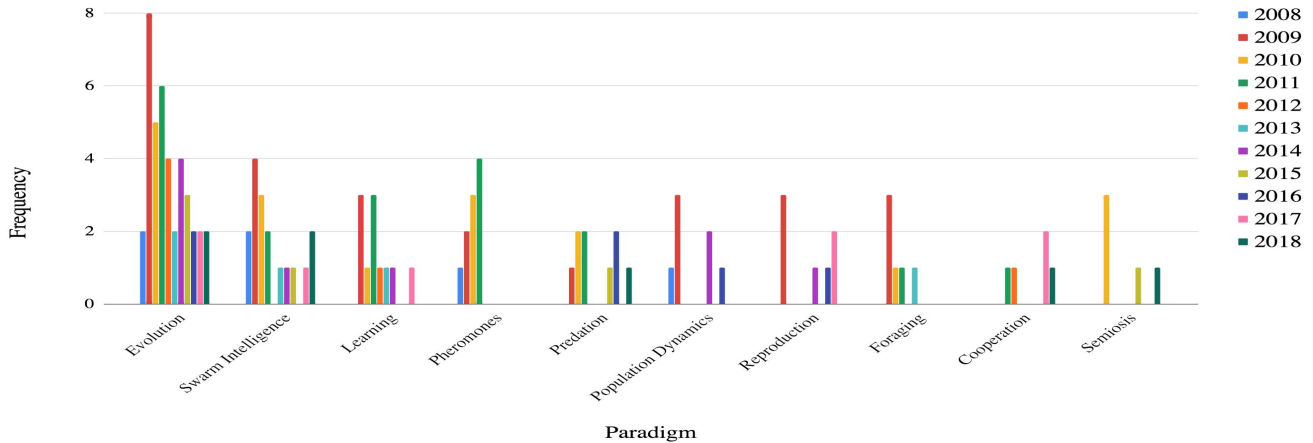


Figure 1: Paradigm Frequencies (2008 - 2018) (N = 115)

(Loula et al., 2010; Lopez, 2010), and swarm intelligence (Leitao, 2009; von Mammen and Jacob, 2009).

Property Name	Freq.	Common Paradigms
Emergence	66	Evolution (40.9%), Predation (13.6%), Swarm Intelligence (10.6%), Reproduction (10.6%)
Self-Organisation	33	Evolution (75.7%), Swarm Intelligence (30.3%), Predation (21.21%)
Adaptation	21	Evolution (42.8%), Learning (19%), Population Dynamics (9.5%)
Stability	4	Evolution (50%), Local Dynamics (25%), Ecosystem Modelling (25%)
Self-Regulating	2	Cooperation (50%), Pheromones (50%), Reproduction (50%)
Self-Repairing	2	Artificial Endocrine system (100%)
Self-Replicating	2	Reproduction (50%), Parsimony (50%), Evolution (50%)
Self-Evolving	2	Predation (50%), Foraging (50%), Evolution (50%)
Robustness	2	Evolution (50%), Co-Construction (50%), Metabolism (50%)
Self-Modification	1	Reproduction
Self-Configuring	1	Artificial Immune Systems, Foraging, Predation
Self-Enhancing	1	Artificial Chemistry, Morphogenesis, Pattern Formation
Autonomy	1	Flocking, Evolution, Predation
Coordination	1	Chemical Reactions, Coordination
Interaction	1	Evolution, Swarm Intelligence
Chaos	1	Velocity History, Limit Cycle
Information Storage	1	Cellular Communication
Synchronisation	1	Imitation/Memetic Behaviour
Dynamic Criticality	1	Genetics
Power-Law Distribution	1	Genetics
Stigmergy	1	Hierarchy, Pheromones, Swarm Intelligence
Plasticity	1	Co-Construction, Metabolism

Table 6: Complex Systems Properties

The second most frequently identified property was *self-organisation*, with 28.7% of the set identifying a systematic order arising from local interactions. It was generated using 40 paradigms, most often from the biological and hybrid paradigm classes (23 (57.5%) and 13 (32.5%), respectively). The most frequently utilised individual paradigms were evolution (Leitao, 2009; Otto and Bannenberg, 2010), swarm intelligence (von Mammen and Jacob, 2009), and choice.

The third most frequently identified CAS property was *adaptation*, present in 18.26% of papers. Adaptation was generated using 29 paradigms, mostly drawn from the biological and hybrid paradigm classes (15 (51.72%) and 10 (34.5%), respectively). The most frequently utilised individual paradigms within these classes were evolution (Bornhofen and Lattaud, 2009), learning (Fernando et al., 2009), and swarm intelligence (Leitao, 2009).

Discussion

Our data extraction identified 93 distinct artificial life paradigms, which were predominantly modelled through either agent-based modelling (ABM)(55.65%) or cellular automata (CA) (18.26%). This facilitates the simulation of complex ecological or biological models using relatively simple implementation mechanics, and account for nearly 75% of the primary studies. The remaining 25% of papers used a diverse set of modelling approaches - 18 distinct approaches in all. This demonstrates that there is still a substantial area of research that uses alternative or even modelling paradigms, but that future work is necessary to ensure that their applicability is fully understood.

While 78.26% of the primary study set identified complex systems properties, 94% of this subset identified at least one of the top three most frequently occurring properties, namely, emergence, self-organisation, and adaptation. In contrast, only 32.17% of papers considered a property outside of the top three. This narrow focus indicates a gap in research where artificial life approaches and paradigms

could consider a wider range of complex adaptive system properties such as stability, autonomy, criticality, and self-regulation among others. For example, autonomy is discussed in only one primary study where the focus is on measurement and not generation (Seth, 2010). Another example is stability, which while considered in four primary studies, is often discussed without the use of any formal metrics (Punithan et al., 2011).

This limitation in existing work also translates into a limitation in evaluation, where in the majority of studies the main focus is on demonstrating that the desired emergent property occurs, without considering its potential side-effects as well as other aspects of complexity. The evaluation of existing papers is also limited through the lack of consideration for scalability, with the majority (84.35%) of the papers in the primary set not considering scalability in their analysis, thus potentially limiting the applicability of the approaches to real-life scenarios.

Complex systems are ubiquitous in a variety of domains where artificial life paradigms could be easily employed, ranging from biological or ecological modelling (Punithan et al., 2011), to project management or the design of logistical networks (Otto and Bannenberg, 2010). Our analysis identified 32 unique application domains and that 34.78% of the primary study set fell into the three most frequent application domains; *simulation* (13% of the primary study set), *social modelling* (11.3%), and *biological modelling* (10.4%). While the remaining number of application domains is fairly large, few papers were published with artificial life applications within that specific domain, with single-mention application domains representing 11.30% of the papers. This shows that artificial life paradigms are considered in a variety of application domains but that there is a need for more in-depth analysis of artificial life use within each domain to identify potential pitfalls related to their use.

Our analysis identified 93 artificial life paradigms that were employed in our primary studies, and for better discussion we grouped them into three main categories. We find that 64.52% of identified paradigms were only utilised once across the primary study set, and that 46.09% of papers utilised paradigms from one category and 44.35% of papers utilised paradigms from two categories. Similar to the discussion about complex systems properties from above, this shows that there is a clear gap in employing and validating artificial life paradigms across a variety of scenarios and application domains, and in their analysis for resulting complex systems properties. There is also a critical need for a comprehensive list of artificial life paradigms that captures the applicability of the paradigm to specific application domains in order to obtain desired properties, with sufficient modelling and implementation details to allow it to be reproduced. We consider this systematic literature review as a first step towards achieving this goal.

Conclusion

We employ a systematic literature review methodology to identify the overlap between artificial life and complex systems modeling. Our analysis identifies 93 artificial life paradigms that are used to model complex systems from a variety of application domains. Nearly 65% of the paradigms were only identified once, showing a gap in their use across a variety of domains that warrants further study. 22 complex systems properties were considered either in the design of the system, and thus facilitated by artificial life paradigms, or in the evaluation of the system, and thus potentially caused by the use of a specific artificial life paradigm. Of the papers that discussed complex systems properties, only around 30% considered properties outside the most popular set comprised of emergence, self-organisation, and adaptation, showing a need for deeper application of complex systems theory. Lastly, a significant gap has been identified with respect to scalability analysis, with only 16% of the papers considering scalability (either in terms of the number of entities or their complex behaviors and interactions) in their evaluation or design. Overall, our analysis identifies that while there is a broad application of artificial life and complex systems theories across a variety of models and domains, there is a need for more in-depth study of artificial life paradigms and complex systems properties in order to fully exploit their benefits.

References

- Azumagakito, T., Suzuki, R., and Arita, T. (2011). Visualizing language evolution as an emergent phenomenon based on biological evolution and learning. *Artificial Life and Robotics*, 16(3):366–372.
- Banerjee, M., Capozzoli, M., McSweeney, L., and Sinha, D. (1999). Beyond kappa: a review of interrater agreement measures. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 27(1):3–23.
- Bedau, M. A. (2003). Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive sciences*, 7(11):505–512.
- Birdsey, L., Szabo, C., and Teo, Y. M. (2015). Twitter knows: Understanding the emergence of topics in social networks. In *Proceedings of the Winter Simulation Conference*, pages 4009–4020.
- Bornhofen, S. and Lattaud, C. (2009). Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing*, 8(2):349–385.
- Bousquet, F. and Le Page, C. (2004). Multi-agent simulations and ecosystem management: a review. *Ecological modelling*, 176(3–4):313–332.
- Cocco, L., Mannaro, K., Concas, G., and Marchesi, M. (2014). Simulation of the Best Ranking Algorithms for an App Store. In *Mobile Web Information Systems*, volume 8640, pages 233–247.

- Davies, M. and Fleiss, J. L. (1982). Measuring agreement for multinomial data. *Biometrics*, 38(4):1047–1051.
- de Buitelir, A., Russell, M., and Daly, M. (2012). Wains: A Pattern-seeking Artificial Life Species. *Artif. Life*, 18(4):399–423.
- Emmeche, C. (1998). Defining life as a semiotic phenomenon. *Cybernetics & Human Knowing*, 5(1):3–17.
- Fernando, S., Matsuzaki, S., and Marasinghe, A. (2009). Modeling towards homeostatic plasticity in neuronal activities. *Artificial Life and Robotics*, 14(2):262–265.
- Froese, T., Virgo, N., and Izquierdo, E. (2007). Autonomy: a review and a reappraisal. In *European Conference on Artificial Life*, pages 455–464. Springer.
- Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, 19(1):1–8.
- Isidoro, C., Fachada, N., Barata, F., and Rosa, A. (2011). Agent-Based Model of Dengue Disease Transmission by *Aedes aegypti* Populations. In *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5777, pages 345–352.
- Janecek, A., Jordan, T., and de Lima-Neto, F. B. (2013). Agent-Based Social Simulation and PSO. In *Advances in Swarm Intelligence*, volume 7929, pages 63–73.
- Joachimczak, M., Kowaliw, T., Doursat, R., and Wrbel, B. (2013). Evolutionary design of soft-bodied animats with decentralized control. *Artificial Life and Robotics*, 18(3-4):152–160.
- Kirshenbaum, M., Palmer, D., McCullick, P., and Vaidyanathan, R. (2008). Explaining Swarm Design Concepts using an Interactive, vBottom-up Simulation Tool. In *Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education*, pages 298–303.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.
- Krol, D. and Popiela, . (2009). Modelling Shortest Path Search Techniques by Colonies of Cooperating Agents. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, volume 5796, pages 665–675.
- Langton, C. G. et al. (1989). Alife i. In *Proceedings of the first international workshop of the synthesis and simulation of living systems*.
- Leitao, P. (2009). Holonic Rationale and Self-organization on Design of Complex Evolvable Systems. In *Holonic and Multi-Agent Systems for Manufacturing*, volume 5696, pages 1–12.
- Lopez, O. J. R. (2010). Self-organized and Evolvable Cognitive Architecture for Intelligent Agents and Multi-Agent Systems. In *Applications of Evolutionary Computation*, volume 6024, pages 392–401.
- Loula, A., Gudwin, R., Ribeiro, S., and Queiroz, J. (2010). On Building Meaning: A Biologically-Inspired Experiment on Symbol-Based Communication. In *Brain Inspired Cognitive Systems 2008*, volume 657, pages 77–93.
- Mittal, S. (2013). Emergence in stigmergic and complex adaptive systems: A formal discrete event systems perspective. *Cognitive Systems Research*, 21:22–39.
- Monteagudo, . and Santos Reyes, J. (2013). Cancer Stem Cell Modeling Using a Cellular Automaton. In *Natural and Artificial Computation in Engineering and Medical Applications*, volume 7931, pages 21–31.
- Niazi, M. A. (2014). Emergence of a Snake-Like Structure in Mobile Distributed Agents: An Exploratory Agent-Based Modeling Approach. *SCIENTIFIC WORLD JOURNAL*.
- North, M. J., Collier, N. T., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M., and Sydelko, P. (2013). Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):3.
- Oswald, Y. and Schmickl, T. (2017). Ultimate Ecology: How a socio-economic game can evolve into a resilient ecosystem of agents. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8.
- Otto, S. and Bannenberg, T. (2010). Decentralized Evolutionary Agents Streamlining Logistic Network Design. In *Parallel Problem Solving from Nature, PPSN XI*, pages 240–249.
- Punithan, D., Kim, D.-K., and McKay, R. (2011). Daisyworld in Two Dimensional Small-World Networks. In *Database Theory and Application, Bio-Science and Bio-Technology*, volume 258, pages 167–178.
- Santé, I., García, A. M., Miranda, D., and Crecente, R. (2010). Cellular automata models for the simulation of real-world urban processes: A review and analysis. *Landscape and Urban Planning*, 96(2):108–122.
- Seth, A. K. (2010). Measuring Autonomy and Emergence via Granger Causality. *Artif. Life*, 16(2):179–196.
- Shibuya, N., Iizuka, H., and Yamamoto, M. (2018). Evolution of Communication Through Differentiation of Communicative and Goal-directed Behaviors. *Artif. Life Robot.*, 23(2):225–234.
- Szabo, C., Teo, Y. M., and Chengleput, G. K. (2014). Understanding complex systems: Using interaction as a measure of emergence. In *Proceedings of the Winter Simulation Conference*, pages 207–218.
- Taylor, C. E. and Cody, M. L. (2015). Bird song: a model complex adaptive system. *Artificial Life and Robotics*, 20(4):285–290.
- Varela, D. and Santos, J. (2018). Automatically obtaining a cellular automaton scheme for modeling protein folding using the FCC model. *Natural Computing*.
- von Mammen, S. and Jacob, C. (2009). Swarming for Games: Immersion in Complex Systems. In *Applications of Evolutionary Computing*, volume 5484, pages 293–302.
- Yamamoto, L. and Miorandi, D. (2010). Evaluating the Robustness of Activator-Inhibitor Models for Cluster Head Computation. In *Swarm Intelligence*, volume 6234, pages 143–154.
- Yu, Y.-M. and Zhao, G.-Q. (2010). Virtual battlefield and combat simulation based on Artificial Life theory. In *Proceedings 6th International Conference on Natural Computation*, volume 6, pages 2820–2825.

Inferring a Graph's Topology from Games Played on It

Douglas G. Moore and Sara I. Walker

BEYOND Center, Arizona State University, Tempe, AZ 85281

douglas.g.moore@asu.edu

sara.i.walker@asu.edu

Abstract

We consider an iterated model of agents playing a two-player game on a graph. The agents change their strategies as the game progresses based on anticipated payoffs. Using only the time series of the agents' strategies, we determine the pairwise mutual information between all agents in the graph, and use these values as predictors of the graph's topology. From this, we assess the influence of various model parameters on the effectiveness of mutual information at recovering the actual causal structure. It is found that the degree to which the functional connectivity reflects the actual causal structure of the graph strongly depends on which game is being played and how the agents are changing their strategies. Further, there is evidence that the edge density of the graph may also have some impact on the accuracy of the inferred network. This approach allows us to better connect the dynamics of the systems under study with the difference in their functional and actual connectivity, and has broad implications for the interpretation and application of information-based network inference. The methods and analyses described can be generalized and applied to other types of network models.

Introduction

There is a wealth of literature on using information-theoretic tools to infer the network topology of collections of interacting random variables, particularly in neuroscience (Ito et al., 2011; Vicente et al., 2011; Lizier and Rubinov, 2012; Ver Steeg and Galstyan, 2012; Banerji et al., 2013; Sun and Boltt, 2014; Sun et al., 2015). In many of these works, the phrase "functional connectivity" is used to underscore the fact that the actual causal interactions are not always readily inferred from the dynamics of the variables. This is often considered yet another example of the inequivalence of correlation and causation. While it appears that functional connectivity is a useful construct in neuroscience, that may not be the case in other fields. However, it is interesting to explore the conditions under which the actual and functional connectivity may coincide.

We consider a somewhat unorthodox model system to explore whether and when information-theoretic tools recapitulate the actual topology of a causal graph. Agents are situated on the vertices of a graph, and allowed to repeatedly

play a two-player, two-strategy, symmetric game. In each round of the game, the agents may change their strategies based on their anticipated payoff. Treating the strategies of the agents as random variables in time, when can we confidently infer the actual connectivity of the graph? This model system was inspired by conversations with colleagues who explored the dynamics of games when agents' perception of which game they're playing is allowed to change (Antonioni et al., 2018). Such a system is intrinsically interesting to consider, but it has value beyond that; namely in potential applications. Biologists have modeled interacting cells (Hummert et al., 2014), organisms (Smith John Maynard et al., 1979; Smith, 1982; Riechert and Hammerstein, 1983; Nowak and Sigmund, 2004) and even cancer (Bach et al., 2001; Gatenby and Vincent, 2003; Basanta et al., 2008; Dingli et al., 2009) as game-playing agents, and game theory has of course found innumerable applications in economics. In both cases, the strategies of the agents may be readily apparent, while the network itself is obscured. The ability to infer functional or actual connectivity based solely on the strategies could lead to more impactful methods of intervening on these systems.

From a technical perspective, this model offers a number of useful features: the graph's topology can be varied, the rules the agents use to update their strategies can be changed, and the space of graphs is nicely parameterized by two bounded variables. Altogether, this provides a number of ways to tweak the system and observe changes in the efficacy of network inference. Further, because this model is a particular type of Boolean network, it seems reasonable that the results and methods herein could be extrapolated and applied to more generic settings.

In this work, we describe the basic formalism, and present early results from simulations¹. We observe that the efficacy of the network reconstruction is dependent on the game the agents are playing, the rule they are using to update their strategies, and possibly the density of edges in the graph. We conclude with a number of possible future directions.

¹All code used for these simulations is publicly available at <https://github.com/dgmoore/gomen>.

Methods

Consider a finite collection of agents $\{A_1, \dots, A_n\}$. Each agent A_i is situated at a vertex v_i of some *simple*² graph $G = (V, E)$ with vertices $V = \{1, \dots, n\}$ and edges $E \subset V \times V$. The agents then play some, two-player, two-strategy, symmetric game with each of their neighbors. We restrict the games to those characterized by a payoff matrix of the form

$$\mathcal{G} = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & S \\ 1 & T & 0 \end{array} \quad (1)$$

with $-1 \leq S \leq 1$ and $0 \leq T \leq 2$. The elements of this matrix \mathcal{G}_{ab} specify the payoff received by the agent should they play strategy a against an opponent playing strategy b . It is assumed throughout that all players are knowingly playing the same game, and that the game does not change. The strategy played by A_i is denoted s_i and can be either 0 or 1, sometimes referred to as *cooperate* or *defect*, respectively. The collection of strategies played by all of the agents is denoted $\mathbf{s} = (s_1, \dots, s_n)$.

The game is played in rounds starting with each agent's strategy selected uniformly at random. In each round, the agents play with their neighbors $N_i = \{j \mid (i, j) \in E\}$ and accumulate payoff

$$P_i(s_1, \dots, s_n) = \sum_{j \in N_i} \mathcal{G}_{s_i s_j} \quad (2)$$

We write this as $P_i(\mathbf{s})$ for compactness. Since the agents know which game they are playing, they may also consider the *counterfactual* payoff they would have received had they played their alternative strategy \hat{s}_i :

$$\bar{P}_i(\mathbf{s}) = P_i(s_1, \dots, \hat{s}_i, \dots, s_n) = \sum_{j \in N_i} \mathcal{G}_{\hat{s}_i s_j}. \quad (3)$$

Based on the difference between counterfactual payoffs $\Delta P_i(\mathbf{s}) = \bar{P}_i(\mathbf{s}) - P_i(\mathbf{s})$, each agent may stochastically change its strategy for the next round with probability

$$q_i(\beta; \mathbf{s}) = \frac{1}{1 + \exp(-\beta \Delta P_i(\mathbf{s}))}, \quad \beta > 0. \quad (4)$$

In words, agent A_i will use strategy \hat{s}_i in the next round with probability $q_i(\beta; \mathbf{s})$. All agents then synchronously update their strategies, and a new round is ready to begin. This process is iterated t times, and a time series of strategies is generated $\{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_t\}$.

The outstanding question is to what degree this time series can be used to infer the topology of the underlying graph. A number of information-theoretic tools have been proposed for assessing connectivity in similar situations. The most popular, particularly among neuroscientists, is transfer entropy or some variation thereof (Ito et al., 2011; Vicente

et al., 2011; Orlandi et al., 2014). However, in this initial work we decided to consider a simpler measure. We start with

$$I(s_i^+, s_j) = \sum_{s_i^+, s_j} p(s_i^+, s_j) \log \frac{p(s_i^+, s_j)}{p(s_i^+)p(s_j)}, \quad (5)$$

the mutual information between agent A_i 's next strategy s_i^+ and agent A_j 's current strategy s_j . Since we know that edges of the graph are undirected, we construct a measure that is symmetric in the agents:

$$\phi_{ij} = \frac{1}{2}(I(s_i^+, s_j) + I(s_j^+, s_i)). \quad (6)$$

This value is computed for every pair of distinct agents, $i \neq j$ for $i, j \in V$, and used as a predictor for the existence of an edge in the graph. If $\phi_{ij} \geq \theta$ for some threshold $0 \leq \theta \leq 1$, we predict that the edge (i, j) exists, and that it otherwise does not. In a binary classification process such as this, one method of characterizing the effectiveness of the classification is in terms of the true-positive rate (TPR) and false-positive rate (FPR)

$$TPR = \frac{TP}{R} \quad \text{and} \quad FPR = \frac{FP}{F}. \quad (7)$$

Here the TPR is the ratio of the number of edges *correctly* predicted to exist (TP) and the actual number of existent edges (R), while the FPR is the ratio of the number of edges *incorrectly* predicted to exist (FP) and the number of possible edges which *do not* exist (F). The value of the threshold can be varied continuously over its range $\theta \in [0, 1]$, yielding a receiver operating characteristic (ROC) curve in the TPR - FPR space (fig. 2). The area under the curve (AUC) acts as a summary statistic for the quality of the classification. A predictor which is no better than uniformly random will yield a perfect diagonal with $AUC = 0.5$. Better predictors will yield curves which deviate from the diagonal with $|AUC - 0.5| \gg 0$. We consider predictors with $AUC \ll 0.5$ to be of high-quality because the ROC curve reflects over the diagonal if we change the classification criterion from $\phi_{i,j} \geq \theta$ to $\phi_{i,j} \leq \theta$. In other words, the predictor was good, but we were using the wrong criterion. It is important to note that this analysis does not tell us what the threshold θ should be, though in principle it can be determined. For now, it is sufficient to know only that such a threshold exists.

Results

We were interested in the relationship between the form of the game \mathcal{G} , the topology of the graph G , and the rule parameter β . To explore this, we considered 50 agents playing on graphs with one of 5 types of topology: a.) Barabási-Albert graphs, b.) Erdős-Rényi graphs, and the c.) cycle, d.) wheel and e.) lattice topologies. The Barabási-Albert model (Barabási and Albert, 1999) generates random graphs using preferential attachment; as new

²Simple here means an undirected graph with no self-loops and at most one edge between any two given vertices.

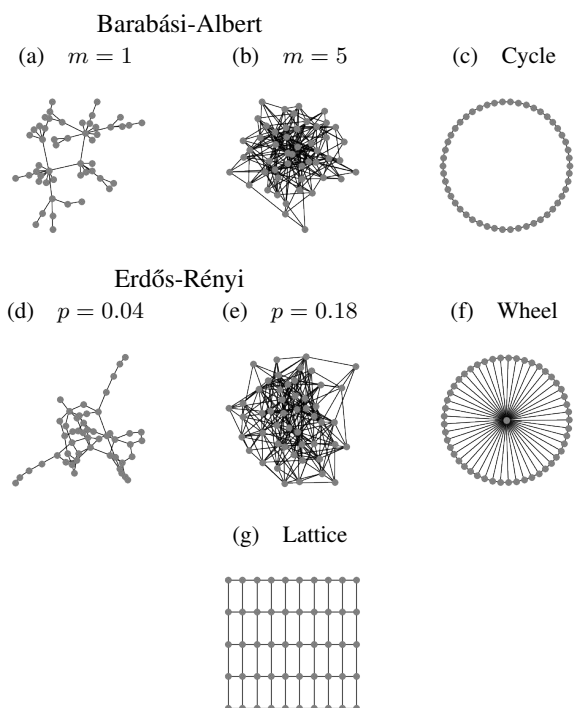


Figure 1: Samples of the topologies considered. The Barabási-Albert (a-b) and Erdős-Rényi graphs (d-e) are generated non-deterministically according to edge density parameters m and p respectively. The cycle (c), wheel (f) and 5×10 lattice (g) topologies have only one representative each.

nodes are iteratively added to the graph they are randomly connected to m existent nodes with a bias toward those with higher degree; we considered only $m \in \{1, 5\}$ in this work. The Erdős-Rényi model (Erdős and Rényi, 1959) starts with the full set of nodes and adds an edge between each pair of distinct nodes with probability p . We chose $p \in \{0.04, 0.18\}$ to ensure that the edge density of the Barabási-Albert and Erdős-Rényi graphs are comparable. The cycle, wheel and 5×10 non-periodic lattice topologies, on the other hand, are non-random topologies, and are depicted in 1.

Once the graphs were constructed, we considered all games of the form eq. (1) with $S \in \{-1.0, -0.9, \dots, 1.0\}$ and $T \in \{0.0, 0.1, \dots, 2.0\}$. We then generated 100 graphs according to each of the parameters above for each combination of game and topology. For the cycle, wheel and lattice topologies, there is only one graph (up to isomorphism) for a given number of vertices, so we reran the simulations on the same topology 100 times. We simulated 10 rounds of play starting from 100 initial conditions selected uniformly at random. The initial conditions were generated independently for each simulation, i.e. each combination of game, graph and rule parameter. The time series of strategies in each simulation of 10 rounds with 100 replicates was used to generate an ROC curve. The 100 ROC curves for each game

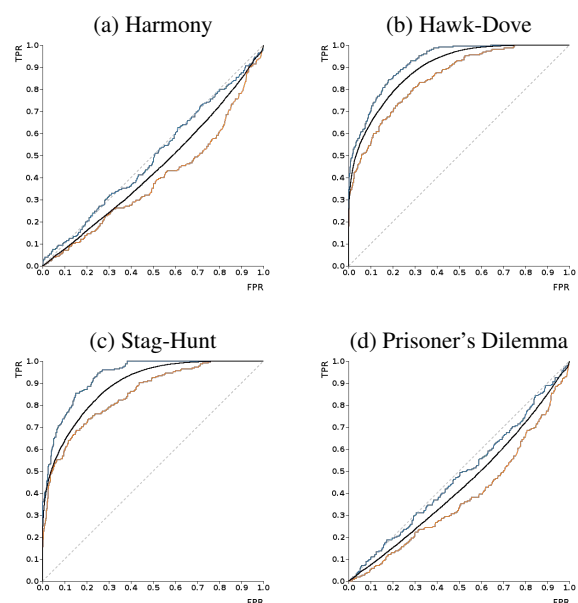


Figure 2: Receiver operating characteristic curves for four canonical games with (T, S) parameterizations of (a) $(1/2, 1/2)$, (b) $(3/2, 1/2)$, (c) $(1/2, -1/2)$ and (d) $(3/2, -1/2)$ as played by 50 agents on Barabási-Albert graphs with $m = 5$. In all cases, the agents employ a threshold function with rule parameter $\beta = 1.0$. Each plot summarizes the results of simulations run on 100 randomly generated graphs. The black curve is the point-wise mean of the 100 resulting ROC curves, while the blue and orange curves are those with the greatest and least AUC, respectively. The dashed gray line represents the curve for a predictor which is no better than chance.

and topology combination (one for each randomly generated graph) were point-wise averaged to yield a mean ROC curve. Finally, the AUC was computed for each mean curve, and a scaled deviation of that value from the worst-case value of 0.5:

$$\overline{AUC} = 2|AUC - 0.5| \quad (8)$$

was plotted as a heatmap on an S - T plot (fig. 3, fig. 4 and fig. 5). Here S and T refer to the parameters which specify a game as in eq. (1). In other words, each point in the plot corresponds to a different game with the color roughly characterizing the effectiveness of ϕ_{ij} as a predictor for the existence of the edge (i, j) in the graph.

Discussion

There are a number of notable features apparent in fig. 3. The plots are roughly symmetric across their diagonals so that opposite quadrants have similar characteristics. The nice thing about these S - T parameterization is that each of the quadrants corresponds to a common two-player, two-strategy, symmetric game. Counter-clockwise from the top-right quadrant, the games are *Hawk-Dove*, *Harmony*, *Stag-Hunt* and *Prisoner's Dilemma*. The interesting point

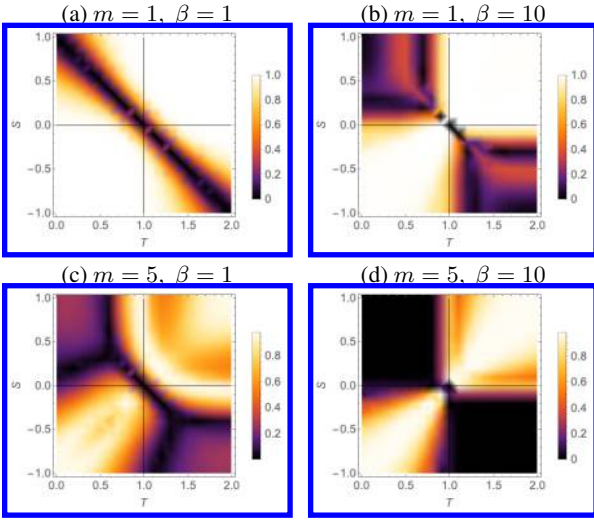


Figure 3: \overline{AUC} heatmaps for games played on Barabási-Albert random graphs. Each row corresponds to a different connection parameter $m \in \{1, 5\}$, while the columns refer to the rule parameter used by the agents $\beta \in \{1, 10\}$. Each point in the plot corresponds to a different game parameterized as in eq. (1) by S and T . For each game, 100 random graphs were selected and an average ROC curve was generated. The color represents how much the AUC under the mean curve differs from the uniformly random predictor as in eq. (8).

to note is that the games opposite one another have the same number of pure-strategy Nash equilibria: Harmony and Prisoner’s Dilemma each have one while Hawk-Dove and Stag-Hunt have two. On its face, this may not appear to be relevant to the problem of inferring the topology of the graph. However, the Nash equilibria play a crucial role given how the agents are updating their strategies: over time, the agents tend to settle on playing an equilibrium strategy. This process happens fairly quickly, typically within the 10 rounds of play simulated in this study. In effect, the simulations of the Harmony and Prisoner’s Dilemma reach equilibrium and remain there, though with some amount of noise induced by the agents’ strategy update rule. However, the transition to equilibrium takes longer for Hawk-Dove and Stag-Hunt because they have two equilibria. The resulting time series then provides more information about the interactions between agents. One additional factor to consider here is that in the two-equilibrium games, it is possible for domains to form wherein agents in a given domain all play the same equilibrium strategy while agents in neighboring domains play a different strategy. The stability of these domains and the impact they have on the graph inference problem is something to be investigated in future work.

One apparent problem with the above interpretation is it suggests that each of the four quadrants of the heatmap should be filled with the same \overline{AUC} value, e.g. that every harmony game should have the same $\overline{AUC} \sim 0$. Of course, that is clearly not the case (fig. 3(a)). Notice though that the

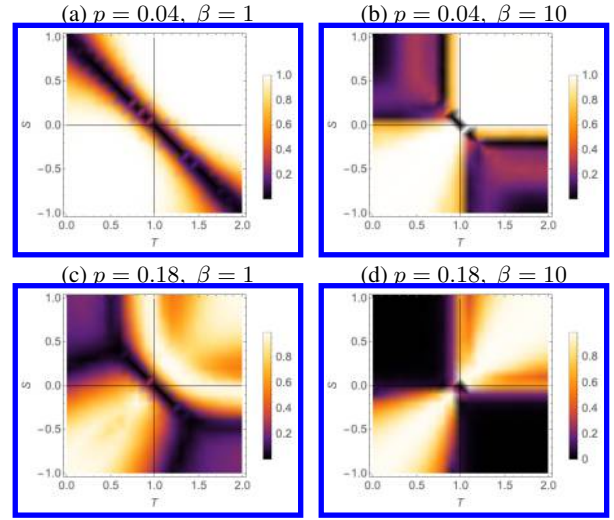


Figure 4: \overline{AUC} heatmaps for games played on Erdős-Rényi random graphs. Each row corresponds to a different connection probability $p \in \{0.04, 0.18\}$, while the columns refer to the rule parameter used by the agents $\beta \in \{1, 10\}$.

classification improves as the game moves away from the negatively-sloped diagonal within the Harmony and Prisoner’s Dilemma quadrants. This effect is due to the agents using a stochastic update rule which is dependent on the difference in payoffs of two strategies. As the game moves further from the negatively-sloped diagonal, the payoff differences make differentiating between the strategies more difficult in games where there is only one equilibrium. As a result, the agents spend more time in transition to equilibrium and consequently provide more information about the topology of the graph. If this analysis is correct, we should expect to see that increasing the rule parameter β allows the agents to differentiate smaller payoff differences, and the quadrants of these heatmaps should tend to become more homogeneous. This is exactly what we see, particularly when edge density of the graphs is low (fig. 3, fig. 4 and fig. 5).

This brings us to the final factor involved in determining the efficacy of the classification: the edge density of the graph. When we compare, for example, the Barabási-Albert graphs with $m = 1$ (fig. 3(a-b)) to the Erdős-Rényi graphs with $p = 0.04$ (fig. 4(a-b)), we see almost identical structure in the \overline{AUC} heatmaps. On average the Erdős-Rényi graphs have the same edge density to their Barabási-Albert counterparts, i.e. they have about the same number of edges. The cycle graph (fig. 5(a-b)) has the same edge density to the Barabási-Albert graph ($m = 1$), and again we see almost identical heatmaps. By increasing the edge parameters for the Barabási-Albert and Erdős-Rényi graphs, m and p respectively, we can see a similar comparison for yet denser graphs (fig. 3(c-d) and fig. 4(c-d)). Again, the Barabási-Albert ($m = 5$) and

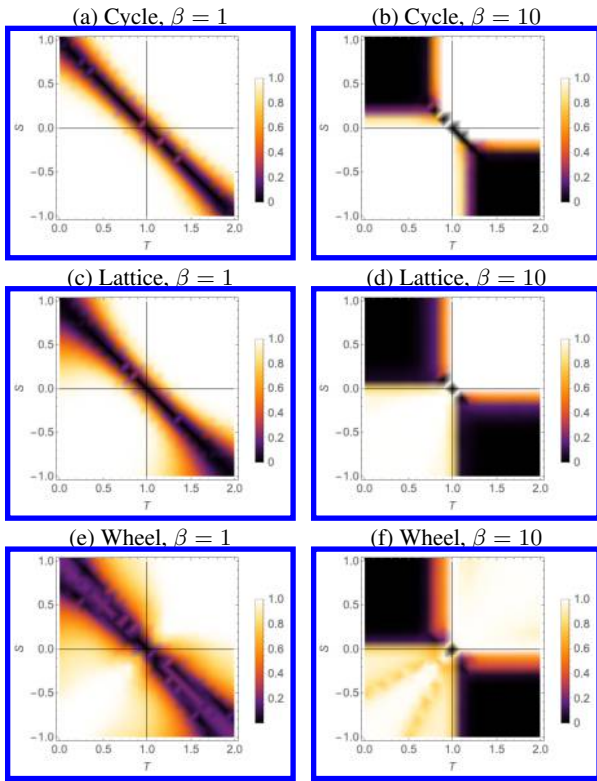


Figure 5: \overline{AUC} heatmaps for games played on the cycle, wheel and lattice topologies. Each row corresponds to a different topology while the columns refer to the rule parameter used by the agents $\beta \in \{1, 10\}$. The edge density of the topologies increases down the column, i.e. the cycle has fewer edges than the lattice which has fewer edges than the wheel.

Erdős-Rényi ($p = 0.18$) graphs have similar edge density. What's more, the same qualitative change is seen for a single type of topology when the edge density changes. For example, comparing fig. 3(a) to 3(c) or fig. 4(a) to 4(c), we see fewer harmony and prisoner's dilemma games admit a faithful inference. The same phenomenon is observed in comparing cycles, lattices and wheels (fig. 5), which further supports the claim. The graphs have remarkably distinct topologies even with only 50-nodes. For example, it is difficult to find any similarity between the Barabási-Albert and cycle graphs other than edge density. However, it is entirely plausible that some other topological similarity between the graphs is causing these effects. Increasing the size of the graph should lead to increasingly divergent topological features. That would allow us to test this claim more fully.

Conclusion

From all of these results, however incomplete, we can begin to draw some insight. It seems profoundly important that in order to be confident that the reconstructed topology – that is the inferred causal structure of the system – is the actual topology, an outside observer must be privy to

the inner workings of the agents. What game are they playing, and how are they deciding which strategy to use next? Of course, the simple retort to this is that reconstructions are failing for the one-equilibrium games because the time series are not diverse enough. If the observer could assess this in advance, it might temper or bolster their confidence. However, it cannot be understated that the rule parameter plays some role in the diversity of the time series. If the rule is too stochastic, the time series will be riddled with *biased* noise. One possibility is to use similar information-theoretic tools to infer which game is being played. It's been shown in previous work that dynamical agents operating under different rules show distinct patterns of information transfer (Valentini et al., 2018). If the observer could first apply such a method to determine the game³, rule, and noise profile, they may be able to yield more faithful reconstructions of the topology.

From here there are a number of avenues for further research. The mutual information is a somewhat simplistic measure for applications like this. We chose it in part because of this simplicity. Two alternative measures, for example, might be conditional mutual information (CMI) or transfer entropy (TE), (Schreiber, 2000). In preliminary work, we briefly explored both. The conditional mutual information approach was problematic in that it required more substantial computational overhead than we had available at the time. One method of using CMI is to compute $I(s_i^+, s_j | s_{ij})$ where $s_{ij} = \{s_k | k \neq i, k \neq j\}$, that is to say condition the pair-wise mutual information on the strategies of all other agents. Since our graphs have 50 nodes, that computation was infeasible using direct approaches. Alternative methods of computing CMI, such as the iterative approaches of (Lizier and Rubinov, 2012; Sun et al., 2015), may make it viable for future work. Transfer entropy is a special case of CMI; it would condition on the target agent's previous strategies. Both approaches are worth exploring more rigorously.

In this work, we were interested in methods that required only one only observe a system and not intervene on it, as in many cases only observation is accessible to the researcher. However, it is well understood that observation alone, particularly in the case of large systems wherein only a small portion of the state space can be explored, is inadequate for extracting causal relationships between components of a system. Interventional methods, such as (Ay and Polani, 2008), are much more effective in such cases, provided you have the ability to control aspects of the system. Subsequent work may explore such an approach.

The dynamics of the models we considered here are similar in many respects to the classic Ising model, with β acting much like temperature and the payoff matrix eq. (1) behaving similarly to interaction strengths. It is reasonable to ex-

³It may not be necessary to know precisely which game is being played. Perhaps knowing the number of equilibria is sufficient?

pect that analogues of heat capacity, entropy, and other thermodynamic measures could be related in some way to the effectiveness of the network inference. Comparing the AUC with some of these thermodynamic analogues is a possibility for a future work.

Another advance could be to look at other methods of quantifying the efficacy of the edge classifications. As the reader may have noticed, we never considered complete graphs. The reason for this is that to construct the ROC curves, you have to compute the false-positive rate. This value is undefined if your dataset has no negative cases. In our application, that means there must be pairs of agents that are not connected.

Two of the most egregious assumptions we've made in this work are that every agent is using the same rule to update its strategy, and every pair of agents is playing the same game. There is little reason to expect that real-world systems behave in this manner, and either assumption can be relaxed. For example, we could associate a different game with each edge. We have no expectation for how this would affect the results, but will explore it in the future.

Finally, as with virtually every simulation study, there is room for more fine-grained parameter sweeps. Our sampling of the S - T space of games was rather coarse; we limited ourselves to an unreasonably small number of initial conditions (100 of the 2^{50} possible) for each graph, and we considered only a handful of graph topologies. All of this will be greatly expanded in future work.

Ultimately, we see that the difference in functional and actual connectivity can sometimes be tied directly to the dynamical structure of the system. It may be impossible to know how similar the functional and actual connectivity of these types of networks are without detailed knowledge of the dynamics of the agents in the system. This may not be a surprise to some, but it is important that researchers to acknowledge it explicitly. Without that, it can be all too easy to blindly apply these network inference methods and over interpret the results, a sin of which at least the first author is guilty.

Acknowledgements

D.G.M. and S.I.W. would like to acknowledge the Lifelong Learning Machines program from DARPA/MTO for funding this work. D.G.M also acknowledges Cole Mathis for the conversations about game theory which inspired this work.

References

- Antonioni, A., Martinez-Vaquero, L. A., Mathis, C., Peel, L., and Stella, M. (2018). Individual perception dynamics in drunk games. *arXiv:1807.08635*.
- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11(01):17–41.
- Bach, L. A., Bentzen, S. M., Alsner, J., and Christiansen, F. B. (2001). An evolutionary-game model of tumour–cell interactions: possible relevance to gene therapy. *European journal of cancer*, 37(16):2116–2120.
- Banerji, C. R. S., Severini, S., and Teschendorff, A. E. (2013). Network transfer entropy and metric space for causality inference. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 87(5):052814.
- Barabási, A. L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.
- Basanta, D., Simon, M., Hatzikirou, H., and Deutsch, A. (2008). Evolutionary game theory elucidates the role of glycolysis in glioma progression and invasion. *Cell proliferation*, 41(6):980–987.
- Dingli, D., Chalub, F. A. C. C., Santos, F. C., Van Segbroeck, S., and Pacheco, J. M. (2009). Cancer phenotype as the outcome of an evolutionary game between normal and malignant cells. *British journal of cancer*, 101(7):1130–1136.
- Erdős, P. and Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297.
- Gatenby, R. A. and Vincent, T. L. (2003). Application of quantitative models from population biology and evolutionary game theory to tumor therapeutic strategies. *Molecular cancer therapeutics*, 2(9):919–927.
- Hummert, S., Bohl, K., Basanta, D., Deutsch, A., Werner, S., Theissen, G., Schroeter, A., and Schuster, S. (2014). Evolutionary game theory: cells as players. *Molecular bioSystems*, 10(12):3044–3065.
- Ito, S., Hansen, M. E., Heiland, R., Lumsdaine, A., Litke, A. M., and Beggs, J. M. (2011). Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model. *PloS one*, 6(11):e27431.
- Lizier, J. T. and Rubinov, M. (2012). Multivariate construction of effective computational networks from observational data. Preprint, Max Planck Institute for Mathematics in the Sciences.
- Nowak, M. A. and Sigmund, K. (2004). Evolutionary dynamics of biological games. *Science*, 303(5659):793–799.
- Orlandi, J. G., Stetter, O., Soriano, J., Geisel, T., and Battaglia, D. (2014). Transfer entropy reconstruction and labeling of neuronal connections from simulated calcium imaging. *PloS one*, 9(6):e98842.
- Riechert, S. E. and Hammerstein, P. (1983). Game theory in the ecological context. *Annual Review of Ecology and Systematics*, 14(1):377–409.
- Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, 85(2):461–464.
- Smith, J. M. (1982). *Evolution and the Theory of Games*. Cambridge University Press.
- Smith John Maynard, Maynard Smith J., and Holliday Robin (1979). Game theory and the evolution of behaviour. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 205(1161):475–488.

- Sun, J. and Bollt, E. M. (2014). Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings. *Physica D. Nonlinear phenomena*, 267:49–57.
- Sun, J., Taylor, D., and Bollt, E. (2015). Causal network inference by optimal causation entropy. *SIAM journal on applied dynamical systems*, 14(1):73–106.
- Valentini, G., Moore, D. G., Hanson, J. R., Pavlic, T. P., Pratt, S. C., and Walker, S. I. (2018). Transfer of information in collective decisions by artificial agents. *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 641–648.
- Ver Steeg, G. and Galstyan, A. (2012). Information transfer in social media. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, pages 509–518, New York, NY, USA. ACM.
- Vicente, R., Wibral, M., Lindner, M., and Pipa, G. (2011). Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67.

Inferring Swarm Models Using a Single Monitoring Robot

Suppanat Ruangdech*², Sabine Hauert^{1,2} and Martin Homer¹

¹Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom

²Bristol Robotics Laboratory, Bristol, United Kingdom

*Correspondence:sr15417@bristol.ac.uk

Abstract

Infiltrating a swarm of artificial or living agents using a single monitoring robot could allow for the assessment of their swarm rules and parameters without the need for any external infrastructure. The inferred swarm model could then be used to control these swarms, for example to guide them to safe areas. In this study we introduce a scheme for autonomous artificial agents to extract knowledge about the interactions within a swarm of interest. By infiltrating the swarm of interest with a monitoring robot and constantly measuring the distance between the infiltrator and its nearest neighbour, the repulsion radius of the swarm agents can be estimated. Though this method works for a range of tested parameters, it is still limited to a specific model of interaction.

Introduction

Understanding the rules that give rise to an observed emergent behaviour typically takes many iterations, either through trial and error or using automatic tools such as machine learning (Hauert et al., 2014). Most important, such studies typically require an external telemetry system that can track the position of all agents in the swarm over time (Puckett et al., 2014; Li et al., 2016). In contrast, our aim is to send a single monitoring robot into a swarm to extract its rules and parameters. As a first step we show in simulation how a robot injected into a swarm moving according to a flocking model based on Couzin et al. (2002) can extract the repulsion radius of the swarm using only local observations. Flocking was chosen as it is a well studied swarm algorithm (Fine and Shell, 2013). In the future, we aim to use the information learned by the robot, to control the swarm, for example by moving it in space.

Methodology

In order to simplify the problem, the following assumptions are made. First, the general structure of the swarm of interest's interaction rules is known. Second, the unknown part of the swarm of interest's interaction rules is the actual parameter of the rules. Thus, the task of the monitoring agent is reduced to inferring the unknown parameter's value.

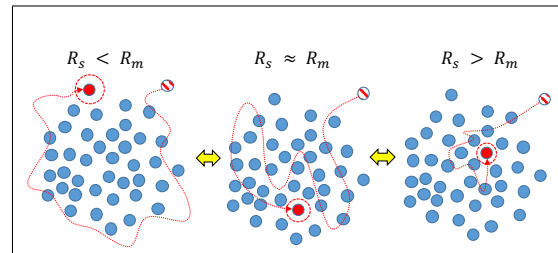


Figure 1: The different areas which the monitoring agent (red) frequents according to the relative size of R_s and R_m .

This study is conducted through custom-made computer simulations. In general, a swarm of 50 identical agents are initiated in a 2D-simulated space. Then, a monitoring agent is deployed into the same space to interact with the swarm and collect data. Each simulation is terminated after a fixed time of 500 simulation seconds.

The behaviours of each agent, including the monitoring agent, are governed by a modified version of the interaction rules from Couzin et al. (2002); we neglect the zone of orientation from that theory. The swarm agents are set to react to the monitoring agent the same way they react to their conspecifics. The repulsion radius of the swarm agents, R_s , which is selected as the unknown parameter, is fixed for a set of simulations, while the monitoring agent's repulsion radius, R_m , is varied.

Results

Observation of simulated flocks suggests found that the monitoring agent travels in different area w.r.t. the swarm as R_m varies (see Figure 1). When $R_s < R_m$, the monitoring agent tends to stay outside of the swarm, while, when $R_s > R_m$, the monitoring agent tends to stay in the center of the swarm. When $R_s \approx R_m$, the monitoring agent's position is more ambiguous compared with the previous two extreme cases. Figure 2 shows the histogram of the distance between the monitoring agent and the centroid of the swarm, which agrees with the qualitative description above. Thus,

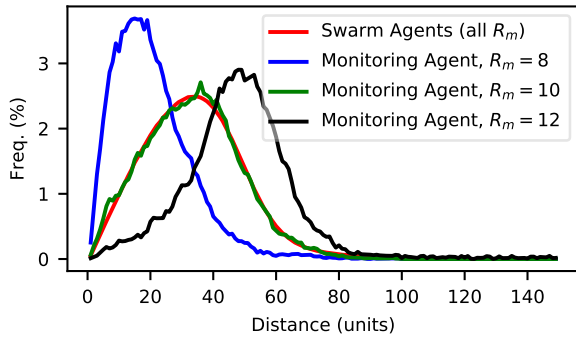


Figure 2: Distributions of the distances from the centroid of the swarm to the monitoring and swarm agent when $R_s = 10$ and R_m is varied.

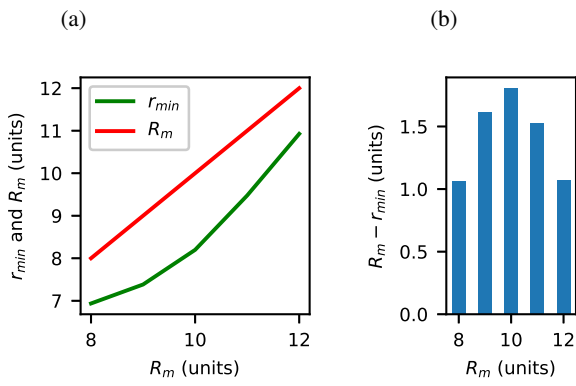


Figure 3: (a) Plots comparing r_{min} and R_m as a function of R_m (b) $R_m - r_{min}$ as a function of R_m , which is maximised when $R_m = R_s$. ($R_s = 10$ in both panels.)

it is possible to estimate R_s by varying R_m and monitoring the area that the monitoring agent frequents. However, this method requires a global view of the system in order to measure the distance between the monitoring agent and the swarm's centroid. Therefore, we instead investigate local measurements—the measurement which the monitoring agent is able to perform.

One of the monitoring agent's local measurements that is affected by the value of R_m is the average distance from the monitoring agent to the nearest swarm agent, r_{min} . Figure 3(a) shows that r_{min} and R_m increase at a different rate. When R_m is lower than R_s , the increment rate of r_{min} is lower than that of R_m . In contrast, when R_m is higher than R_s , r_{min} is increasing at a higher rate than R_m . As a result, the difference between r_{min} and R_m is maximised when R_m is equal to R_s , see Figure 3(b). Therefore, R_s can be estimated by varying R_m until the difference between r_{min} and R_m is maximised.

This method of estimating R_s works for a range of R_s

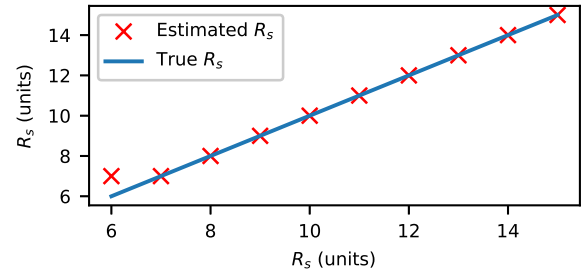


Figure 4: Estimated R_s as a function of true R_s

values. Figure 4 shows that the method introduced is capable of correctly guessing R_s in the range of 7 to 15 units. When $R_s < 7$, the measurements of r_{min} are more noisy, affecting the accuracy of the method.

Conclusions

This study suggests that it is possible to study the interactions between agents in a swarm of interest by infiltrating the swarm with a robot instead of monitoring the swarm with an external system. In the case of the model selected in this work, the repulsion radius of the swarm agent can be estimated by varying the repulsion radius of the monitoring agent until a specific targeted value is maximised.

However, the conditions in which this method was tested are limited, and there are many issues to address before a practical implementation would be possible. Firstly, it was tested in the selected model of interaction. Also, the swarm agents have to perceive the monitoring agent as one of their own. In the future, we aim to expand the parameters and rules that can be inferred by the monitoring agent, as well as expanding to swarm behaviours behind flocking. We also hope to study the benefits of applying the information gained to manipulate the swarm of interest.

References

- Couzin, I. D., Krause, J., James, R., Ruxton, G. D., and Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218(1):1–11.
- Fine, B. T. and Shell, D. A. (2013). Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, 35(2-3):195–219.
- Hauert, S., Mitri, S., Keller, L., and Floreano, D. (2014). Evolving cooperation: From biology to engineering. Technical report, MIT Press.
- Li, W., Gauci, M., and Groß, R. (2016). Turing learning: a metric-free approach to inferring behavior and its application to swarms. *Swarm Intelligence*, 10(3):211–243.
- Puckett, J. G., Kelley, D. H., and Ouellette, N. T. (2014). Searching for effective forces in laboratory insect swarms. *Scientific reports*, 4:4766.

Convolutional Neural Networks for Cellular Automata Classification

Eric Silverman

MRC/CSO Social and Public Health Sciences Unit,
University of Glasgow, Glasgow G2 3AX, UK
eric.silverman@glasgow.ac.uk

Abstract

Wolfram famously developed a four-way classification of CA behaviour, with Class IV containing CAs that generate complex, localised structures. However, finding Class IV rules is far from straightforward, and can require extensive, time-consuming searches. This work presents a Convolutional Neural Network (CNN) that was trained on visual examples of CA behaviour, and learned to classify CA images with a high degree of accuracy. I propose that a refinement of this system could serve as a useful aid to CA research, automatically identifying possible candidates for Class IV behaviour and universality, and significantly reducing the time required to find interesting CA rules.

Some Cellular Automata (CAs) are capable of unexpectedly complex behaviour despite being based on simple rules. Wolfram's early investigations into the behaviour of one-dimensional Elementary CAs (ECAs) led him to propose a four-way qualitative classification scheme for CAs according to the behaviour they exhibit from random initial conditions (1984): Class I CAs converge to a homogeneous state; Class II CAs generate periodic or stable structures; Class III CAs generate random or chaotic patterns; and Class IV CAs produce complex localised structures.

The CA community has devoted a great deal of effort toward understanding these classes of behaviour, refining their definitions, and cataloguing interesting CA rules (see e.g. Li et al., 1990). However, determining whether a CA is in Class IV is not only difficult, it is undecidable (Culik and Yu, 1988). Nonetheless, classifying CAs in this way, or with related schemes, remains a useful and succinct way to describe the behaviour observed in CAs qualitatively, and identify CAs worthy of further exploration.

Below I demonstrate how a convolutional neural network (CNN) can successfully classify one-dimensional CAs into the four Wolfram classes.

Methods

CNNs were chosen for this task due to their notable success in computer vision tasks (Krizhevsky et al., 2012). This CNN takes greyscale images of CA behaviour as input, and outputs the Wolfram class of the image. The first iteration,

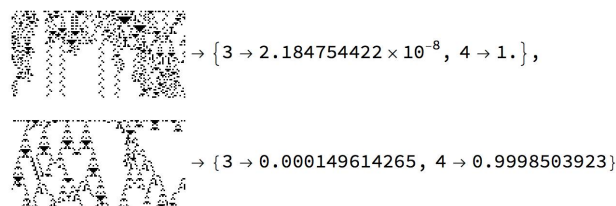


Figure 1: Classification certainty of Network II on novel CA examples. Network II successfully classified 99.8% of 6,145 novel images in the test set.

Network I, consisted of nine layers: a $16 \times 59 \times 118$ convolution layer; $16 \times 59 \times 118$ ReLU layer; $16 \times 58 \times 116$ convolution layer; $16 \times 58 \times 116$ ReLU layer; $16 \times 1 \times 1$ pooling layer; 16-node flattening layer; 256-node fully-connected layer; 4-node fully-connected layer; and 4-node SoftMax layer¹. The network was built in Mathematica 12 and trained on a set of 32,768 120×60 -pixel images of the evolution of all 256 elementary CA rules from different random initial conditions. Network I was trained for two hours on four CPU cores, at which point accuracy on the training set had exceeded 99.93%.

A second CNN with the same structure, Network II, was trained on 3- and 4-colour totalistic CA rules as well as the ECA data, in order to increase the network's exposure to Class IV behaviour. The network was built in Mathematica 12 and trained on a new set of 16,384 ECA examples, plus an additional 8,192 samples drawn from eight known Class IV 3- and 4-colour totalistic rules (codes 1041, 1388, 1635, 1815, 2007, 2043, 2049, and 1004600). This network was trained for two hours using four CPU cores, at which point accuracy on the training set exceeded 99.2%.

Network III, again using the same structure, was trained on larger images with a resolution of 200 by 120 pixels. The training set included an additional 8,192 images generated

¹This architecture was inspired by a project by Thales Fernandes on identifying ECA rules from images: <https://community.wolfram.com/groups/-/m/t/1417114>

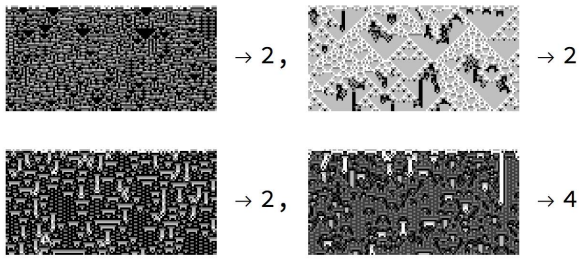


Figure 2: The four highest-entropy inputs to Network II from a test set of $k=5$, $r=1$ examples.

from $k=5$, $r=1$ totalistic rules to provide further examples of Class IV behaviour². This network was trained for three hours on four CPU cores, at which point accuracy on the training set exceeded 99.7%.

Results

Network I achieved an accuracy of 100% on a test set composed of 4,096 additional ECA and totalistic CA examples (see Figure 1). Testing with images from previously unseen $k=4$, $r=1$ totalistic CAs produced poor results (18.6% accuracy), with Class IV inputs frequently misclassified as Class II/III.

Network II was trained with the larger data set in an effort to improve generalisation. This network correctly classified 99.8% of a test set of 6,145 novel images of Class IV 3-colour totalistic rules; Figure 1 shows some examples. The network was not able to generalise to 5-colour CAs, classifying a set of 1,024 examples with 57.6% accuracy. Figure 2 shows four of the highest-entropy $k=5$, $r=1$ test set inputs (where the class determination was the most uncertain). The high-entropy inputs from the test set suggest the most common errors are mistaking visually complicated Class II CAs for III or IV, and Class III CAs for Class IV.

Network III, having been trained on images derived from 5-colour CAs, was 99.9939% accurate on a 16,384-image test set including 2, 3, 4 and 5-colour CA images; in the entire test set only one image was misclassified (a Class III was mistaken for Class IV). For this network, the highest-entropy images from the test set were all Class IV examples, but they were all correctly classified (see Figure 3).

Conclusions and Future Work

Network II's ability to generalise to new examples of Class IV behaviour in the space of 3-colour and 4-colour totalistic rules suggests that CNNs can be a useful aid in the search

²These data were generated by rules found through a random search and classified manually. Details of the CA rules and methods used in this project can be found at <https://github.com/thorsilver/Neural-Networks-for-CA-Classification>

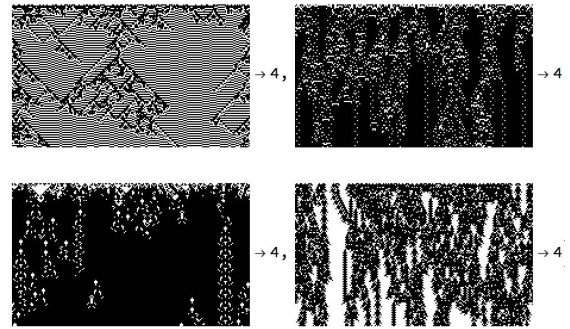


Figure 3: The highest-entropy images in the test set according to Network III. All but one of the images in the 16,384-image test set were correctly classified.

for CA complexity. A suitably-trained CNN could identify likely Class IV candidates quickly and significantly reduce the numbers of CA rules that need to be manually checked. This CNN can make classification judgments using only a single visual input, as compared to other methods that may require time-consuming statistical tests.

Network III demonstrates that only a few examples of behaviour in higher-complexity rule spaces are necessary to allow the network to identify Class IV rules in that space. However, success in 6-colour CAs remains elusive; these CAs are nearly universally seen as Class IV by the network due to their increased visual complexity. Work is ongoing with deeper networks trained on larger images, with the goal of producing a network capable of detecting candidate Class IV behaviour even in unexplored CA rule spaces.

Acknowledgements

Eric Silverman is part of the Complexity in Health Improvement Programme supported by the Medical Research Council (MC.UU.12017/14) and the Chief Scientist Office (SPHSU14). Thanks are due to Stephen Wolfram and Hector Zenil for their valuable comments and advice on this work.

References

- Culik, II, K. and Yu, S. (1988). Undecidability of CA classification schemes. *Complex Systems*, 2(2):177–190.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.
- Li, W., Packard, N. H., and Langton, C. G. (1990). Transition phenomena in cellular automata rule space. *Physica D: Nonlinear Phenomena*, 45(1-3):77–94.
- Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35.

Biochemical networks display universal structure across projections and levels of organization

Harrison B. Smith¹, Hyunju Kim² and Sara I. Walker^{2,3,4,5}

¹Earth-Life Science Institute, Tokyo Institute of Technology, Meguro-ku, Tokyo, Japan

²Beyond Center for Fundamental Concepts in Science, Arizona State University, Tempe, AZ, USA

³School of Earth and Space Exploration, Arizona State University, Tempe, AZ, USA

⁴ASU-SFI Center for Biosocial Complex Systems, Arizona State University, Tempe, AZ, USA

⁵Blue Marble Space Institute of Science, Seattle, WA, USA.

hbs@elsi.jp

Abstract

Biochemical reactions underlie all living processes. Like many systems, their web of interactions is difficult to fully capture and quantify with simple mathematical objects. Nonetheless, a huge volume of research has suggested many real-world systems—including biochemical systems—can be described simply as ‘scale-free’ networks, characterized by power-law degree distributions. More recently, rigorous statistical analyses upended this view, suggesting truly scale-free networks may be rare. We provide a first application of these newer methods across two distinct levels of biological organization: analyzing an ensemble of biochemical reaction networks generated from 785 ecosystem-level metagenomes and 1082 individual-level genomes (representing all domains of life). Our results confirm only a few percent of biochemical networks meet the criteria necessary to be more than super-weakly scale-free. We perform distinguishability tests across individual and ecosystem-level biochemical networks and find there is no sharp transition in the organization of biochemistry across distinct levels of the biological hierarchy—a result that holds across network projections. This suggests the existence of common organizing principles operating across different levels of biology, which can best be elucidated by analyzing all possible coarse-grained projections of biochemistry in tandem across scales.

Introduction

Broido and Clauset recently developed a methodology to compare the degree distributions of network projections of different complexities, classifying the degree to which they are scale-free on a scale from “Not scale-free” all the way to “strongest” [1]. This provides a framework for statistically analyzing many projections of a given system to determine how well scale-free structure describes the real underlying system when projected onto its different coarse-grained representations.

Herein, we build from the work of Broido and Clauset with specific application to the problem of characterizing biochemical systems. A novelty in our approach is recognizing that in order to really understand the structure of real-world biological systems, the relevant scale(s) for per-

forming such analysis must also be considered. In particular, many biological systems are hierarchical, with networks describing interactions across multiple levels. For example, one may study the biochemistry of individual species, but ultimately the function of an individual in a natural system depends on a complex interplay of interactions among the many species comprising its host ecosystem. In this way, biochemistry is hierarchically organized into individuals and ecosystems. Indeed, much discussion about universal properties of life has shifted focus from individuals to ecosystems as the relevant scale best capturing the regularities of biological organization [2, 3]. It is unclear at present whether analysis of biochemical networks at the level of individuals or ecosystems will best uncover their structure and permit identifying generative mechanisms for biology, or whether all levels must be considered simultaneously.

Results

Utilizing the framework developed by Broido and Clauset [1], we perform statistical analysis of an ensemble of biochemical systems generated from 785 ecosystem-level metagenomes and 1082 individual-level genomes (representing all three domains of life). We use the full set of biochemical reactions encoded in each (meta)genome to construct eight distinct network representations of each respective biochemical system. This resulted in 8656 network projections for the 1082 individual-level biochemical datasets, and 6280 network projections for the 785 ecosystem-level biochemical datasets. We determine whether or not these datasets are scale-free, and analyze the aspects of them, and their diverse projections, that tend to lend themselves to be more or less scale-free.

Our results indicate most biochemistry at the individual and ecosystem-level is characterized by networks that are “super-weakly” scale-free (**Fig. 1**). That is, while the power-law is better than other models for fitting the shape of their degree distributions, the power-law is not itself a good model. When doing a goodness-of-fit test, we find that the majority of network representations of each ge-

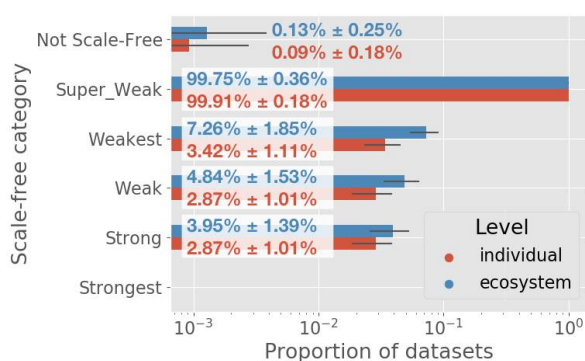


Figure 1: **The vast majority of individual and ecosystem level networks are not "scale-free"**. Most datasets are super weak, indicating that when compared to other models, a power-law distribution is a better fit. However, the power-law distribution is not a "good" fit for most dataset network representations. Overlaid values show the percent of networks of each level which fall into each category ($\pm 2SD$).

nomic/metagenomic dataset have $p < 0.1$, indicating there is less than a 10% chance that our data is truly power-law distributed. This effectively rules out the possibility that our data is drawn from a power-law shaped degree distribution, despite the fact that, when compared to other distributions through log-likelihood ratios, 99% of all datasets do not favor alternative heavy tail distributions for the majority of their network-projections

While other literature [4, 5] has advocated for unipartite networks (with all compounds participating in a reaction connected), we find that these networks overestimate power-law goodness-of-fit p -values and α values compared to reaction and bipartite networks.

Furthermore, we demonstrate that using random forest distinguishability analyses on a combination of all the results of scale-free analyses completed in this work can predict, better than chance, whether the data comes from individuals or ecosystems.

Discussion

The fact that multiple levels and multiple projections of biochemistry reveal common structure suggests universal principles may be within reach if cast within an ensemble theory of biochemical network organization.

Whether or not the observed structure is truly a universal property of life's chemical systems is more difficult to conclude. Achieving such a theory requires recognizing that, unlike simple physical systems where statistics over individual components is sufficient to describe and predict their behavior, biological systems require additional information about the structure of interactions among their many components. Perhaps incorporating additional reaction data on

the particular flows of atoms between compounds would further elucidate regularities of biological structure. But how to project this structure onto simple mathematical objects that can be quantifiably characterized and compared remains a central problem of complex systems science. In physics, the relevant coarse-graining procedure is well understood, but we are not so far in complexity science: the first hurdle we must traverse is to identify the proper coarse-grained network representations for analysis. Existing literature cautions against using unipartite network projections, as it is argued they can lead to "wrong" interpretations of system properties such as degree in biochemical networks [6, 7]. We find instead that whether or not this conclusion should be drawn is highly dependent on the particular characteristics of degree or the degree distribution under consideration.

The similarities and differences in the structure of different projections provides insight into the actual structure of the underlying system of interest. In physics we have become accustomed to one unique coarse-grained descriptor providing insight into the structure of a system. It may be that to really understand complex interacting systems, such as the systems of reactions underlying all life on Earth, we must forget the allure of simple, singular models. Instead, to characterize the regularities associated with living processes, we should perform statistical analyses over many (still relatively simple) coarse-grained projections.

Acknowledgements

We thank the Emergence@ASU team (especially Doug Moore, Cole Mathis, and Jake Hanson) for feedback through various stages of this work.

References

- [1] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature communications*, 10(1):1017, 2019.
- [2] Eric Smith and Harold J Morowitz. *The origin and nature of life on Earth: the emergence of the fourth geosphere*. Cambridge University Press, 2016.
- [3] Hyunju Kim, Harrison B Smith, Cole Mathis, Jason Raymond, and Sara I Walker. Universal scaling across biochemical networks on earth. *Science Advances*, 5(1):eaau0149, 2019.
- [4] Petter Holme. Model validation of simple-graph representations of metabolism. *Journal of The Royal Society Interface*, 6(40):1027–1034, 2009.
- [5] Petter Holme and Mikael Huss. Substance graphs are optimal simple-graph representations of metabolism. *Chinese Science Bulletin*, 55(27-28):3161–3168, 2010.
- [6] Steffen Klamt, Utz-Uwe Haus, and Fabian Theis. Hypergraphs and cellular networks. *PLoS computational biology*, 5(5):e1000385, 2009.
- [7] Raul Montanez, Miguel Angel Medina, Ricard V Sole, and Carlos Rodríguez-Caso. When metabolism meets topology: Reconciling metabolite and reaction networks. *Bioessays*, 32(3):246–256, 2010.

Toward the Self-Organisation of Emergency Response Teams Based on Morphogenetic Network Growth

Nicolas Toussaint^{1*}, Emma Norling² and René Doursat¹

¹Centre for Advanced Computational Science, Manchester Metropolitan University, Manchester, UK

²Department of Computer Science, University of Sheffield, Sheffield, UK

*corresponding author: nicolas.toussaint@stu.mmu.ac.uk

Abstract

Focusing on the challenge of fostering the self-assembly of socio-technical networks, we present the application of Morphogenetic Engineering principles in this domain to a 2D spatial case study involving a team of first responders. Our model and simulation illustrate how members of a rescue team could be guided via hand-held devices toward better coordination and positioning at appropriate locations, based on peer-to-peer communication and local landmarks in the environment (such as incidents or exits), without the need for a centralised control centre. Using Raspberry Pi devices, we illustrate this scenario in various situations that require quick decision-making to control and manage. Our work suggests the possibility of novel forms of bottom-up self-organisation among groups of users and machines, in contrast to top-down imposed hierarchies and policies.

Introduction

Crises are inevitable. They range from large-scale wildfires to injured individuals in a crowd. The ability to respond and manage a crisis properly is crucial to limit casualties, prevent further damage, and help minimise subsequent risks. The specificities of each emergency scenario, which often includes misleading or incomplete information, make the coordination of rescue services particularly difficult. Compounding the problem, crisis situations can change rapidly as new events also occur unexpectedly (secondary hazards). Emergency responders must react and adapt to a dynamically shifting situation in real time by modifying their plans accordingly. For example, if someone breathes undetected toxic gas during an operation and is incapacitated, other agents must react by protecting themselves, evacuating everyone else from the new risk area, and take care of the injured. Detrimental interference among agents' actions may also arise from simultaneous task assignments leaving each other unfinished, or when a new urgency makes agents deviate from a global objective. Despite all these pitfalls, it is crucial to be able to constantly reevaluate the dynamic situation and make appropriate decisions toward reducing impact on human life and infrastructure.

So far, many research works focused on the amount of data and information that can be collected and processed in

real time to improve the assessment of emergency situations and decision-making accuracy. The recent development of new technologies has contributed to making this endeavour increasingly affordable. New methods range from pervasive sensors found in everyday devices to fast communication and sophisticated AI algorithms able to aggregate multiple streams of data and recognise patterns. In an emergency context, the Internet of Things (IoT) is thought to have a positive impact (Yang et al., 2013), whether during preliminary evaluation of an incident, critical to a plan-based approach (Chen et al., 2008), or during the course of an event to track and monitor people's actions, locations, and considerable other information (Scheurer et al., 2017; O'Flynn et al., 2018; Ciabattini et al., 2017).

For instance, 3D mapping of a disaster inferred from data gathered by bystanders (Sadhu et al., 2016) can help design an appropriate strategy. Indoor activity can be reconstructed from correlated sounds and inertial data captured with wearable devices, or outdoor drones. Such technologies could detect a first responder lying on the ground, coupled with perturbations in their vital signs, which would prompt their peers to treat this as a new incident and/or casualty. Active landmarks (mobile sensors) can also be placed by the agents (Palmieri et al., 2016) at the start of an operation to obtain a more precise picture of the unfolding incident. Information can also be extracted from social media feeds (Conrado et al., 2016). Smart systems have been designed to gather data from a variety of sources (Yuan and Detlor, 2005) and analyse the information they contain (Bartoli et al., 2015), paving the way toward an emergency response information system (ERIS) that can handle big data.

Information-gathering hardware and software systems also have shortcomings. Problems include difficulty to use, the need for special training and maintenance of key technological components, a high price tag, and lack of reliability in critical circumstances. Moreover, the flood of information sent to a main hub, the "Emergency Operations Centre" (EOC), from an ERIS can become rapidly overwhelming. This information is used centrally to design an appropriate strategy and decide which actions or movements need to be

undertaken to achieve an effective response. Task assignments are then decided and communicated back to the corresponding field agents. However, both decision-support systems and decision makers can be overburdened by the huge number of input and output features. The complexity of innovative information technologies often occupies too much of the EOC agents’ cognitive space (Weidinger et al., 2018).

In this paper, we argue that self-organisation properties can be exploited and put to work to alleviate these difficulties. Endowing field agents with more autonomy and local power can greatly support an EOC toward achieving efficient coordination. In fact, reversing the terms, the EOC would only assist the bottom-up emergence on the ground, or be required when treating radically disruptive global events. Self-organisation properties allow the system to negotiate new environmental and organisational perturbations without falling into an “error state” (Serugendo, 2009), in which carefully crafted rules assigned to each agent have been rendered obsolete. Disturbances can involve disconnection of agents from the ERIS network, misunderstandings between agents, and so on. Typically, the responsibility to detect and identify such events would fall to the EOC (Chen et al., 2008), which would have to modify agents’ instructions to adapt to the dynamically evolving situation. The measures taken should then steer the system back into a “rational state” in which coordination is considered efficient again. By contrast, self-organised systems are composed of many independent entities (here, field agents) interacting with each other and their local environment, and creating a collective behaviour that may also fulfil a goal such as the emergence of a functional structure. All entities follow the same set of instructions with possible variants. The outcome essentially relies on the multitude of local choices made by the agents via their interactions, instead of centrally assigned instructions unable to take into account sudden problems or failures.

There are only few examples of self-organised emergency response systems. One of them features adaptive collaboration without global awareness in a scenario of search and rescue, where dynamically changing factors determine the likelihood for help requests to be sent or accepted (Fraseri et al., 2018). In this work, however, the situation does not require the formation of a coherent modular network structure. Collective structure formation can be helped by models based on *gradient* propagation (Doursat, 2008; De Wolf and Holvoet, 2006), i.e. the peer-to-peer exchange and mutual update of positional information values. This concept inspired by developmental biology has been invoked to propose self-organising robotic swarms (Doursat and Sánchez, 2014) and socio-technical networks (Ulieru and Doursat, 2011), in which complex and persistent structures emerge from connectivity choices and “hop counters” propagated among nodes. A developmental approach based on self-assembly mechanisms controlled by the same ruleset or

“genome” in every agent combines the advantages of robustness and adaptability of the growing structure.

While this article does not intend to offer an alternative to existing centralised systems, it aims to show how they could be complemented by a layer of decentralisation. It also illustrates the feasibility of applying an abstract model of network self-assembly to a more concrete spatial and distributed context via idealised emergency scenarios. Finally, it makes a step toward more practical use cases that employ Raspberry Pis (RPis) to represent devices carried by field agents. The rest of the text is organised as follows. First, we introduce the abstract model and its customisation to a 2D environment; then, we describe the experimental software/hardware setup of our application; finally, we discuss the results obtained so far and sketch out future work.

Model

In this section, we briefly introduce the abstract model of programmable network self-organisation (Ulieru and Doursat, 2011) on which the present work is based. Then, we present a customisation of this model to 2D space and its adaptation to a distributed external environment.

Abstract Model Overview

The core of the model is a feedback loop going through three steps corresponding to the three main components of network self-assembly: *links*, *ports* and *gradients*. All nodes of the network possess a predefined set of input/output pairs of ports, denoted by (X, X') , (Y, Y') , and so on. Each port is in one of the following states: ‘closed’ (not accepting connections), ‘open’ (accepting) or ‘busy’ (open but not accepting). A link can connect two nodes i and j via open and matching output/input ports only, e.g. $X'_i \leftrightarrow X_j$. Optional “port tagging” can further restrict connections during runtime by creating specialised port subtypes and mutual compatibility rules, e.g. X'_i^a cannot connect to X_j^b . Nodes exchange integer gradient values over the links, denoted by (x, x') and so on, to convey positional information. These values are hop counters that increase by 1 every time they cross over to another node. Each node calculates the minimum of all received values within a given port type (and other correction rules apply). A small program, or *ruleset*, is embedded in each node, taking in input local gradient and environmental values and opening/closing ports in output. The ratio of both values tunes the propensity of the structure to react to the environment. This common ruleset constitutes the “genome” of the network, as it controls the patterns of connectivity in each node based on information received from the neighbours—hence the “shape” of the network.

In sum: (a) gradient values are exchanged and updated (G step), possibly several times; then, (b) ports may be opened/closed depending on these values (P step); and (c) new nodes may be added/removed and linked/detached where ports are available/busy (L step); then these steps are

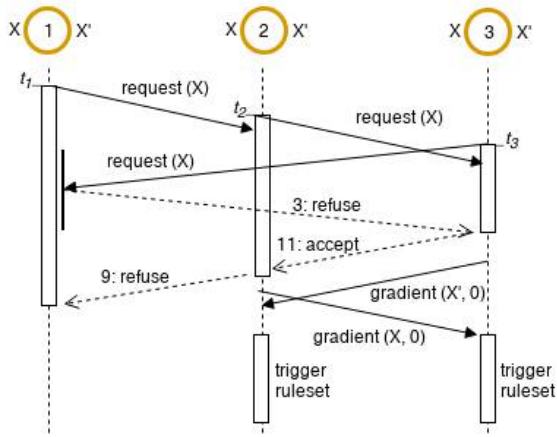


Figure 1: A request-sending scenario involving three nodes. Each answer is tagged with the corresponding line number from Algorithm 1. Here all nodes possess the same open ports X and X' , and exchange requests to find matching ports to connect to. A request is sent to the opposite port of a recipient node, which may wait for an answer to its own request before replying in turn. In this case, node 1 declines a request as $t_3 > t_1$, triggering a cascade of other accept/decline answers. Eventually, a link is created between nodes 2 and 3, which then start exchanging gradient values.

Algorithm 1 Request Receiving

```

1: procedure IS_VALID(timestamp, value)
2:   if (timestamp > local_timestamp or
       not has_compatible_port(value) or
       has_waiting_request) then
3:     return false
4:   else
5:     has_waiting_request  $\leftarrow$  true
6:     state  $\leftarrow$  wait_main_thread()
7:     if has_changed(state) then
8:       has_waiting_request  $\leftarrow$  false
9:     return false
10:  else
11:    return true

```

repeated. Complex deterministic structures can emerge from the change in time of all node states, guided by their common ruleset. The graph builds itself from the bottom up following the sequence of all alternatives that nodes face and the actions they take based on local information only.

Distribution and Concurrency

To prevent interference and race conditions among nodes during these steps, which would put the network into an inconsistent state, we also distinguish between different timescales. In this framework, gradient propagation G must perform faster than program execution P inside each node, itself faster than link creation L . The implication is that several rounds of G routines can be processed before one

round of P is done; and similarly several P 's before one L . Clearly, if it were not the case then concurrent changes in two different parts of the network could become incompatible, i.e. happen too fast for each one to take into account the other. While this could be remedied by running the algorithm on a central host based on a sequential node scheduler, it will not be robust in a truly asynchronous scenario where each node runs on its own device.

Once this temporal hierarchy is established, no concurrency issues remain within the first two steps, and further synchronisation is not required. Gradient updates and propagation G are atomic operations independent from the previous state, so that the specific order in which values are sent and received is not critical. Port operations P only change the nodes' internal states locally and do not immediately affect peer nodes (that is, before L is triggered). Link creation L , however, can still harbour race conditions since several nodes can be simultaneously sending and receiving requests for new connections and each outcome can influence subsequent behaviour. This may cause problems such as when a node accepts an incoming request at the same time that its outgoing request to create a link is accepted on the same port, yet the port can only hold one link; or when two competing ports receive links at the same time, whereas one should have caused the other to be closed and vice-versa.

To create new links, nodes regularly broadcast requests containing a list of their open ports to their peers. Recipients compare this list to their own available ports to identify matching pairs. If at least one match is found, the node sends back a positive response with the port identified and both nodes acknowledge the new link. To prevent concurrent link requests, we designed a priority mechanism (Algorithm 1) which assigns an order to these requests. It enables nodes to process them sequentially and individually, delaying responses for other link requests until completion of the current one. The sequential order is determined by two criteria (ordered by importance): 1) the origin of the requests ('sent' requests are ranked higher than 'received' requests); and 2) the time stamp (the older the request, the higher the ranking). Prioritising sent requests ensures that the ports advertised in the requests are kept open until the reception of the response. In the case where a node can accept a received request after sending its requests to other nodes, leading to port closures, recipients would then have to check that matching ports are still available before creating a link, generating additional communication. This criteria however only gives a local order, which can be contradictory across different points of view. For example, if node i sends a request to node j and vice-versa, and both are prioritising a different request, this will create a deadlock. The second criterion based on time stamps prevents this problem by creating a global priority order. Furthermore, to reduce unnecessary waiting, agents only set aside the first compatible request received while immediately declining the remaining requests.

Embedded in Space

While the previous model was designed at an abstract level for all kinds of graph topologies, we propose here to apply it to 2D Euclidean space. In this context, new common features are added such as the *positions* of nodes and objects, and an equation to *move* nodes in space, derived from distance-based interactions and forces among agents and environmental landmarks. Nodes represent agents in space and self-organise into complex shapes guided by their common genetic ruleset, with possible variations depending on the environmental cues they encounter. In the examples presented here, these external influences have been placed in such a way that the simple rules followed by the agents (e.g. connecting to peers, accepting requests, moving in certain directions) create “interesting” structures. In the remainder of the text, we use the term ‘agent’ indifferently to refer to a ‘node’ or its associated ‘field agent’.

Interactions: Interactions between agents in 2D are the same as in the abstract method. Distances are not taken into account during communication as nodes are actually relying on a pervasive broadcasting system with ID-tagged messages to emulate point-to-point connections. However, this assumption could be weakened to constraint the channels of communication to a certain region of space based on a maximum radius of signal reception, which would impose limits on the possible combinations of nodes.

Agents are now also able to perceive high-level features (such as casualties, fire exits, etc.) in their local environment via *sensors*. Sensors are activated whenever specifically associated features are present in the detection range, in which case appropriate forces are applied on the agent. Depending on the “mode” of the sensors, these forces are either attractive, aiming to keep the agent within the proximity of a feature of interest, otherwise they are repulsive. Agents can also adjust external influences, following their ruleset, by changing a sensor mode as needed. For example, at first an incident can attract agents to encircle it; but when the area becomes overcrowded, rules change the sensor mode of unneeded agents and, as a result, scatter them over a large area. This strategy increases the region covered by perceptual agents, which improves the system’s overall response to possible secondary hazards. An agent may also dispose of its sensors, i.e. desensitise itself to certain types of landmark and specialise its behaviour.

Movements: Agents move to create *spatial* chains and graphs that follow the corresponding abstract *topological* chains and graphs created by the G-P-L connectivity routines, and based on an optimal length for each link. For example, if there is a chain of three agents connected through their (X, X') ports, they should move in 2D space in such a way that the middle node positions itself between the other

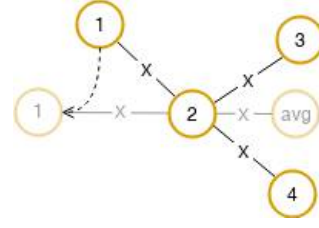


Figure 2: Example of local connectivity configuration. Node 1 only has one connection (on the X' port), while it requests the coordinates of node 2 and the centre of mass of nodes 3 and 4 in order to calculate its new position (see text). In this case, it ends up being on the other side of node 2 with respect to the other nodes.

two (details below). In addition, nodes can also be attracted toward, or pushed away from, particular landmarks that they sense in the environment. Free agents disconnected from their peers can move in arbitrary ways. Combined together, neighbourhood and landmark interactions give rise to forces that reshape the network’s spatial structure in accordance with both internal connectivity and external environment. However, converging to a global spatial layout that satisfies all these constraints can be impossible when they are contradictory, resulting in a frustrated state. In this case, a suboptimal equilibrium characterised by a nonzero but small global force can still be achieved. In sum, application of spatial rules is intertwined with the connectivity routines to ultimately drive the graph growth.

The core of the motion algorithm is as follows: at each internal step, an agent i asks each one of its connected peers k to send (a) its position (denoted by \vec{U}_k in Eq. 1) and (b) the average of the positions of k ’s “output” peers (from the viewpoint of i as an “input”) via the same type of link (denoted by \vec{U}'_k). In the example of Figure 2, node 1 sends a positional request to node 2 via the X' port; then node 2 asks its linked neighbours (here 3 and 4) their positions via the X' port. The total displacement vector of i is calculated from this information, possibly combined with the coordinates of certain landmarks \vec{U}_l detected by the sensors:

$$\Delta\vec{U}_i = \frac{1}{n} \left(\sum_{k=1}^n \vec{V}_k + \alpha \vec{W}_k \right) + \frac{1}{m} \sum_{l=1}^m \beta \vec{W}_l \quad (1)$$

where node \vec{U}_i has n neighbours and m landmarks in its detection range; \vec{V}_k and \vec{W}_k , described below, represent two types of displacements generated by “forces” among agents; \vec{W}_l reflects the same second type of force applied to agents by landmarks; and the α and β factors (resp. 4 and 1 here), adjust the balance of these three constraints.

$$\vec{V}_k = \begin{cases} (\vec{U}_k - \vec{U}'_k) + (\vec{U}_k - \vec{U}_i) & \text{if } \vec{U}'_k \neq \vec{0} \\ \vec{0} & \text{otherwise} \end{cases} \quad (2)$$

\vec{V}_k drives node i toward a position symmetrically disposed from $\vec{U}_k^{(i)}$ with respect to U_k (if U_i indeed has second-degree neighbours beyond k , otherwise \vec{V}_k has no effect).

$$\vec{W}_k = \left(\|\vec{U}_k - \vec{U}_i\| - d_0 \right) \frac{\vec{U}_k - \vec{U}_i}{\|\vec{U}_k - \vec{U}_i\|} \quad (3)$$

\vec{W}_k (resp. \vec{W}_l) is a vector pushing node i at an optimal distance d_0 from node k (resp. landmark l). This distance depends on the type of the element, whether it is a node or a landmark, and whether the latter is attractive or repulsive. In the case of repulsive landmarks, the distance is set to a high value such that the node is driven out of the landmark's detection range and influence.

Finally, the resulting displacement vector $\Delta\vec{U}_i$ is constrained by the motion capacity of the node represented by a maximum distance d_{\max} , and must be renormalised if it exceeds this distance:

$$\Delta\vec{U}_i \leftarrow \begin{cases} d_{\max} \frac{\Delta\vec{U}_i}{\|\Delta\vec{U}_i\|} & \text{if } \|\Delta\vec{U}_i\| > d_{\max} \\ \Delta\vec{U}_i & \text{otherwise} \end{cases} \quad (4)$$

Gradient loops: One potential problem with this model that must be avoided is the development of loops exclusively composed of one port type, as this would lead to an infinitely increasing gradient propagation. In the abstract model, where a single graph grows from the incremental addition of nodes, one at a time, blocking strategies can prevent this problem by closing specific ports at appropriate instants. For example, chains can be constrained to expand in only one direction (e.g. either X or X'), thus preventing both ends from connecting together and forming a loop. In the asynchronous 2D case, however, nodes are already all present as physical entities at the start, and several nodes may potentially trigger concurrent graph growths simultaneously. Therefore, embryonic structures at equivalent stages of development should be able to cooperate and merge into a larger graph to avoid unnecessary duplicates (e.g. several arcs of circular node chains forming around the same landmark). However, since port closing does not discriminate between loop development and graph merging, another mechanism must be implemented into the model to specifically prevent single-port loops. To this goal, we augment the gradient's information with the identifier of its source node. As a result, each extremity of a chain also possesses the node ID of the other end. The link creation routine L is also amended as follows: when a node broadcasts to its peers connection requests containing a list of its open ports, it also verifies for each peer that it is, precisely, *not* the source of the opposite gradient, otherwise that peer must be ignored.

Experimental Setup

In this section, we present a software and hardware implementation of the model described above with a case study to

illustrate its behaviour. First, a reduced scenario including a small number of physical agents, here eight RPis, shows the applicability of the model to a distributed environment and how it can lead to a real-world application. Then, a more comprehensive setup using simulated agents demonstrates complex self-organisation. In both scenarios, we focus here on the main features of the interactions between agents and for now ignore realistic details such as obstacles.

Model Implementation

Throughout the remainder of the paper, a *software agent* (SA) refers to an autonomous program that can run individually on a field agent's device or among other SAs on a central computer. In both cases, SAs embody the nodes of our model, whereas *environmental agents* (EAs) represent external landmarks that can affect the settings or evolution of a simulation and its real-world deployment.

Environmental agents and EOC: In a full-fledged distributed system, each EA would only contain a partial description of the local physical environment surrounding a landmark. For the needs of our implementation, however, where generic RPis stand in for field devices but lack actual sensing/actuating capabilities, we revert some of the decentralisation principles and concentrate the EAs into a single EOC running on a laptop. This EOC also contains the common ruleset specified at the start and broadcast to all SAs. As both the list of landmarks and the agent's rules can change during the simulation, the latter is broadcast again while the former is kept centrally.

After each move, an SA sends to the EOC a list of its active sensors. The EOC responds with the new state of each sensor, associated with the landmarks' positions if any has been detected. Another role of the EOC is to display a global view of the current simulation to be observed and monitored by the modeller. To this purpose, software agents send a notification after every action (such as gradient value updates, link creations, movements, etc.), which are then aggregated by the EOC into a global picture.

In summary, while landmarks should not normally be provided by the EOC but detected directly by the field agents, the ability to continually broadcast ruleset updates and provide a visualisation panel remain useful features in a concrete scenario. The states and positions of field agents can be used alongside other sources of information by the EOC to improve the situational awareness of possible central decision-makers in real time. The broadcasting functionality enables the EOC to react to critical changes by overriding outdated rulesets.

Software agents: SAs are given a code that determines how they interact, process the information received and make decisions. While the architecture and program of the model's core G-P-L routines is immutable, the modeller can

Algorithm 2 Agent Ruleset

```
1: open  $X$ , close  $X'$ ,  $Y$ ,  $Y'$ 
2: open  $C$ , close  $C'$ 
3: turn_on  $S_c, S_x$  ▷ casualty and exit sensors
4: circle_size  $\leftarrow 8$ 
5: if  $s_c = 0$  then ▷ if no casualty detected
6:   open  $X$ 
7:   set_attractor  $S_c$ 
8: if  $s_c > 0$  and  $x < \text{circle\_size}$  then
9:   open  $X, X'$ 
10:  close  $C, C'$  ▷ prioritise encircling
11:  if  $x > 0$  then
12:    close  $Y$ 
13:  if  $x = (\text{circle\_size} - 1)$  then
14:    open  $Y', C', X'$ -tagged-by- $s_c$ 
15:  if  $x' = (\text{circle\_size} - 1)$  then
16:    open  $Y, X$ -tagged-by- $s_c$ 
17: if  $x \geq \text{circle\_size}$  and  $x' = 0$  then
18:   close  $X, X', Y, Y'$ 
19:   set_repellent  $S_c$ 
20: if  $c = 0$  then
21:   turn_off  $S_x$ 
22: if  $c > 0$  then
23:   open  $C'$ , turn_on  $S_x$ 
24: if ( $s_x > 0$  and  $c > 0$ ) or ( $c = 0$  and  $x = 0$  and  $c' > 0$ )
    then
25:   close  $C'$  ▷ adaptive chain length
```

focus on the “genomic” ruleset embedded inside P, which is specific to each use case. Unlike many agent-based modelling solutions, composed of a set of agents in a shared container and randomly activated by a scheduler, each individual SA is run in a process of its own. It also possesses its own graphical interface to display practical information to the field agent (more details in Results). For this experiment, we used versatile RPIs to host the SAs. For real-world use by field agents, these devices could be later augmented with sensing capabilities or replaced by other digital assistants such as mobile phones. Here, sensor information related to the field agents themselves (such as GPS coordinates or orientation in space) is simulated locally.

Use Case Scenario

In our test scenario, a crowd has gathered in some venue to watch a community event. The venue is a static grid of dimensions 700×500 with an exit at position (350, 10). Initially, field agents are stationed arbitrarily in space. They have a detection range of $d_r = 200$ and at each internal step they can cover a maximum distance of $d_{\max} = 20$. Their goal is to detect any incident and contain it by forming a circle to make space for the arrival of first aid. After completing the isolation process, the remaining field agents

form a chain between the incident and the exit to facilitate the evacuation of the casualties. Here an incident happens at location (400, 450), requiring eight agents to encircle it. Once the structure has stabilised, another incident is added at (250, 150) that agents must address.

In this context, the ruleset (Algorithm 2) was designed so that the agents’ connectivity behaviour ultimately solves the above challenge (Fig. 3). To this goal, they are endowed with three pairs of ports: (X, X') , (Y, Y') and (C, C') , alongside two high-level sensors: S_c , activated when a casualty is inside the agent’s detection range, and S_x , triggered by an exit. Both X and C ports are opened first so that idle agents can contribute to a circle (connected via X) or a chain (via C). When an agent is part of a growing circle, both X and X' are open, enabling concurrent circles to merge together and new idle agents to be added in no particular order. When the X -chain reaches a sufficient size, appropriate nodes connect via Y to complete the circle (lines 13-16). Furthermore, the (X, X') ports are tagged with the value of S_c , i.e. the casualty’s ID, restricting possible connections to peers near the same casualty. If too many agents are around the casualty, excess agents disconnect themselves from the structure and move away (lines 17-19). After the circle containment process is complete, and if there are other agents remaining, the C' port is opened to trigger the growth of a chain in the direction of an exit (Fig. 4).

Results

First, we implemented this scenario in eight RPIs (Fig. 3a,b) and a central laptop (Fig. 3a,c) to observe the model’s behaviour in a more concrete context. Since the small number of agents was just enough to encircle the first casualty, the scenario was stopped at this point (no exit chain was formed). Each RPI independently executed one SA, while the computer held the EOC and list of EAs (here, one casualty and one exit). Figure 3b shows screenshots of the subjective views from three of the eight devices. This visualisation interface is composed of the local environment and sub-graph perceived by the corresponding field agent (e.g. landmarks and connected peers) along with personal information, such as its connectivity state. The top bar also displays the direction and distance to the next calculated position.

Through this first trial, we wanted to evaluate the collective ability of agents to consistently develop a robust shape over multiple runs, solely based on their local actions. SAs were not assigned predetermined locations neither in 2D nor in the graph; instead, they were given the same program (including Algorithm 2 at the core of the P routine) and were driven via these rules and their interactions toward forming a specific and stable (albeit temporary) network topology. Moreover, although the order of link requests, sensor activations and, ultimately, graph nodes was not deterministic and produced significant variations at each new simulation, the final overall circle structure was reliably the same.

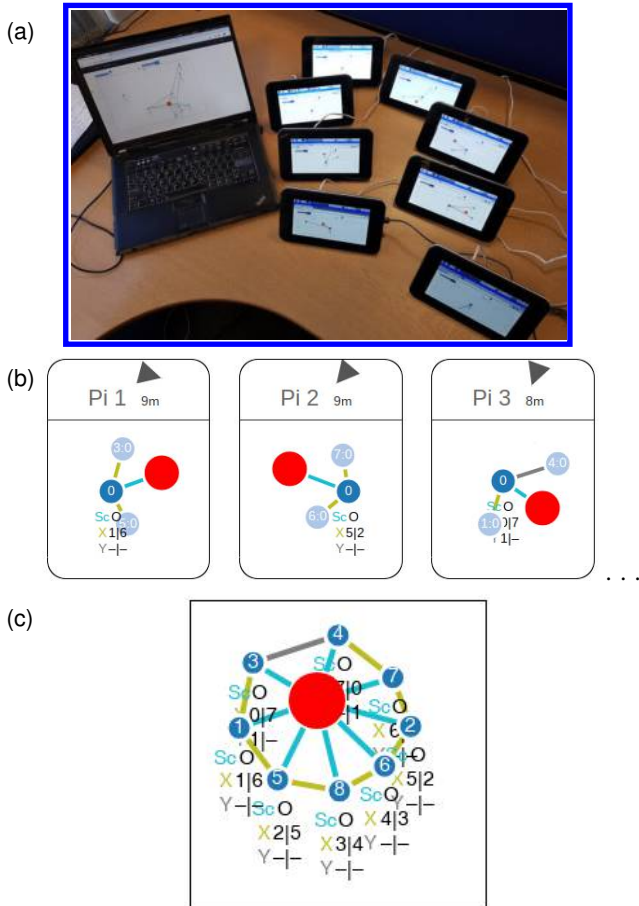


Figure 3: First experiment with eight RPi's. Agents (in dark blue) discover an incident (in red) and encircle it. Links represent detection via sensors (in cyan) or interactions with other agents (in green). (a) Physical setup: eight RPi's communicate with each other and a laptop. (b) Three of the agents' subjective views, showing the local connectivity and landmark, as displayed on each device. Directions to reach the next calculated position are featured in the top bar. (c) Global view on the central computer. Both types of views also indicate the state of ports and gradient values.

In a second experiment, 15 SAs were simulated on a single machine (Fig. 4). While this setup is clearly not equivalent to the constraints of running on multiple asynchronous devices, it nevertheless resulted in the same type of circle and chain structures. As for the precedent evaluation, the resulting global structures were consistent throughout the runs. Figures 4c,e are representative of these graphs, while intermediate steps differ according to the specific agents' interactions and locations. Based on this, we could make a few interesting observations. First, the ability to collectively react and grow appropriate graphs is triggered by the detection of landmarks. Rules are designed to prevent matching pairs of ports from connecting when agents are roaming freely in the initial state (lines 1, 2). This situation is unlocked whenever a casualty sensor is activated (lines 8-16). Here, when

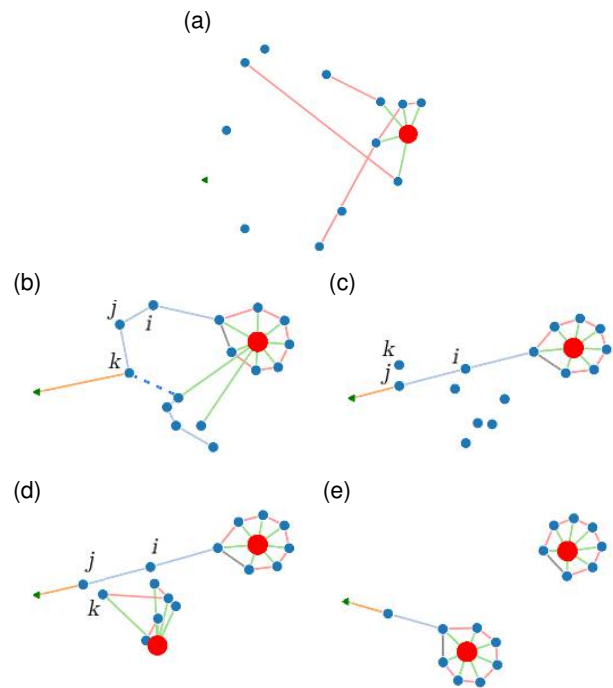


Figure 4: Timeline of a simulation with 15 agents on one machine. (a) Agents are randomly scattered in 2D space (not all visible here), then an incident happens (in red): the nearest agents detect it (green links) and start sending requests to peers to contain it (red links). (b) The incident is encircled, then a meandering chain (blue links) has formed past the exit (green triangle) and breaks up (dashed link). (c) Part of the chain straightens while other nodes detach from it. (d) A second event requisitions the idle agents. (e) End of the simulation: the two incidents are contained.

an incident occurs (red disc), five field agents detect it (green links) and open their X' port (Fig. 4a). This port acts as a recruiter attracting more field agents toward the point of interest. Once sufficiently near, agents behave similarly and, if enough are present, the circle is closed (Fig. 4b). Here, two extra agents are still within detection range of the event (longer green links) but are moving away.

Then, the relative influence of internal gradient values with respect to external events (sensor values) is reflected in the ruleset. Here the circle formation results purely from the former, whereas its particular location is guided by the latter. The additional chain structure also stems from the port logic, whereas its exact length varies with the distance between landmarks. After encirclement, the other agents form a long 7-node chain (Fig. 4b) until node k comes within detection range of the exit (green arrow). This causes k to disconnect from its successor (dashed link) and break up the chain. The loose end of the chain disassembles, while forces among the remaining consecutive nodes make the chain straighten (Fig. 4c). This causes j to draw near to the exit and connect directly to it, which in turn releases k and allows it to participate in other events (Fig. 4d,e).

Finally, priorities among specific parts of the structure can be hard-coded in the rules. For example, a node belonging to a circle could close its C port (line 10) to avoid being enrolled in a chain. On the contrary, a node belonging to a chain keeps its X port open so that it stays available for a circle. This type of rule asymmetry results in promoting circles before chains to represent the fact that containment is considered more important than evacuation in this scenario. In fact, after forming the first chain, node i was poached and recruited into a new circle around the second casualty, while j remained chained to the exit and i (Fig. 4e).

Conclusion and Future Work

In this paper we showed how agents could self-organise and form consistent structures according to their environment without top-down directions from a central control. The agents adapted their behaviour as a result of local or distant events, without the need to assess their own usefulness. For instance, if a global sight of the whole circle is blocked by smoke or because it is too big, appropriate assignments and positions are automatically computed by the model.

We presented an example where a first responder team deployed in a simplified representation of some venue provides protection to possible casualties and facilitates emergency evacuation. A reduced scenario with a small number of RPIs was performed to verify the model's behaviour in a context closer to reality. We also simulated an extended version to highlight interesting properties and the ability of the model to accommodate a second casualty without the need to overwrite the rules. Furthermore, as coordination is handled by the port and gradient logic, the field agents are better able to focus on concrete value-added actions and less preoccupied about coordination.

For future work, several directions can be envisioned. The model could be integrated into a crowd simulator to analyse its behaviour in a more complex environment. The use of real floor plans instead of a square lattice to represent the environment would also help reduce the gap between reality and simulation. Field agent roles could be diversified (e.g. police, paramedic) to further regulate interactions based on expertise, while different types of sensors (e.g. toxic gas detector) could be carried only by certain agents. Link requests could also be limited to the nearest neighbours, instead of broadcast to all, to reduce the need for field agents to move in order to accommodate the forces resulting from the structure.

Acknowledgements

NT's thesis work, supervised by RD, is supported by the School of Computing, Mathematics and Digital Technology at Manchester Metropolitan University. NT wishes to thank PhD fellow Zoe Bartlett for her help reviewing and correcting the English, and co-supervisors EN and Bruce Edmonds for continued advice throughout his work.

References

- Bartoli, G. et al. (2015). A novel emergency management platform for smart public safety. *Int. J. Commun. Syst.*, 28(5):928–943.
- Chen, R. et al. (2008). Coordination in emergency response management. *Commun. ACM*, 51(5):66.
- Ciabattoni, L. et al. (2017). Real time indoor localization integrating a model based pedestrian dead reckoning on smartphone and BLE beacons. *J. Amb. Intel. Hum. Comp.*, pp1–12.
- Conrado, S. P. et al. (2016). Managing social media uncertainty to support the decision making process during emergencies. *J. Decis. Sys.*, 25(sup1):171–181.
- De Wolf, T. and Holvoet, T. (2006). Design patterns for decentralised coordination in self-organising emergent systems. In *Int. Workshop Engineering Self-Organ. Applications*, pp. 28–49. Springer.
- Doursat, R. (2008). Programmable architectures that are complex and self-organized-from morphogenesis to engineering. In *ALife 2008*, pp181–188.
- Doursat, R. and Sánchez, C. (2014). Growing fine-grained multicellular robots. *Soft Robotics*, 1(2):110–121.
- Fraseri, M. et al. (2018). Adaptive autonomy in a search and rescue scenario. In *SASO 2018*, pp150–155. IEEE.
- O'Flynn, B. et al. (2018). First responders occupancy, activity and vital signs monitoring. *Int. J. Adv. Net. Services*.
- Palmieri, F. et al. (2016). A cloud-based architecture for emergency management and first responders localization in smart city environments. *Comput. Electr. Eng.*, 56:810–830.
- Sadhu, V. et al. (2016). Argus: Smartphone-enabled human cooperation via multi-agent reinforcement learning for disaster situational awareness. In *2016 IEEE Int. Conf. Auton. Comp.*, pp251–256. IEEE.
- Scheurer, S. et al. (2017). Sensor and feature selection for an emergency first responders activity recognition system. In *2017 IEEE SENSORS*, pp1–3. IEEE.
- Serugendo, G. D. M. (2009). Robustness and dependability of self-organizing systems-a safety engineering perspective. In *Symp. Self-Stab. Sys.*, pp254–268.
- Ulieru, M. and Doursat, R. (2011). Emergent engineering: a radical paradigm shift. *Int. J. Autono. Adap. Commun. Sys.*, 4(1):39.
- Weidinger, J. et al. (2018). Is the frontier shifting into the right direction? a qualitative analysis of acceptance factors for novel firefighter information technologies. *Inform. Syst. Front.*, 20(4):669–692.
- Yang, L. et al. (2013). How the internet of things technology enhances emergency response operations. *Technol. Forecast. Soc.*, 80(9):1854–1867.
- Yuan, Y. and Detlor, B. (2005). Intelligent mobile crisis response systems. *Communications of the ACM*, 48(2):95–98.

Towards Complex Artificial Life

Lance R. Williams¹

¹Department of Computer Science, University of New Mexico, Albuquerque, NM 87131
williams@cs.unm.edu

Abstract

An object-oriented combinator chemistry was used to construct an artificial organism with a system architecture possessing characteristics necessary for organisms to evolve into more complex forms. This architecture supports *modularity* by providing a mechanism for the construction of executable modules called *methods* that can be duplicated and specialized to increase complexity. At the same time, its support for *concurrency* provides the flexibility in execution order necessary for redundancy, degeneracy and parallelism to mitigate increased replication costs. The organism is a moving, self-replicating, spatially distributed assembly of elemental combinators called a *roving pile*. The pile hosts an asynchronous message passing computation implemented by parallel sub-processes encoded by genes distributed throughout the pile like the plasmids of a bacterial cell.

Introduction

Since its beginning, the field of artificial life has been concerned with the twin problems of the origin of life on Earth and its evolution into forms of increasing complexity. Because these problems are among the most important in science, the idea that experiments with artificial chemistries, organisms, and ecologies hosted on computers might substitute for direct observation of events from the lost history of the early Earth remains extremely seductive. Still, progress has been slower than many might have expected, and artificial life's (arguably) most compelling demonstrations are already several decades old. It follows that a new approach is needed. In this paper we describe an artificial organism constructed using an object-oriented combinator chemistry. While more complex than any previously described, it demonstrably possesses a system architecture compatible with its evolution into still more complex forms.

Phylogenetic reconstructions indicate that all life on Earth descends from a last universal common ancestor (LUCA) that existed as early as 3.8 billion years ago (Glansdorff et al., 2008). This organism was probably a chemical autotroph living near a geothermal vent. Notwithstanding its likely inability to synthesize amino acids, it was already quite complex, containing an estimated 355 genes. Significantly, like all of its descendants, it possessed the molecular

machinery needed to transcribe DNA into RNA, and translate RNA into proteins. Fossil stromatolites show that by 3.7 billion years ago, the tree of life rooted at LUCA had branched many times, yielding a diversity of more complex organisms occupying a range of niches in complex ecologies (Nutman et al., 2016).

Although the mystery of its origin is paramount among the open questions in our field, the question of how an organism of LUCA's non-negligible complexity evolved into a diversity of still more complex forms may be more immediately amenable to investigation using the artificial life approach. In software engineering terms, did LUCA possess a system architecture that facilitated its further evolution? If so, what were the essential characteristics of this architecture? Could an artificial organism with an architecture possessing these same characteristics be designed? Would an artificial organism so designed placed in an artificial world where it was forced to compete with other organisms of the same kind for resources evolve into a diverse ecology of still more complex organisms, given enough time? We believe that the answers to the first, third and fourth questions are all 'yes' and these beliefs motivate the present work. As for the characteristics of LUCA's system architecture that allowed it to evolve into more complex forms, two of the most likely are discussed in the section that follows.

Accumulation of Complexity

It has been proposed that a sustained increase in complexity of the most complex entities of an evolving population is a *hallmark* of open-ended evolution (Taylor et al., 2016). Although this idea seems very compelling, it begs the question of how complexity is defined. In this section, we assume a specific definition for complexity and describe two classes of *mechanisms* that together explain its accumulation in ancient lineages—the first are the source of its increases; the second mitigate its cost.

The *Kolmogorov complexity* of a string is defined as the length of the shortest program that prints it. Unfortunately, Kolmogorov complexity's value as a measure of the complexity of artificial organisms is limited because random strings require longer programs than non-random strings. A

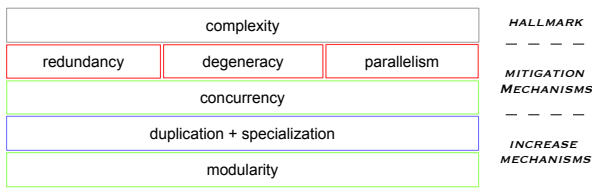


Figure 1: *Modularity* facilitates increases in complexity by allowing *duplication* and *specialization* of *modules*. *Processes* are executable modules that *concurrency* allows to be executed in different orders. In a concurrent system, duplication of processes can increase *redundancy*, while duplication followed by specialization can increase *degeneracy*. These mitigate the cost of increased complexity by increasing *robustness*. *Parallelism* mitigates the cost of increased complexity by decreasing the time an artificial organism needs to reproduce.

measure that discounts randomness is required. The *logical depth* of a string is the *time* required to print it given its shortest representation (Bennett, 1988). Because random strings are incompressible, they are their own shortest representations, and have low logical depth.

Now consider a string that is a compressed representation of a decompression program. When the program is applied to the string, it prints *itself*. It follows that the program plus string system is a *quine* with logical depth equal to its replication time. The implication is profound—if complexity is equated with logical depth, then (absent parallelism) complex organisms require more time than simpler organisms to reproduce. It follows that complex organisms are at a disadvantage relative to simpler organisms in zero sum competitions for resources.

Because this is a bold assertion, it's worth noting that in the natural world, complex organisms are not intrinsically better at staying alive either. Indeed, the theory of *constructive neutral evolution* posits that only the variance in complexity of organisms has increased over time; its modal value has not (Carroll, 2001). There are innumerable more simple organisms than complex organisms (no matter how you count) and organisms as complex as ourselves merely occupy the tail of a very broad distribution.¹

Increases in complexity in individual lineages are introduced by evolutionary “ratchets,” devices which increase complexity in ways that cannot be reversed (Luke et al., 2011). Although there are others, the most important ratchets are *duplication* and *specialization*. By means of these devices, complexity accumulates in lineages over time irrespective of whether or not it confers an adaptive advantage (see Figure 1). Sometimes it does; more often it doesn't.

According to this theory, complex organisms exist primar-

¹“That which does not kill us makes us stranger.” — Trevor Goodchild, *Aeon Flux*.

ily due to the fact that life on Earth is ancient. Generally speaking, they do not survive by virtue of their complexity; they survive despite it. For this reason, we believe that an artificial organism capable of open-ended evolution must possess a system architecture in which both complexity increasing ratchets and factors mitigating the costs of complexity increases can be formulated. The essential characteristics of the system architecture are *modularity* and *concurrency*.

Modularity exists at many levels in the biochemical apparatus of the cell. Protein structural domains, individual proteins, protein complexes and protein interaction networks have all been described as “modules” (Pereira-Leal et al., 2006). Significantly, there are examples of increased biological complexity originating from the duplication and specialization of modules at each of these levels.

If modularity provides the *modules* that are duplicated and specialized to increase complexity, then concurrency allows the modules to be composed in ways that mitigate the costs of those increases. Executable modules are *processes* and concurrency is the property of a system that allows processes to be executed in different orders without affecting the result. More precisely, concurrency allows processes to be executed in *partial orders* defined solely by data dependencies. This flexibility increases robustness.² While the connection between modularity and evolvability has often been emphasized, the importance of concurrency to an evolvable system architecture has not been previously noted. This is probably because concurrent execution is the default for biochemical systems. However, this is not true of computational systems. Indeed, to our knowledge, there is no artificial organism apart from our own (see Figure 2) that replicates using operations that can be performed in different orders.

A system is *redundant* if it contains multiple instances of the same component and if working instances can substitute for broken instances in the event of failure. Duplication creates multiple process instances and concurrency allows one instance to execute instead of another, yielding redundancy.

A system is *degenerate* if it can solve the same problem in different ways (Edelman and Gally, 2001). Concurrency supports degeneracy because it allows a process derived by duplication and specialization of an antecedent process to execute instead of the antecedent. Redundancy and degeneracy increase robustness because they allow organisms to survive component failure and respond in a variety of ways to complex environments.

Parallelism is the simultaneous execution of processes on multiple processors. Absent a global clock, parallelism is impossible without concurrency; absent parallelism, complex organisms are at a disadvantage relative to simpler organisms in the competition for resources, since they require more time to reproduce.

²‘Robustness’ in the engineering sense, not in the sense it is used in evolutionary biology, where it is generally understood to mean stability of the genotype-to-phenotype mapping.

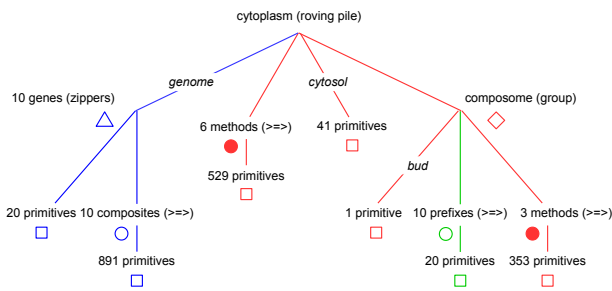


Figure 2: The *artificial protocell* is a moving, self-replicating, spatially distributed assembly of 1855 primitive combinators called a *roving pile*. Its genome consists of 10 genes represented by *zippers* that are distributed throughout the pile like the *plasmids* of bacterial cells. Methods in the *cytoplasm* are executed in parallel and in parallel with those in the *composome*, which execute concurrently.

Autocatalytic Set

Combinators which return monadic values are the building blocks of programs in functional programming. They differ from other notional program building blocks, *e.g.*, *byte-codes*, in that monadic combinators do not require additional address operands to implement computations which would require statement-level control in imperative programming, *e.g.*, iteration. Like polypeptides in biochemistry, programs exhibiting complex behavior can be constructed from combinators simply by sequencing them.

Object-oriented combinator chemistry (OCC) is an artificial chemistry with composition devices borrowed from object-oriented and functional programming languages (Williams, 2016). *Actors* are embedded in space and subject to diffusion; since they are neither created nor destroyed, their mass is conserved. Actors can associate with one another by means of *groups* and *bonds*. This allows working sets to be constructed and the actors in these working sets to be addressed in different ways. Actors use programs called *methods*, constructed from combinators, to asynchronously update their own states and the states of other actors in their neighborhoods. The fact that programs and combinators are themselves reified as actors makes it possible to build programs that build programs from combinators of a few primitive types using asynchronous spatial processes that resemble chemistry as much as computation.

A composite combinator can be represented as a binary tree with primitive combinators as leaves and interior vertices signifying Kleisli composition (\gg). In OCC, the *compose* primitive combinator joins two trees with (\gg) while the *decompose* primitive combinator splits a non-leaf tree into its two subtrees. Composite combinators can be promoted to executable methods using the *unquote* primitive combinator.

A *zipper* is an implementation of a data structure that al-

lows it to be traversed and updated without mutation (Huet, 1997). All zippers consist of three parts. The *front* represents the portion of the data structure that has already been traversed, the *back* represents the portion yet to be traversed, and the *focus* is a data item between the front and the back that can be examined or replaced.

A composite combinator's simplest assembly sequence builds it by adding one primitive combinator at a time via Kleisli composition, *i.e.*, it is a *right fold* with (\gg). This produces a lopsided tree that can be implemented as a *list zipper*. Both the back and the front of the zipper are composite combinators with the primitives comprising the front composed in reverse order. The zipper's focus is a single primitive combinator.

In a reified implementation in OCC, a *next* bond joins the back and front while a *hand* bond joins the back to the focus. The zipper is traversed by pushing the focus onto the front (using *compose*), and popping a primitive combinator from the back (using *decompose*). This primitive combinator becomes the new focus.

A reversed copy of a composite combinator can be constructed by traversing its zipper representation. This is accomplished by replacing the front with a *pair of fronts*. These are connected to the back with *prev* and *next* bonds. At each step of the traversal, the focus is pushed onto the first front and a primitive combinator from the neighborhood with type matching the focus is pushed onto the second front. This process is repeated until the back consists of a single primitive combinator, at which point the pair of fronts represents a reversed original and a reversed copy. These can (in turn) be reversed (producing a non-reversed original and non-reversed copy) by a second traversal of the zipper in the opposite direction. This requires creation of a *pair of backs*. The second back (initially a primitive combinator from the neighborhood with matching type) is joined to the first back by making it a member of the first back's *group*.³

Note that all of this is accomplished using a very small number of operations that push (and pop) primitive combinators and make (and break) bonds. Significantly, by using zippers, we eliminate the need for pointers to characters within arbitrarily long string representations of programs, *e.g.*, as in Hickinbotham et al. (2011).

The copying process is implemented by six methods; see Figure 3 (left). Initially, the front and focus of a zipper representing a composite combinator to be copied are both primitive combinators (the first two forming the composite). The following operations are performed sequentially:

- *AcsA* creates the second front by finding a primitive combinator matching the front in the neighborhood and creating the *next* bond.

³A group is used instead of a *hand* bond so that the form of the input to method *acsE* is distinguishable from the form of the input to method *acsC*.

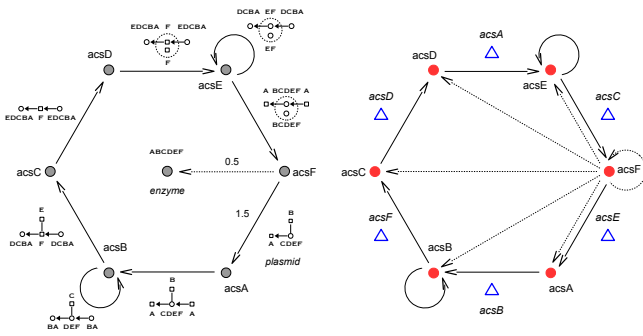


Figure 3: Six stage process used to synthesize methods and copy zippers showing changes to zipper conformation and function of each method in the parallel pipeline (left). The addition of six zippers representing the methods implementing the process itself yields an autocatalytic set (right).

- AcsB traverses the zipper in the forward direction, extending the pair of fronts representing the reversed original and reversed copy. At each step, the copy is extended using a primitive combinator of matching type found in the neighborhood.
- AcsC adds the final pair of primitive combinators to the pair of fronts leaving the zipper without a focus and with a back consisting of a single primitive combinator.
- AcsD creates the second back by finding a primitive combinator in the neighborhood matching the first back and joining it to the first back's group.
- AcsE traverses the zipper in the backward direction, reversing the pair of fronts by popping primitive combinators off both of them and pushing these primitive combinators onto the pair of backs.
- AcsF finishes the reversing process and uses the first back to construct a zipper in the initial state. AcsF then (in effect) flips a coin. If the result is *heads*, acsF unquotes the second back, promoting it to a method. If the result is *tails*, acsF uses the second back to a construct a copy of the original zipper in the initial state.

If the six copying methods acsA–acsF are placed in the world with a zipper representing a seventh method, then half of the time, the zipper representing the seventh method will be copied. The other half of the time, an instance of the seventh method will be synthesized. Once the world contains multiple copies of the zipper representing the seventh method, the six copying methods will begin to execute in parallel, forming a production pipeline for inert (zipper) and active (method) instances of the seventh method.

At this point, an interesting possibility suggests itself. If the six copying methods acsA–acsF are placed in the world with six zippers representing the copying methods *themselves*, then the twelve entities will form an *autocatalytic*

set (Farmer et al., 1986). Over time, the six methods will use the six zippers to construct additional copies of both methods and zippers. The methods and zippers are the spatially distributed components of a modular, concurrent, parallel, self-replicating system; see Figure 3 (right). However, despite these noteworthy attributes, the autocatalytic set is not a *bona fide* artificial organism because it does not segregate its components from the components of other systems, and absent this *compartmentalization*, Darwinian evolution is impossible.

Membranes

After elemental building blocks, reaction catalysts, and molecules for storing energy and information, compartments are probably the next most important ingredient in the recipe for life. Given their amazing utility, it is remarkable that, in our universe, we basically get them for free. This is due to the existence of lipid compounds that, when placed in water, spontaneously assemble into *liposomes*, vessels defined by bilayer membranes. Yet membranes are not uncomplicated. Consider the problem of how to make one grow. To insert a molecule into a lipid bilayer, a set of forces must be applied on the lipid molecules adjacent to the point of insertion to create a gap and these forces must propagate through the bilayer. They must be combined with the attractive forces the lipids exert on each other and the forces exerted on the membrane by the cytoplasm. This *mass spring system* requires a physics far more complex than the rudimentary one underpinning OCCC, which has no analog of force.

However, there is a still harder problem associated with growth. In order for a cell to grow, two different actions must be coordinated. First, the volume must increase. This can be done by adding something to the cytoplasm. Yet if pressure is to remain constant, the membrane must also increase in area. Complicating matters, the cytoplasm's volume and the membrane's area must increase at different rates. Assuming a spherical cell, an increase in the volume by ΔV requires a corresponding increase in surface area by

$$\Delta A = \pi(r^3 + \Delta V)^{\frac{2}{3}} - \pi r^2$$

which depends on the cell's radius, r . Given the dependence on r , it follows that there is no single local operation that can maintain constant pressure by pairing imports to both cytoplasm and membrane.

Fortunately, membranes are not the only way to achieve the compartmentalization necessary for the creation of life. In fact, in the physical universe, a thing as simple as a water droplet in oil can function as a compartment.⁴ In a computational universe, a compartment is simply a data structure for representing a compact, spatially embedded set. Using a Jordan curve to represent membership in such a set

⁴Sokolova et al. (2013) have demonstrated transcription and translation in *E. coli* lysate contained in water-in-oil droplets.

by partitioning space into two disjoint regions, one (*inside*) containing the set's elements, the other (*outside*) containing everything else, is merely one possibility.

Roving Piles

North, *east*, *south* and *west* are new relations in OOC in multisets of actors, or *groups*. We will call the edges of group relations, *links*, to distinguish them from the edges of actor relations, which we call *bonds*. As with actors and bonds, groups can possess at most one link of each type. *East* and *west* are inverse relations, *i.e.* $E(x,y) = W(y,x)$; the same is true of *north* and *south*. Because they correspond to the four cardinal compass directions, links of these four types are called *cardinal links*. Cardinal links are used to connect *base groups*. *Up* and *down* are a second inverse relation on groups that can be used to represent a stack of additional groups above any base group. A base group is a group without a *down* link; a base group without an *up* link is said to be *uncovered*. A *roving pile* is a connected component of base groups embedded in the 2D lattice together with the groups contained in stacks above them. The set of base groups form the pile's *footprint* and base groups with one or more empty cardinal links form its *boundary*.

In OOC, methods in the same stack execute concurrently but not in parallel; they compete for a shared processor resource in zero sum fashion. However, methods in different stacks in the same pile execute in parallel. So that piles can move and grow, and so that actors within piles can freely mix, groups in piles are subject to the following three operations:

1. *Diffusion*. A non-base group can be moved to an adjacent stack.
2. *Retreat*. An uncovered base group on the boundary can be moved to an adjacent stack if its removal from the footprint will not split the footprint into separate connected components.
3. *Advance*. A covered base group on the boundary can be replaced in the footprint by the group above it and used to extend the footprint in the direction of an empty cardinal link.

Ideally, these operations would be implemented as described above and performed at random. Unfortunately, the retreat and advance operations, as described, cannot be implemented using only local rules.

Determining whether or not the removal of a group from the footprint will split the footprint into separate connected components is inconsistent with an implementation on an ACA substrate since it is a function of non-local properties of the cardinal link relation. For example, the footprint might consist of base groups forming a square with sides one group wide and n groups long; see Figure 4 (left). Although it is clear that any single group can be removed without splitting the footprint, this can only be determined by

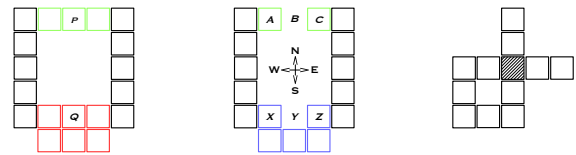


Figure 4: Even though it would not split the pile's footprint, an uncovered base group at P cannot join the stack to its east because this cannot be determined by local analysis alone (left). In contrast, an uncovered base group at Q can do so because it would not split the subset of the footprint within its Moore neighborhood (red). Although a covered base group at A can advance the footprint east (and C is in the footprint) no link to C will be created (middle). In contrast, because there is a path between X and Z in the subset of the footprint contained in the Moore neighborhood of Y (blue), a covered base group at X can advance the footprint east and create a link to the base group at Z. Because the evolution of roving pile shape is governed solely by local rules, pile footprints can overlap (right). However, actors in overlapping neighborhoods cannot interact.

traversing a path of length $4n - 1$ links. For this reason, the implementation of the *retreat* operation in OOC is based on a stronger (sufficient but not necessary) property. More specifically, an uncovered base group can be removed if and only if it will not split the subset of the footprint contained in its Moore neighborhood into separate components. This stronger property can be enforced using only local rules.

Implementation of the *advance* operation presents a similar problem. To understand this, consider a roving pile with a square footprint like the one described above, but with a single group removed; see Figure 4 (middle). In principle, an advance operation could fill the gap, completing the square. However, this would require a process able to determine whether or not base groups adjacent to the advance site are part of the same pile as itself. Again, this can only be done by traversing a path of length $4n - 1$ links. The solution is to perform an exhaustive enumeration within the neighborhood surrounding the advance site; see Figure 5. This is done to avoid (as much as possible using local rules only), the situation where spatially adjacent regions of the footprint are not connected.

Observations of a working implementation show that roving piles remain flat (low average stack height) and connected. Smaller piles (those containing less than fifty actors) constantly evolve in shape while rapidly moving around the lattice on random walks. Holes created by expelling actors in uncovered base groups are quickly filled. Larger piles extend and retract pseudopod-like extensions but remain largely immobile in aggregate.

Four primitive combinators were added to OOC to serve as an interface to the roving pile data structure:

- *Safe* fails if the actor it is applied to cannot be removed

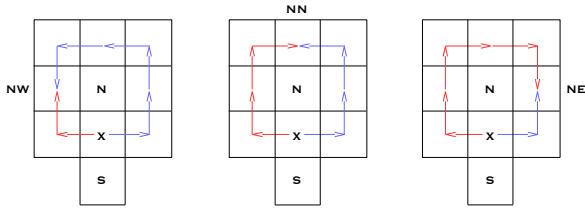


Figure 5: A covered base group at X with an empty north link can be replaced in the footprint by the group above it in the stack. Now unlinked, this *advance group* can be used to extend the footprint northward to N. This requires mapping the footprint in the neighborhood of N using search. Any base groups discovered at NW (left), NN (middle) and NE (right) become the advance group’s *west*, *north* and *east* links. The group that replaced it in the footprint at X becomes the advance group’s *south* link. Corresponding advance operations are performed in the other three cardinal directions. The four together depend only on the topology of the footprint inside a 5×5 neighborhood centered on X.

from the pile without splitting the pile’s footprint. It is used as a *guard* for actions that change actors’ positions in the pile or expel actors from the pile.

- *Expel* removes an actor from the pile. The actor becomes invisible to actors inside the pile and visible to actors outside the pile. This action fails if the actor cannot be expelled without splitting the pile’s footprint.
- *Request* creates a *proxy group* representing a request for the pile to import an actor of the same type as the actor it is applied to. It fails if it is applied to an actor which is not a primitive combinator.
- *Seed* creates a new pile containing a single group.

Artificial Protocell

Recent work has described liquid droplets containing enzymes catalyzing growth that spontaneously fission into two equal sized droplets upon reaching a critical size (Zwicker et al., 2017). The authors (and others) suggest that droplets like these could form the basis of an artificial protocell in *vitro*. The possibility of designing a roving pile with analogous behavior that could form the basis of an artificial protocell in *silico* leads us to ask whether an autocatalytic set comprised of method and zipper instances of *acsA–acsF* could be hosted in a roving pile. A *viable* protocell hosted in a roving pile would contain both the autocatalytic set and the primitive combinators needed to synthesize it. These primitives would be consumed during the process of copying methods and zippers, but be replenished by pairing *compose* actions that consume primitives with *request* actions that replace them while also yielding geometric growth. This growth would culminate in binary fission. Assuming that the components of the mother protocell are divided among

its two daughters at random, then the probability that both daughters will be viable becomes closer and closer to one as the mother’s size increases.

The approach sketched above seems like a simple and elegant pathway to an artificial organism possessing modularity, concurrency and parallelism. Unfortunately, there are several practical difficulties. First, the phenomenon of droplet fission is based on the fact that instability increases as droplet size increases. Because an analogous mechanism devised for roving piles would require the computation of the non-local property of pile size, there can be no *simple* mechanism for pile fission. However, even if a mechanism could be devised, the pile size of the mother protocell required to reasonably guarantee the viability of both daughters would still be quite large (in the tens of thousands). For both of these reasons, a different solution was sought.

Absent splitting a mother into two equal-sized daughters, a daughter must be constructed, method-by-method and zipper-by-zipper, in a process more like *budding* than fission. An efficient construction process would export, to the daughter, one method and zipper instance of each gene, and the primitive combinators necessary to synthesize both. To keep track of what has already been exported, and to recognize when the daughter has received the full complement of components, the mother protocell needs to maintain a checklist of some kind. We call the group of actors comprising and managing this checklist, the *composome*, since it serves as the protocell’s repository of *compositional information*.

The simplest composome would consist of the methods implementing the export and budding processes, and a set of composites (one per gene) to represent the checklist. The copying process in the cytoplasm would translate zippers into composites, and each of these would be exported to the daughter as a composite, method or zipper; composites exported as composites would be used to construct the daughter’s composome. Composites in the mother’s composome would be marked with self-bonds during the export process to indicate which composites, methods and zippers have been exported and which have not. After the full complement has been exported, the bond between mother and daughter (now viable) would be severed.

Although the approach sketched above works, it has shortcomings. First, it is clearly inefficient to use composites to represent methods and zippers since each has the same length as the method and zipper it represents. Second, requiring two *identical* copies of each gene (a zipper in the cytoplasm and a composite in the composome) would undermine evolvability, since a point mutation in either copy would render the protocell non-viable. Recognition of these shortcomings lead to a better approach, described below.

If the composites constructed in the cytoplasm possessed short, unique, non-executable *prefixes*, and these prefixes could be used to form the checklist in the composome, then the protocell would be far more efficient. Since there is only

one copy of each gene (a zipper in the cytoplasm), evolvability is not undermined; see Figure 6. This design, for an artificial organism with an architecture featuring modularity, concurrency and parallelism, has been implemented and tested in OOCC. It efficiently and reliably replicates across multiple generations and possesses only 10 genes:

- CopA–copE perform operations that are identical to acsA–acsE except for three small differences. First, copA–copE are all prefaced by a *quit* combinator that is executed when the method is exported to the daughter composome. This causes the method to quit the composome and join the daughter cytoplasm. Second, copA–copE are modified so that the fronts, focii, and backs of all zippers are contained inside single groups. This avoids the tangling that results when the separate parts of a spatially extended zipper joined by bonds occupy different branches of a pile. Third, all actions that consume primitives in the pile are balanced by requests to replace them.
- CopF does the final compose operation needed to complete a composite representation of a gene for export, then restores the zipper to the conformation expected by the copA method.
- CytX contains a short executable sequence, *me* \ggg *quit* \ggg *smash* \ggg *none*, followed by a much longer non-executable sequence containing one of each of the primitives necessary for replication (in no particular order). The short executable sequence causes cytX to quit the daughter composome and *smash* itself so that the primitive combinators comprising cytX itself form the *cytosol* of the daughter.
- ExpX exports composite representations of genes as methods and zippers and marks *prefixes* in the mother composome with self-bonds to keep track of progress. The first two combinators of the composite are removed and composed to form its prefix. If the prefix with matching type in the mother composome has no directed self-bond, then the composite is unquoted and added to the daughter composome together with its prefix.⁵ If the prefix with matching type has a directed self-bond but no undirected self-bond, then the composite and its prefix are used to construct the zipper representation of the gene and this is added directly to the daughter cytoplasm. Finally, if the prefix with matching type has both directed and undirected self-bonds, the composite and its prefix are super-

⁵Because unquoted suffixes are methods, they will execute in the daughter composome when placed there. Cytoplasm-based methods, e.g., copB, are prefaced by a pair of combinators, *me* \ggg *quit*, that causes them to quit the daughter composome; composome-based methods, e.g., expX, lack this device. Like the cytX method used to create the cytosol, this is a simple use of programmed self-assembly by the daughter.

fluous, so they are expelled.⁶

- BudA checks to see if any actor in the composome has a bond. If none do, then it expels a primitive from the mother pile and applies the *seed* combinator to it, creating the daughter pile. It adds a second primitive to the composome and creates a directed bond between it and the first primitive. Finally, requests are made to the pile to replace both primitives.
- BudZ checks to see if all prefixes in the composome have directed self-bonds. If they do, it deletes all prefix self-bonds (directed and non-directed) and also deletes the bond connecting the mother and daughter, which are now both viable protocells.

The artificial protocell is sequential at the top level since it exports methods and zippers one at a time, as they become available, but employs pipeline parallelism in their production. There are only two steps in the pipeline that require more than $O(1)$ time. These are implemented by the copB and copE methods, which require time proportional to the number of primitive combinators comprising the method being copied, $O(M)$. However, the rate limiting step of the replication process is copB, which must wait for the arrival in the neighborhood of primitive combinators imported by the pile. It follows that the parallel time complexity of the replication process is

$$O\left(\frac{MN}{B}\right) = O\left(\frac{M}{B}\right) \sum_{k=0}^{N-1} O\left(\frac{N}{N-k}\right)$$

where M is the average number of primitive combinators per gene, N is the number of genes, and B is the number of instances of copB.⁷ Significantly, the time required for self-replication decreases as additional genes encoding the copB method are added (with diminishing return when $B > N$). It follows that the protocell is a rare example of a self-replicating system where increased complexity, because it yields increased parallelism, pays for itself.

Conclusion

Because it discounts randomness, computational depth is a useful measure of an artificial organism's complexity. Absent parallelism, organisms of increased computational depth require more time to replicate. This means that they are at a disadvantage relative to simpler organisms in zero sum competitions for space. It follows that artificial organisms can only evolve into more complex forms if they divide

⁶Although OOCC doesn't have an *if-then-else*, equivalent functionality can be achieved in a single method when all actions are reverseable. For example, if z' is the action that reverses z , then the sequence, $z \ggg x \ggg z' \ggg y$ will execute the action z when x fails and y when x succeeds.

⁷A breakfast cereal company includes a plastic dinosaur (one of N different types) in each box of cereal. It is straightforward to show that a grandmother must buy $O(N) = \sum_{k=0}^{N-1} O\left(\frac{N}{N-k}\right)$ boxes on average before her grandson has one of each type.

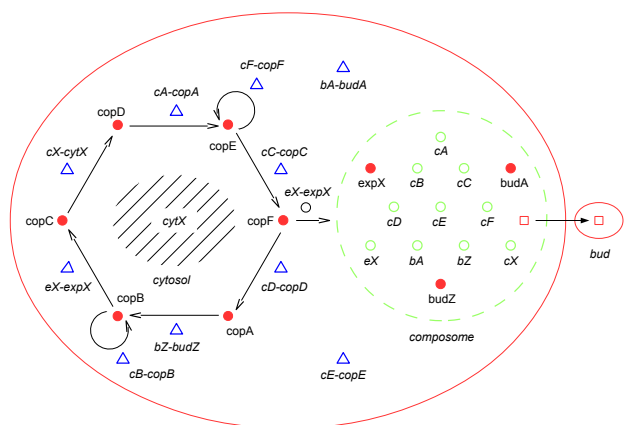


Figure 6: Schematic diagram of artificial protocell showing copying of zippers in cytoplasm and export of zippers and methods by composome through directed bond to daughter protocell. Six methods implement the copying process in the cytoplasm, three methods implement the budding and export process in the composome, and one method smashes itself to form the cytosol. The methods in the cytoplasm execute in parallel and in parallel with those in the composome (which execute concurrently). The composome contains ten *prefixes* (length two composites with unique types) that are marked with self-bonds to signify the zippers and methods that have already been exported.

the problem of self-replication among parallel subprocesses. In the absence of a global clock, parallelism is impossible without concurrency, which allows subprocesses to be executed in different orders.

Artificial organisms can increase in complexity by means of duplication and specialization of modules representing subprocesses. In addition to enabling parallelism, concurrency can mitigate the cost of increased complexity by providing a variety of execution paths, some of which include these duplicated and specialized modules. This can yield increased robustness through redundancy and degeneracy. We believe that modularity and concurrency were already present in the cellular architecture of the last universal common ancestor of all life on Earth and that these characteristics can be credited in part for its subsequent evolution into forms of increased complexity.

Apart from a modular and concurrent architecture, an artificial organism needs a device for separating its genome and replication machinery from those of other organisms. We introduced a new data structure, called a roving pile, capable of representing a set of actors inhabiting an arbitrarily large four-connected component of sites in a 2D lattice. Roving piles move and grow and actors within roving piles freely mix, which is essential for message passing and for the assembly of methods from combinators.

Lastly, we used an object-oriented combinator chemistry

to construct an artificial organism with an architecture featuring modularity, concurrency and parallelism. This organism replicates by means of an asynchronous message passing computation implemented inside of a roving pile containing 1855 primitive combinators. Its genome consists of 10 genes represented by zippers that are distributed throughout the pile like the plasmids of a bacterial cell.

References

- Bennett, C. H. (1988). Logical depth and physical complexity. In *A Half-century Survey on The Universal Turing Machine*, pages 227–257. Oxford University Press.
- Carroll, S. (2001). Chance and necessity: the evolution of morphological complexity and diversity. *Nature*, 409:1102–1109.
- Edelman, G. M. and Gally, J. A. (2001). Degeneracy and complexity in biological systems. *Proceedings of the National Academy of Sciences*, 98(24):13763–13768.
- Farmer, J., Kauffman, S. A., and Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D: Nonlinear Phenomena*, 22(1):50 – 67. Proceedings of the Fifth Annual International Conference.
- Glandsdorff, N., Xu, Y., and Labedan, B. (2008). The last universal common ancestor: emergence, constitution and genetic legacy of an elusive forerunner. *Biology Direct*, 3(1):29.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2011). Molecular microprograms. In *European Conference on Artificial Life (ECAL)*, pages 297–304.
- Huet, G. (1997). Functional pearl: The zipper. *Journal of Functional Programming*, 7(5):549–554.
- Luke, J., Archibald, J. M., Keeling, P. J., Doolittle, W. F., and Gray, M. W. (2011). How a neutral evolutionary ratchet can build cellular complexity. *IUBMB Life*, 63(7):528–537.
- Nutman, A. P., Bennett, V. C., Friend, C. R. L., Kranendonk, M. J. V., and Chivas, A. R. (2016). Rapid emergence of life shown by discovery of 3,700-million-year-old microbial structures. *Nature*, 537:535–538.
- Pereira-Leal, J. B., Levy, E. D., and Teichmann, S. A. (2006). The origins and evolution of functional modules: lessons from protein complexes. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1467):507–517.
- Sokolova, E. et al. (2013). Enhanced transcription rates in membrane-free protocells formed by coacervation of cell lysate. *Proceedings of the National Academy of Sciences*, 110(29):11692–11697.
- Taylor, T. et al. (2016). Open-ended evolution: Perspectives from the OEE workshop in York. *Artificial Life*, 22(3):408–423.
- Williams, L. (2016). Programs as polypeptides. *Artificial Life*, 22(4):451–482.
- Zwicker, D., Seyboldt, R., Weber, C. A., Hyman, A. A., and Jülicher, F. (2017). Growth and division of active droplets provides a model for protocells. *Nature Physics*, 13:408–413.

Environmental Harshness and Fitness Improving Innovations

Peter Andras¹

¹School of Computing and Mathematics, Keele University
p.andras@keele.ac.uk

Abstract

Fitness improving innovations occur in populations of organisms as genetic changes (mutations) that allow better fit with the environmental niche of the organisms. Similarly, fitness improving innovations may occur in the context of human communities as well in terms of socio-economic innovations (e.g. new ways of organizing the military, new products or services) that lead to more efficient use of available resources. Here we explore the link between such innovations and the harshness of the environment, where the populations live. Environmental harshness characterizes the availability of population growth supporting resources in the environment. Our analysis shows that if the harshness of the environment varies smoothly with the distance, the expected extent of fitness improving innovations and of the resource utilization efficiency of populations depends in a combined linear and harmonic manner on the harshness of the environment at the location of origin of the populations. We explore the implications of this result for particular cases of both biological and social environments (e.g. gene drives, business innovation).

Introduction

At any time there is usually a diverse community of organisms at any geographical location (Gould, 2002). These organisms typically belong to more than one, and often many, distinct populations of organisms. The same happens also in terms of humans belonging to culturally distinct populations and companies and organizations belonging to distinct sectors of the economy (Diamond, 1997; Arthur, 2009). Some of these populations are more successful than others. The successful populations spread widely and become dominant, in terms of the number of individuals, among the co-existing populations for a considerable period of time over some extensive geographical range (Andras, 2015; Diamond, 1997; Gould, 2002; Nielsen et al, 2017; Turchin, 2006).

The spreading of populations generally can be described as a reaction-diffusion process (Volpert and Petrovskii, 2009). Generally, individuals belonging to a particular population migrate to geographically neighboring areas and establish the presence of their population there. The meaning of neighboring areas depends on the context, e.g. in the case of animal or plant populations it is a close distance sufficiently hospitable location, but in the context of globally mobile companies may mean almost any location around the Earth with sufficient support services present.

Populations of living organisms change by accumulating mutations in their DNA that lead to the emergence of features

or behaviors at cellular or organismal scale, which give a selective advantage to the individuals harboring these mutations (Maynard Smith and Szathmary, 2000). This process leads to the emergence of new, genetically and reproductively separated, populations of organisms (Yom-Tov and Geffen, 2011). Similar processes happen at the level of socio-economic organization of human communities as well, leading to the emergence of new, culturally distinct, populations and new, technologically distinct, areas of economic activity filled by a corresponding new population of human organizations (Arthur, 2009; Fukuyama, 2011; Turchin, 2006). The actual process of how these changes accumulate to generate the innovations leading to the emergence of new populations is not yet fully known (Maynard Smith and Szathmary, 2000).

Examples of innovations that define new populations include the development of foot adaptations in animals (Holowka and Lieberman, 2018) or the development of drought resistance in plants (Kooyers, 2015). In the case of human populations an example is the development of settled agriculture (Diamond and Bellwood, 2003) or in the case of companies the development of new products or technologies (Arthur, 2009) such as provision of social media services or the development of diesel engines.

A number of environmental factors influence the spreading of populations (Andras, 2015; Barraclough, Vogler and Harvey, 1998; Fukuyama, 2011). Such factors include the presence of geographical barriers (e.g. mountains, sea, rivers), variation in the harshness of the environment (e.g. cold/warm, arid/wet areas, presence/absence of disease spreading vectors), the extent of competition among co-existing populations (e.g. increased number of populations in overlapping environmental niches may lead to more competition among populations), or the speed of growth of co-existing populations (e.g. plentiful resources may facilitate fast growth of all populations). These environmental factors modulate the spreading of populations by altering the diffusion process that drives the population spreading (Andras, 2015).

Previously it has been reported that the average extent of innovations emerging in populations increases with the harshness of the environment where the new population originates (Andras, 2015). In terms of interpretation it has been suggested that the likely reason is that in harsher environments fitness innovations of larger extent are needed to result in sufficient growth of the population to spread to other areas and in such environments larger innovations have amplified effect on the growth success of the populations. On the other hand, in

less harsh areas, even small innovations may provide sufficient advantage for a comparative growth benefit that may lead to the emergence of a successful population (Andras, 2015). However, the reported findings have also shown an unexplained wavy nature of the variation of the mean extent of innovation with the harshness of the environment superimposed on the above noted linear relationship.

Here we explore and explain the nature of this wavy relationship between the harshness of the environment and the average extent of fitness innovations. We also show that the resource utilization efficiency of successful populations also follows a similar, but phase sifted, wavy relationship with the harshness of the location of origin, similarly superimposed on a linear relationship. These results have implications for a number of research and technological applications area that relate to the emergence of innovations in populations, such as the use of gene drive technologies, emergence of antibiotic resistance, and policies for support of business innovation.

The rest of the paper is organized as follows. First we review briefly the relevant background. Next we describe the core equations and theoretical derivations about the distribution of fitness improving innovations and the impact of environmental factors on this. This is followed by a brief summary of the simulation environment that we used. Then we describe our results and the interpretation and discussion of these. Finally, the paper is closed by the conclusions section.

Background

The spreading of populations of organisms and culturally different human populations is usually modeled using reaction-diffusion equations (Volpert and Petrovskii, 2009). This has the following general form

$$\partial q(x,t) / \partial t = \nabla(D(q,x) \cdot \nabla q(x,t)) + F(q,x,t) \quad (1)$$

where $q(x,t)$ is the size or quantity (i.e. normalized number of individuals) of a given population at location x and time t , $D(q,x)$ is the diffusivity of the environment at location x and for population size q , and $F(q,x,t)$ is the reaction terms, which expresses the local dynamics of the population given its size $q(x,t)$ at spatial position x and time t . This last term includes the impact of death and birth of individuals on the population size and also the impact of other factors, such as the competition with other populations, the availability of resources at the location and so on. The impact of the harshness of the environment on populations can be incorporated into this term within the reaction-diffusion model of population spreading. The diffusivity element of the model characterizes the ease / difficulty of spreading from one location to a neighboring location. This element would include for example the presence of natural barriers, such as mountains or rivers in the environment (Andras, 2015).

While the reaction-diffusion model of population spreading is very useful for conceptual and formal analysis of the spreading process it has its limitations. These stem mainly from the limits of formal analysis for cases with non-simple forms of the reaction term and also for cases with complex structure for the diffusivity element. Building models that aim to capture realistic conditions and constraints often lead to

models with such non-simple reaction terms and diffusivity elements (Andras, 2015; Cheng et al, 2014). In such cases agent based models can be used very effectively (Montenegro et al, 2016). In these models a discrete version of equation (1) is used in combination with the modeling of populations by a collection of agents that make a few algorithmic and possibly stochastic decisions (e.g. reproduction, movement, division, etc.). Such agent-based models of population spreading can also include the process of accumulation of innovations (e.g. simulating the impact of genetic mutations or of cultural-technical changes).

The emergence of genetic changes that lead to cellular or organismal level fitness improving innovations has been researched for many years (Maynard Smith and Szathmary, 2000). Although there are still many aspects of this process that are unclear, there is important progress in some respects. The best understood aspects of such processes relate to the emergence of antibacterial resistance in bacteria (Blair et al, 2015) and the evolution of viruses (Vijaykrishna, Mukerji, and Smith, 2015). Recent works also show how the replication of mammalian genes leads to the emergence of fitness improving genetic innovations (Carelli et al, 2016) and the emergence and fitness improving usefulness of mutations in plants have been also analyzed using observations of natural spreading of relatively recently introduced invasive plant species (Exposito-Alonso et al, 2018). In the context of emergence cultural-technical changes that lead to fitness improving innovations among companies or human populations there is also considerable related research (Arthur, 2009; Fukuyama, 2014). Genomic studies of human population spreading can indicate the origins of innovations and the directions of spreading, although do not convey information about how the innovations came about (Nielsen et al, 2017). Both biological evolution and simulation-based research show that environmental barriers, which provide relative isolative protection to evolution of species, increase the frequency of fitness improving innovations, while the lack of such barriers reduces this frequency (Mazancourt et al, 2008; Millien, 2006; Yom-Tov and Geffen, 2011; Andras, 2015).

Previous works on modeling the spreading of human populations and animal species have used agent-based models to analyze the location of origin and the spreading pattern of populations (Andras, 2015; Montenegro et al, 2016). Andras (2015) used agent-based simulation to show that the average resource utilization efficiency and average time persistence of successful populations reduces linearly with the harshness of the location of origin of these populations. Similarly the paper also shows that the average extent of fitness improving innovations of successful populations increases linearly with the harshness of the location of origin. However, in all these cases the paper shows that superimposed on the linear relationship there is also a periodic, wavy relationship with the harshness of the location of origin. This aspect of the relationship has not been fully explained in Andras (2015).

Fitness Improving Innovations and Environmental Factors

We assume that the populations exist in an environment characterized at each location by the environmental harshness,

h . This effectively sets the level of resources available at that location, higher h implying more scarcity of resources. Note that this definition of environmental harshness is very generic and it may incorporate effects of the physical environment, such as aridity, temperature or availability of shelter, and also effects of the live environment, such as predation risk.

Populations are characterized by their resource utilization efficiency, r , which determines to what extent they can use the resources available at a given location. The population growth rate depends on the resource utilization efficiency of the population, on the available resources and also on the general growth support of the available resources and the competition between different co-existing populations for the resources. A relatively general assumption is that the population growth rate, ρ , is proportional with the general growth support, γ , the extent of the competitiveness of the population, η , a gradually saturating function of the resource utilization efficiency of the population (e.g. $\ln(1+r)$) and is inversely proportional with the harshness of the environment, h . The growth rate should also take account the death rate of the population, θ , which is assumed to be the same for all competing populations, but is altered locally additively proportionally with the general growth support, γ , and inversely proportionally with the harshness of the environment, h . The corresponding equation is:

$$\rho = \gamma \cdot \eta \cdot \ln(1+r) / h - \theta + \gamma / h \quad (2)$$

A fitness innovation is defined as a change in the resource utilization efficiency of a part of a pre-existing population, Δr , due to mutations to the genes of the organism or changes to the cultural-technical processes of the human population (Maynard Smith and Szathmary, 2000; Turchin, 2006). A fitness improving innovation is such a fitness innovation that improves the resource utilization efficiency of the population, i.e. when $\Delta r > 0$. For example, this may mean the emergence of enzymes that allow the animals to digest their food more efficiently, or the emergence of molecular changes that allow the build-up of larger and stronger muscle mass, or the emergence of cultural changes that allow better agricultural practices that make possible the feeding of a larger population without requiring extension of the available agricultural land area (Diamond and Bellwood, 2003; Fukuyama, 2014; Maynard Smith and Szathmary, 2000; Turchin, 2006).

The general growth support of the environment, γ , is a general environmental factor that modulates the effectiveness of the impact of the environmental harshness on the populations. For example, large scale climate effects may alter in such way the impact of environmental harshness, or the general accessibility of a technology for all human populations may change the effectiveness of the environmental harshness.

The competitiveness of a population depends on the relative resource utilization efficiency of the population in comparison with other co-existing populations. A conceptually simple approach to defining such competitiveness is to consider a function that saturates both for high and low values, such as the sigmoidal function of an expression that quantifies the relative strength of the resource utilization of the population. For example, we may define the extent of competitiveness as:

$$\eta = 1 / (1 + \exp(-\alpha \cdot r / (\sum_k r_{pop-k}))) \quad (3)$$

where α is an environmental factor that characterizes the steepness of the competition in the given environment, and the summation is applied over all resource utilization values of all co-existing populations. We note that a similar approach is used in the context of calculating the competitiveness of plant species (Kattenborn et al, 2017; Hodgson et al, 1999). In our approach the sigmoidal transformation reduces the competitiveness differences in the top and bottom extremes.

Finally, the spreading of the populations will also be influenced by the presence of natural barriers, such as mountain ranges, rivers, sea, which make difficult to cross the barrier (Andras, 2015; Barraclough, Vogler and Harvey, 1998). The presence of natural barriers can be incorporated into the diffusivity of the environment, $D(x)$, where x is the location. The other environmental factors noted above, h , α , γ , θ , and the resource utilization efficiency, r , get incorporated into the reaction term, $F(q,x,t)$, of the population diffusion equation (1), where q is the size of the population and t is the time.

We aim to determine the mean resource utilization efficiency of populations originating from locations with environmental harshness h and also the mean amount of fitness improving innovation for these populations. Given that in natural environments the harshness varies relatively smoothly with the location in the environment, we adopt this assumption. We also assume that the chance of generating any resource utilization innovation is β , where $0 < \beta < 1$, independently of the location and the amount of innovation. This is a simplifying assumption.

First, let us consider the count of populations with resource utilization efficiency r at locations with environmental harshness h at time t , $g_e(r,h,t)$. At any time the difference in the number of populations moving in and those that move out of the area, all with resource utilization efficiency r , is the population move induced change of the number of such populations, in areas with harshness h . Some of these populations may die out and some new population with resource utilization efficiency r may emerge following a mutation in populations with different resource utilization efficiency. The number of newly emerging populations with resource utilization efficiency r , depends on the past number of the source populations summed up over all possible r values. In terms of equations we can write

$$\beta \cdot \sum_{\Delta r} g_e(r - \Delta r, h, t - \Delta t) = \beta \cdot \sum_{\Delta r} (g_e(r - \Delta r, h, t - \Delta t) - g_e(r - \Delta r, h, t)) + \beta \cdot \sum_{\Delta r} (g_e(r - \Delta r, h, t)) \quad (4)$$

Turning the above equation in continuous form as Δr and Δt become very small, we get the differential equation formulation for the number of emerging new populations with resource utilization efficiency r :

$$-\beta \cdot \int_{\Delta R} \partial g_e(r - \Delta r, h, t) / \partial t d\Delta r + \beta \cdot \int_{\Delta R} g_e(r - \Delta r, h, t) d\Delta r \quad (5)$$

where ΔR is the value range of Δr -s – we note that this formulation allows fractional counts of populations as well. Summing up the noted parts, we get the equation for the number of populations with resource utilization efficiency r , including all changes:

$$\begin{aligned} \partial g_e(r,h,t) / \partial t = & -a \cdot \partial^2 g_e(r,h,t) / \partial h^2 - \\ & \delta \cdot (h/r) \cdot g_e(r,h,t) - \\ & \beta \cdot \int_{\Delta R} \partial g_e(r-\Delta r,h,t) / \partial t d\Delta r + \\ & \beta \cdot \int_{\Delta R} g_e(r-\Delta r,h,t) d\Delta r \end{aligned} \quad (6)$$

where $\delta \cdot (h/r)$ is the extinction multiplier, proportional with the harshness of the environment and inversely proportional with the resource utilization efficiency of the populations. Considering $G_e(r,h,t) = \int_{\Delta R} g_e(r-\Delta r,h,t) d\Delta r$, we can write

$$\begin{aligned} \partial^2 G_e(r,h,t) / \partial t \partial \Delta r = & -a \cdot \partial^3 G_e(r,h,t) / \partial h^2 \partial \Delta r - \\ & \delta \cdot (h/r) \cdot \partial G_e(r,h,t) / \partial \Delta r - \\ & \beta \cdot \partial G_e(r,h,t) / \partial t + \\ & \beta \cdot G_e(r,h,t) \end{aligned} \quad (7)$$

We define the following function:

$$u_{r,h,t}(r',h',t') = \beta \cdot \partial G_e(r,h,t) / \partial t + \beta \cdot G_e(r,h,t) \quad (8)$$

which is the number of populations with resource utilization efficiency r that newly emerged at a location with harshness h at time t and then spread all around. This definition implies that

$$u_{r,h,t}(h',t') = 0 \quad (9)$$

for all $h' \neq h$ and $t' \leq t$ and also for $h' = h$ and $t' < t$. Using the above reasoning, we can write the equation

$$\begin{aligned} \partial u_{r,h,t}(h',t') / \partial t' = & -a \cdot \partial^2 u_{r,h,t}(h',t') / \partial h^2 - \\ & \delta \cdot (h'/r) \cdot u_{r,h,t}(h',t') \end{aligned} \quad (10)$$

given that we are not interested in this case of innovation induced emergence of new populations.

These populations persist for sufficient time, τ , if

$$\int_H u_{r,h,t}(h',t+\tau) dh' > 0 \quad (11)$$

where H is the full range of environmental harshness values. Considering the populations that persist for long time, some of these will become dominant and successful in the sense they will account for a considerable part of the overall population in the whole area. The practical meaning of sufficiently long persistence, sufficient size and whole area will depend on the context (e.g. for example this may mean a hundred years, 5% of the total population over an area of the size of an average European country, in the case of human populations). Considering that the populations will spread over locations with all h values, over a long time period, and that there will be always other populations with both higher and lower r values around, the ratio of becoming successful should not change with h , r or t , and this fixed ratio is denoted as κ . We note that if the r values can go only up to a certain limit r^* then there will be an effect on the success ratio that depends on the value of r when this is close to r^* . So, the proportion of successful populations with resource utilization efficiency r originating from an area with harshness h at time t is:

$$\kappa \cdot \int_{T(\tau)} \int_H u_{r,h,t}(h',t+\tau) dh' d\tau' / u_{r,h,t}(h,t) \quad (12)$$

where $T(\tau) = [\tau, T^* - \tau]$ is the considered time domain, T^* being the maximum time considered for the calculations.

Thus the chance for a population with resource utilization efficiency r originating from an area with harshness h to become a successful population is:

$$q(r,h) = \kappa \cdot \int_{T\#} \int_{T(\tau')} \int_H (u_{r,h,t}(h',\tau'' + \tau) / u_{r,h,t}(h,\tau'')) dh' d\tau'' \quad (13)$$

where $T\# = (0, T^*)$ is the full considered time domain and $T(\tau'') = [\tau'', T^* - \tau'']$.

Then we can calculate the chance of having a population as a result of a Δr fitness innovation, while it is originating from an area with harshness h at time t , by summing up (integrating) over all r values. This is:

$$v_{time}(\Delta r,h,t) = \int_R q(r+\Delta r,h) \cdot g_e(r,h,t) dr / \int_R g_e(r,h,t) dr \quad (14)$$

where R is the full range of r values. Integrating this over time, gives us the overall chance of having successful populations originating at a location with harshness h and as a result of a fitness innovation Δr :

$$v(\Delta r,h) = \int_{T\#} (\int_R q(r+\Delta r,h) \cdot g_e(r,h,\tau') dr / G_e(r,h,\tau')) d\tau' \quad (15)$$

where $T\#$ is defined as before.

Using these equations we can calculate the mean resource utilization efficiency and the mean fitness innovation for populations originating at locations with harshness h as follows:

$$r_m(h) = \int_R r \cdot q(r,h) dr \quad (16)$$

$$\Delta r_m(h) = \int_{\Delta R} \Delta r \cdot v(\Delta r,h) d\Delta r \quad (17)$$

where ΔR is the full range of Δr values and R is defined as above.

Given the form of equation (7) that defines $G_e(r,h,t)$ we conclude that if the $\partial^2 / \partial h^2$ component of equation (7) is non-zero then $G_e(r,h,t)$ will have a dampened harmonic (i.e. dampened sinusoidal) component in it. Considering the dependence of $q(r,h)$ and $v(\Delta r,h)$ on $G_e(r,h,t)$, the mean values of r and Δr will contain similarly a dampened harmonic component in addition to a linear component. The reason behind this is that equations similar to equation (7) can be written for $r_m(h)$ and $\Delta r_m(h)$ as well. This is valid as long as the environmental harshness, h , varies smoothly with the distance between the locations. If this is not the case, this result no longer holds. For example, if h would vary randomly, then the $\partial^2 / \partial h^2$ component of equation (7) would zero out and the expectation would be to have no dampened sinusoidal component in these mean values as a function of the environmental harshness. However, such random variation is very unlikely in natural environments.

The above reasoning explains the finding reported in Andras (2015), which found this unexpected harmonic variation in the mean values of r and Δr . Below we explore experimentally further this nature of the dependence of r and Δr on the harshness, h .

Simulation Environment

To explore the variation of the mean resource utilization efficiency, $r_m(h)$, and the mean fitness innovation, $\Delta r_m(h)$ with the harshness of the locations of origin, h , we used a simulated environment where populations spread and evolve by adopting fitness innovations. The simulation environment that we used has been described in Andras (2015), here we provide a brief description of the key features.

The world is implemented as a 2-dimensional grid of 100×60 spatial locations, without gluing of the opposing edges together (i.e. the world is not toroidal). The world contains a number of natural barriers (i.e. simulated mountain ridges), which are located randomly and have randomly set crossing difficulty (i.e. height). In all simulations we used 20 natural barriers. The world has a number of harshness hotspots and the harshness of the world locations vary smoothly with the distance from the centre of the hotspot. In some simulations we used 10 harshness hotspots, while in other simulations we used only one harshness hotspot. We adopted two options for the harshness variation with distance. In the first case the harshness depends on the inverse of the Euclidean distance, while in the second case it depends on the inverse of the Manhattan distance:

$$h(x) = 1 / (\varepsilon + ((x_1 - x_{01})^2 + (x_2 - x_{02})^2)^{1/2}) \quad (18)$$

$$h(x) = 1 / (\varepsilon + (|x_1 - x_{01}| + |x_2 - x_{02}|)) \quad (19)$$

where x_0 is the centre of the harshness hotspot and $\varepsilon > 0$.

Initially, around 1.5% of the spatial locations host a population. Each population is characterized by their resource utilization efficiency, r , that is set by calculating the value associated with a population specific sequence of bits (0 or 1) b of length L as follows:

$$r = \sum_{k=1, L1} \sum_{j=1, L2} b_{L1 \cdot (k-1) + j} \cdot 2^j \quad (20)$$

where $L1 \cdot L2 = L$ and in our case $L1 = L2 = 10$. Populations may give rise to a new population by experiencing a mutation in their bit string. The likelihood of such mutations is $\beta = 0.00008$ in our simulations. The fitness innovation corresponding to the mutation is the difference of the r values after and prior to the mutation. Following such mutations a part of the populations adopts the new resource utilization efficiency coding bit string. The population is split proportionally with the two r values.

The populations at each location grow and spread. The spreading follows a discretized version of the reaction-diffusion equation (1):

$$\begin{aligned} Q(x, t+1) = & \sum_{\mu, \nu \in \mathcal{P}} (\varphi_{x+(\tau, \nu), -\mu, -\nu, t} \cdot Q(x+(\mu, \nu), t) - \varphi_{x, \mu, \nu, t} \cdot \\ & Q(x, t)) \\ & + \rho(x) \cdot Q(x, t) \end{aligned} \quad (21)$$

where $Q(x, t)$ is the size of the population at spatial position x at time t , $\mathcal{P} = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}$, and $\varphi_{x, \tau, \nu, t}$ are stochastic diffusivity parameters, and $\rho(x)$ is the growth rate of the population at the spatial position x . The stochastic diffusivity

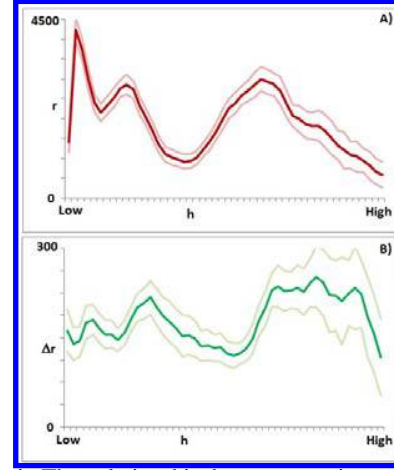


Figure 1. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A), and mean fitness innovation, Δr (B), in the case of multiple harshness hotspots in the environment (mean value dark line, 95% confidence intervals, light lines).

parameters are set such that

$$\begin{aligned} \varphi_{x+(\tau, \nu), -\tau, -\nu, t} = \omega \quad & \text{if} \\ Q(x, t) \text{ is sufficiently large, given the height value at the} & \\ \text{neighboring location } x+(\mu, \nu) & \\ \text{otherwise } \varphi_{x+(\tau, \nu), -\tau, -\nu, t} = 0 & \end{aligned} \quad (22)$$

where ω is a randomly set value, in the case of the reported simulations $\omega \in (0, 0.4)$. If $\varphi_{x+(\tau, \nu), -\tau, -\nu, t} = 0$ the populations encounters a barrier that cannot be crossed. The growth rate of the population is given by equation (2), with $\theta = 0.005$. The implementation of equation is slightly modified compared to equation (3) by adding constant terms into it:

$$\eta = 0.1 / (1 + \exp(\alpha / 2 - \alpha \cdot r / (\sum_k r_{pop-k})) \quad (23)$$

The value of α was set to 10 in some simulations and it was varied in the range of 6 to 12 in other simulations. The value of γ in equation (2) was set to 1.4 in some simulations and then it was varied in the range of 0.8 to 1.7 in other simulations.

Each simulation was run for 30,000 time turns. Typically, we started initially with around 90 populations. The populations spread and evolved. The competition for resources drove to extinction some of the populations. The newly emerged populations that innovated successfully spread and became dominant. In the simulations we considered a population dominant if it contained over 0.5% of the total population at any one time turn in the simulated world. We did not set a particular persistence limit, but naturally, for any new population it took some time to become dominant according to the above definition (typically more than 20 time turns).

For each population we considered the harshness of the location of their origin and we calculated for each harshness value, h , the mean value of the resource utilization efficiency, $r_m(h)$, and of the fitness innovation, $\Delta r_m(h)$, considering all populations originating from a location with harshness value h . For each condition that we considered we did 10 different simulations. Each simulation generated in the range of 2,500 –

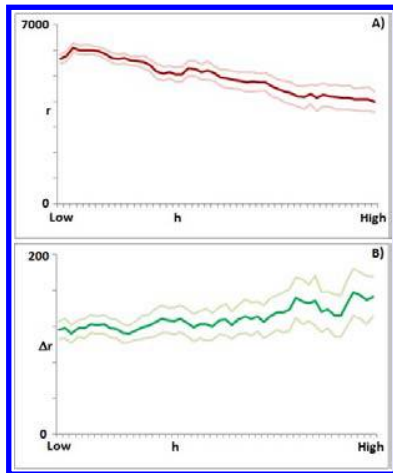


Figure 2. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A), and mean fitness innovation, Δr (B), in the case of random harshness in the environment (mean value dark line, 95% confidence intervals, light lines).

3,000 successful populations that were taken into consideration. Only the average values, where there was at least 10 different populations originating from locations with a given harshness h , were considered in the analysis and in almost all cases the averages were calculated from more than 20 instances of appropriate populations.

The computer program implementing the simulations described above is available on request from the author.

Results and Discussion

First, we considered the default case with multiple harshness hotspots in the environment and Euclidean distance based determination of the harshness of locations. The results are shown in Figure 1. As expected both relationship displays the harmonic component added onto a linear relationship with a negative slope in the case of the resource utilization efficiency and a positive slope in the case of the fitness innovation. In other words, the mean resource utilization efficiency, r_m , drops with the increase of the harshness of the location of origin and the mean fitness innovation, Δr_m , grows with the increase of the harshness of the location of origin innovation.

Intuitively this means that successful population originating from low harshness environment gain their advantage from being highly efficient in the utilization of resources. On the other hand this result also means that successful populations originating from harsh environments stand out by making high fitness innovations. The harmonic element of the relationships is due to the smooth variation of the environmental harshness with distances between locations, as we explained it earlier. This harmonic element complicates the above outlines picture of the relationships with environmental harshness. This component implies that there will be intermediate ranges of environmental harshness where either the high resource

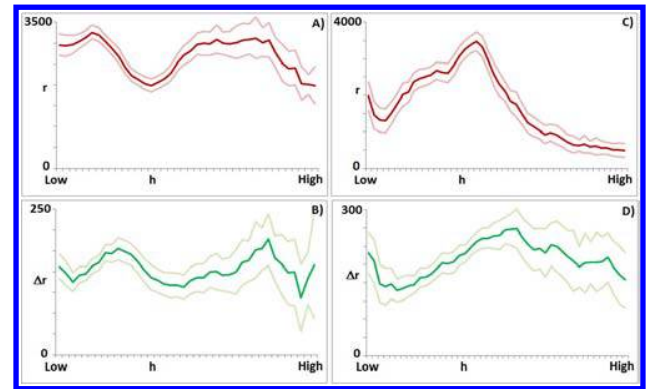


Figure 3. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A), and mean fitness innovation, Δr (B), in the case of slow variation of harshness with the Euclidean distance with one harshness hotspot; and the mean resource utilization efficiency, r (C), and mean fitness innovation, Δr (D), in the case of fast variation of harshness with the Euclidean distance with one harshness hotspot (mean value dark line, 95% confidence intervals, light lines).

utilization efficiency or high fitness innovation will be more likely than elsewhere for the locally originating new populations.

Next, we considered the un-natural case of random variation environmental harshness. The results are shown in Figure 2. While there is still some very small level of harmonic component in the relationships between h and r and Δr , the relationships are mainly linear as we expected, based on our analysis. The case of random variation of environmental harshness shows the core relationship between harshness and resource utilization efficiency and fitness innovation.

We explored next the case of having a single harshness hotspot with different variation of the harshness with the distance from the centre of the hotspot. We considered equations (18) and (19) for the definition of harshness and we also added a multiplier in the front of them to make the harshness variation fast (high multiplier – 20) or slow (low multiplier – 10). The results are shown in Figures 3 and 4. In these scenarios again we see that the harmonic component is clearly present in addition to the linear variation of r and Δr with h . The data in the figures also shows that the period length of the harmonic component extends as the speed of variation of h increases with the distance, in the case of both distance choices.

This means that faster variation of harshness with the distance comes with slower harmonic variation of r_m and Δr_m with the variation of h . This implies that if harshness changes rapidly with distance, there will be only one or at most a few harshness ranges where high resource utilization efficiency or high fitness innovation may emerge. On the other side, if the harshness changes slowly with the distance, there will be multiple, possibly many, harshness ranges where high resource

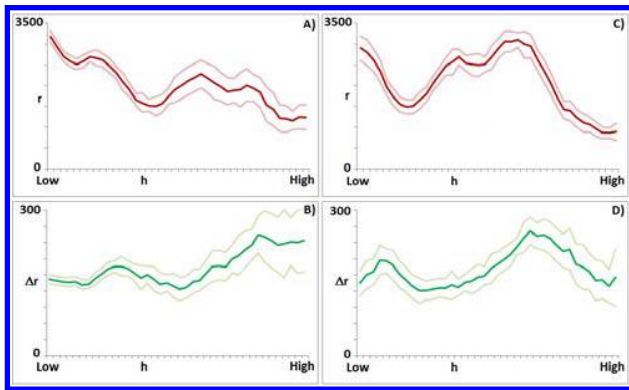


Figure 4. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A), and mean fitness innovation, Δr (B), in the case of slow variation of harshness with the Manhattan distance with one harshness hotspot; and mean resource utilization efficiency, r (C), and mean fitness innovation, Δr (D), in the case of fast variation of harshness with the Manhattan distance with one harshness hotspot (mean value dark line, 95% confidence intervals, light lines).

utilization efficiency and high fitness innovations may emerge in new populations. A practical prediction derived from this is the expectation to have one or a few ranges of height values in relatively steeply rising mountainous areas, where the most successful new species may arise, while in areas with slow change of height the expectation is to have many ranges of height where highly successful new species may emerge.

We also considered the impact of variation of the competition strength, α , and of the general population growth speed parameter, γ . The results are shown in Figures 5 and 6. In these respects, we found that stronger competition and faster growth of the populations reduces the period length of the harmonic component. The implication of this is that in environments, which are generally more competitive or support faster growth there will be multiple or many ranges of harshness values where the most innovative and most efficient new populations are likely to emerge. On the other side in environments that stunt the population growth or suppress competition between populations there will be only one or a few ranges of harshness values where the most innovative and most efficient populations may emerge.

We note that in all smooth harshness variation cases the variation of the mean resource utilization efficiency and mean fitness innovation with the harshness follow similar, but phase-shifted harmonic relationships. The conceptual explanation of this phase shift, is the difference between equations (13) and (15), which define $q(r, h)$ and $v(\Delta r, h)$, which are the likelihoods of having a newly emerging population with resource utilization efficiency r or fitness innovation Δr emerging at a location with harshness h (i.e. $v(\Delta r, h)$ is defined using an integral of $q(r, h)$).

Our theoretical analysis provides the explanation for the observed combined harmonic and linear variation of the resource utilization efficiency mean value and fitness innovation mean value with the harshness of the environment at the location of the origin. The simulation experiments confirm the expectations based on the theoretical explanation

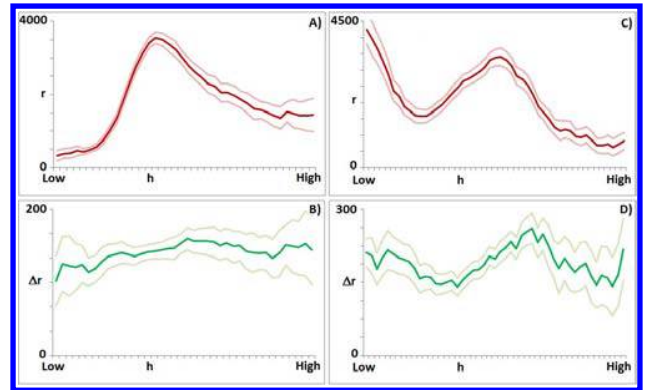


Figure 5. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A, C), and mean fitness innovation, Δr (B, D), in the case of low (A, B) and high (C, D) level competition (α) between populations, using the Euclidean distance and multiple harshness hotspot (mean value dark line, 95% confidence intervals, light lines).

and also offer a heuristic estimation of the impact of the key environmental parameters (i.e. speed of variation of harshness with distance, competition strength, population growth speed) on the nature of the relationship between the environmental harshness and the mean resource utilization efficiency and mean fitness innovation (i.e. which change in the parameters).

Our result is interesting because it implies that due to the harmonic component of the studies relationships there will be areas with particular ranges of environmental harshness where the population with the largest fitness innovations and also with the most efficient resource utilization are much more likely to emerge than in other areas with relatively similar features. In other words, the likelihood of emergence of such new species in natural environments does not vary linearly with the harshness of the environment, but follows the combined linear and harmonic relationship.

In a natural context, this implies that for example, genetic mutations that can prevent the effectiveness of a gene drive (Unckless, Clark and Messer, 2017) intervention against mosquitoes, may emerge much more likely in environment locations with certain levels of harshness, and are much less likely to emerge elsewhere. Being able to determine where such locations are likely to be, would help to maintain the effectiveness of the gene drive intervention. Similarly, this kind of analysis may help identify locations where antimicrobial resistance of bacteria (Blair et al, 2015) is more likely to emerge and possibly help to alter the environmental antimicrobial resistance of bacteria (Blair et al, 2015) is more likely to emerge and possibly help to alter the environmental features such that the antimicrobial resistance development happens in areas where it can be identified early and fought effectively.

In a socio-economic setting, this has implications for example for the likelihood of emergence of firms with disruptive innovation and those with high efficiency. Our results imply that in the context of a steeply varying regulatory and economic environment, there will be only a few ranges of regulatory and economic settings that will be conducive for the emergence of highly innovative or efficient firms. Similarly,

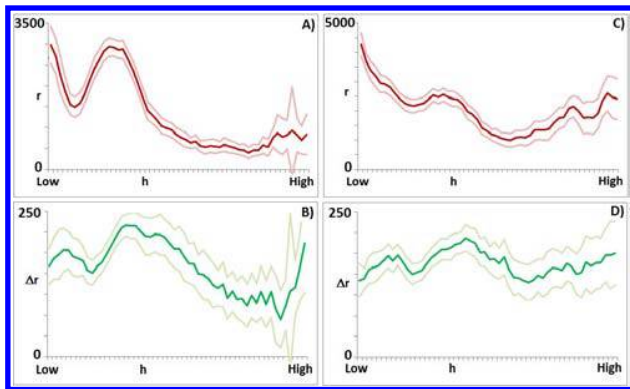


Figure 6. The relationship between environmental harshness, h , and mean resource utilization efficiency, r (A, C), and mean fitness innovation, Δr (B, D), in the case of low (A, B) and high (C, D) general growth support (γ) for populations, using the Euclidean distance and multiple harshness hotspot (mean value dark line, 95% confidence intervals, light lines).

less competition and more barriers of growth will reduce the likely ranges of regulatory and economic settings, in which highly innovative or efficient firms may emerge. On the other side, more competition, fewer growth barriers, and more uniform regulatory – economic landscape favor the presence of multiple ranges of settings in which innovative or efficient firms may emerge. Our results also imply that most likely the preferred regulatory – economic settings will not be the same for the emergence of the most innovative and most highly efficient firms.

Conclusions

In this paper we present a formal analysis of the relationship between environmental harshness and the mean resource utilization efficiency and mean fitness innovation of populations originating from locations with a given environmental harshness. We show that these relationships contain a linear and a harmonic component if the environmental harshness changes smoothly with the distance between locations. We explore experimentally the relationships considering a number of simulation environments with different parameters. The experimental analysis confirms the theoretical expectations. The experimental analysis also shows the dependence of the nature of these relationships on the environmental parameters. The experimental data also highlights that there is a phase shift between the harmonic components of the two relationships.

The experimental results show that faster variation of environmental harshness with distance extends the period length of the harmonic component of the relationships. The data also shows that more general growth support for the populations and more competition between the populations makes the period of the harmonic component of the relationships shorter.

The results are interpreted in both biological and social context. Our analysis may help to identify ways of improving the effectiveness of fight against the emergence of antimicrobial resistance in bacteria or of the application of gene drives. The interpretation in socio-economic context points to

the impact of the regulatory – economic landscape on the likelihood of emergence of firms with disruptive innovations and of highly efficient firms.

References

- Andras, P. (2015). Environmental Factors and the Emergence of Cultural – Technical Innovations. *Proceedings of the European Conference on Artificial Life 2015*, pp.130-137.
- Arthur, W.B. (2009). *The Nature of Technology*. Penguin Books, London.
- Barracough, T.G., Vogler, A.P. and Harvey, P.H. (1998). Revealing factors that promote speciation. *Philosophical Transactions of the Royal Society of London B*, 353:241-249.
- Blair, J.M.A., Webber, M.A., Baylay, A.J., Ogbolu, D.O. and Piddock, L.J.V. (2015). Molecular mechanisms of antibiotic resistance. *Nature Reviews Microbiology*, 13:42-51.
- Carelli, F.N., Hayakawa, T., Go, Y., Imai, H., Warnefors, M and Kaessmann, H. (2016). The life history of retrocopies illuminates the evolution of new mammalian genes. *Genome Research*, 26:301-314.
- Cheng, H., Yao, N., Huang, Z.-G., Park, J., Do, Y. and Lai, Y.-C. (2014). Mesoscopic interactions and species coexistence in evolutionary game dynamics of cyclic competitions. *Scientific Reports*, 4:7486.
- Diamond, J. (1997). *Guns, Germs, and Steel: The Fates of Human Societies*. Norton, New York.
- Diamond, J. and Bellwood, P. (2003). Farmers and their languages: The first expansions. *Science*, 300:597-603.
- Exposito-Alonso M., et al. (2018). The rate and potential relevance of new mutations in a colonizing plant lineage. *PLoS Genetics*, 14:e1007155.
- Fukuyama, F. (2011). *The Origins of Political Order*. Profile Books, London.
- Gould, S.J. (2002). *The Structure of Evolutionary Theory*. Harvard University Press.
- Hodgson, J.G., Wilson, P.J., Hunt, R., Grime, J.P. and Thompson, K. (1999). Allocating C-S-R plant functional types: a soft approach to a hard problem. *OIKOS*, 85:282-294.
- Holowka, N.B. and Lieberman, D.E. (2018). Rethinking the evolution of the human foot: insights from experimental research. *Journal of Experimental Biology*, 221:jeb174425.
- Kattenborn, T. et al. (2017). Linking plant strategies and plant traits derived by radiative transfer modeling. *Journal of Vegetation Science*, 28: 717-727.
- Kooyers, N.J. (2015). The evolution of drought escape and avoidance in natural herbaceous populations. *Plant Science*, 234:155-162.
- Maynard Smith, J. and Szathmari, E. (2000). *The Origins of Life: From the Birth of Life to the Origin of Language*. Oxford University Press.
- De Mazancourt, C., Johnson, E. and Barracough, T.G. (2008). Biodiversity inhibits species' evolutionary response to changing environments. *Ecology Letter*, 11:380-388.
- Millien, V. (2006). Morphological evolution is accelerated among island mammals. *PLoS Biology*, 4: e321.
- Montenegro, A., Callaghan, R.T. and Fitzpatrick, S.M. (2016) Using seafaring simulations and shortest-hop trajectories to model the prehistoric colonization of Remote Oceania. *PNAS*, 113:12658-12690.
- Nielsen, R., Akey, J.M., Jakobsson, M., Pritchard, J.K., Tishkoff, S. and Willerslev, E. (2017) Tracing the peopling of the world through genomics. *Nature*, 541:302-310.
- Turchin, P. (2006). *War and Peace and War. The Rise and Fall of Empires*, Plume, New York.
- Unckless, R.L., Clark, A.G. and Messer, P.W. (2017). Evolution of resistance against CRISPR/Cas9 gene drive. *Genetics*, 205:827-841.
- Vijaykrishna D., Mukerji R. and Smith G.J.D. (2015) RNA Virus Reassortment: An Evolutionary Mechanism for Host Jumps and Immune Evasion. *PLoS Pathogens*, 11:e1004902.
- Volpert and V. Petrovskii, S (2009). Reaction-diffusion waves in biology. *Physics of Life Reviews*, 6:267-310.
- Yom-Tov, Y., and Geffen, E. (2011). Recent spatial and temporal changes in body size of terrestrial vertebrates: probable causes and pitfalls. *Biological Reviews*, 86: 531-541.

Reproductive division of labor in a colony of artificial ants

Peter Bae¹ and Chris Marriott¹

¹University of Washington, Tacoma, WA, USA 98402
cmarriot@uw.edu

Abstract

We simulate an ant colony in which an ant's genetics can determine behavioral, morphological and physiological differences between workers and queens. We show that depending on the benefits conferred to workers and queens different reproductive division of labor strategies evolve. In particular, we observe both generalist colonies and colonies with specialized worker and queen castes. Generalist colonies were subject to selection for optimal response thresholds. Colonies with castes evolved a discrete queen caste and either a discrete or continuous worker caste. As a secondary experiment we expose our evolved colonies to a changing environment to test their ability to adapt cooperative foraging strategies and we find all reproductive division of labor strategies were effective at cooperative foraging.

Introduction

Modern life has evolved through several major evolutionary transitions (Szathmáry and Smith, 1995; Smith and Szathmáry, 1997) that resulted in the origin of protocells, prokaryotic cells, eukaryotic cells, plastids, multicellularity, eusocial animal behavior, and human societies. These major transitions are characterized by common features in which lower-level evolutionary units form cooperative groups through *division of labor* and *specialization* and then emerge as new higher-level evolutionary units with novel inheritance systems (Szathmáry, 2015). The transition from a population with no reproductive division of labor to one that does is critical in each of these transitions.

Reproductive division of labor (RDoL hereafter) describes colonies in which one or a small number of members in the colony are responsible for reproduction (i.e. queens). All other members (i.e. workers) carry out non-reproductive tasks like excavation, brood care, foraging, and defense. RDoL occurs only in *eusocial* animals like ants, bees, termites, and some mole rats (Crespi and Yanega, 1995; Burda et al., 2000; Wilson and Hölldobler, 2005; Nowak et al., 2010; Gadau et al., 2012). RDoL is one of three criteria usually used to identify eusocial animals. In addition eusocial animals have multi-generational communal cohabitation and shared responsibility for caring for their young.

We have chosen to focus our experiment on ants. There are over 14000 species of ants and social behavior is understandably varied among these species. Gadau et al. (2012) review the genetic impact of social evolution on seven different ant species for which the genome has been sequenced. These species show a variety of RDoL strategies. On one end of the spectrum is *Harpegnathos saltator* a species with very little RDoL and is considered to be the closest (of the seven species considered) to the ancestral ant species with no RDoL. On the other end of the spectrum you have species like *Atta cephalotes* in which the colony has a single queen that is larger than the workers and lives decades longer.

RDoL is characterized by behavioral specialization that may also correlate with morphological or physiological specialization. When morphological and physiological differences do exist the resulting castes can be discrete or continuous. A discrete caste has very little variation among the members of the caste. The species *Atta cephalotes* mentioned above has discrete castes. A continuous caste is one which has variation in some property within the caste and there are no gaps in the continuum. *Solenopsis invicta* is an example of an ant species in which worker size is on a continuum.

The mechanisms underlying RDoL vary between species and usually include factors in genetics, nutrition, hormones, and social contact between individuals (Beshers and Fewell, 2001; Korb et al., 2009; Gadau et al., 2012). In our experiment we are interested in the genetic factors that can impact the evolution of different RDoL strategies. One model of RDoL that takes genetic factors into considerations is the response threshold model (Bonabeau et al., 1998). The response threshold model assumes that for each task the ant has a threshold for a task-specific stimuli and the ant will select the task only when the stimuli is over the threshold. RDoL is achieved through variation among thresholds in the colony. Genetic factors can impact the ant's innate response threshold, its perception of the stimuli and its decision to carry out a task.

In Nowak et al. (2010) the authors argue that the evolution of eusocial behavior occurs through four vital stages.

The first stage is the formation of groups out of otherwise solitary individuals for the purpose of mutual defense, proximity to food and/or mates (i.e. multi-generational cohabitation). In the second stage preadaptations favorable to group life, like a propensity to cooperate and provisioning before birth, are selected for (i.e. shared responsibility for young). In the third stage the natural behavior of adult females is suppressed in many members of the population (i.e. RDoL). Finally, in the fourth stage emergent properties of the interactions between individuals can evolve leading to behaviors like cooperative foraging.

We have chosen to focus our experimentation on the third stage in which RDoL emerges and the fourth stage in which cooperative foraging strategies can evolve. We have previously studied the emergence of eusocial behavior in foraging agents capable of social learning (Marriott and Chebib, 2016) and division of labor in artificial ants in the fourth stage (Marriott and Gershenson, 2011).

We have reimplemented and extended the model of Marriott and Gershenson (2011) to create a simulation of an artificial ant colony. Ants in our model begin as cohabiting individuals sharing a single mound with preadaptations toward cooperation and provisioning. Thus, they are representative of a semisocial colony around stage two of the evolution of eusocial behavior and are intended to mimic an ancestral ant species with no RDoL or a modern species like *Harpegnathos saltator*. Every ant in the initial population is a generalist and ants forage via a solitary exploration strategy.

Our ants have a gene that in experimental settings determines the response thresholds for the breeding and foraging tasks and potentially morphological and physiological differences between queens and workers. In our control setting the gene has no effect on response thresholds and ants select the breeding task and forage task with equal probability. We see a variety of RDoL strategies evolve ranging from a colony of generalists to discrete and continuous worker and queen castes.

As a secondary experiment we have exposed our evolved colonies to a changing environment from one in which solitary exploration is adaptive into one in which cooperative foraging is adaptive. This is similar to the seasonal environments used in (Marriott and Gershenson, 2011). Our ants have a second gene which determines the response threshold for the two foraging strategies. We test the ability of our evolved colonies with different RDoL strategies to evolve cooperative foraging strategies when exposed to the second environment.

Model

Our ants live in a 2d discrete 80×60 toroid environment. The ant mound is placed in the “middle” cell and food is distributed among the cells depending on experimental settings. In our primary experiment we use an environment that

Season	Length	d	f	r
1 (Sparse)	50000	3	200	0
2	25000	8	500	0
3	25000	13	1250	500
4	25000	18	1500	1500
5 (Dense)	75000	23	5000	5000

Table 1: A summary of food growth parameters by season used in our secondary experiment.

we call *sparse*. Our secondary experiment transitions from a sparse environment to a second environment we call *dense* through three intermediate seasons that are a blend of the two environments.

Our environments differ in how food is added to the environment initially and over time. Food can either *spawn* or *regenerate*. When food spawns every 50 ticks (so long as max food is not reached) a new food patch is created with f units of food and placed in the environment in a cell that is at least d cells away from any other patch. When food regenerates every 50 ticks a food cell grows by r units of food (so long as max food is not reached). Table 1 summarizes the values used by the five seasons in our secondary experiment. The first season’s settings are identical to our sparse environment.

Each cell in the environment also records the level of incoming and outgoing pheromones. Outgoing pheromones are left by ants on their way out of the mound and can be followed back to the mound. Incoming pheromones are left by ants on their way back to the mound with food and can be followed back to the food source.

During every tick, each tile attempts to spread half of its pheromone level to its immediate neighbors in four directions. If the neighboring tile’s pheromone levels are higher than half of the current tile’s pheromone levels no spread occurs. Pheromones decay linearly at a rate of one two-hundredth of the maximum energy level.

There are two tasks that ants in the colony must perform. Food must be gathered (forage task) and larvae must be created (breed task). We call these two tasks foraging and breeding and ants specializing in these tasks (or performing these tasks) workers and queens respectively. All ants begin as generalists capable of foraging and breeding. An idle ant (one who has just matured or has just finished a task) will select a new task at random according to its response thresholds. These thresholds are determined by the ant’s genetics (see below).

When an ant selects the forage task it leaves the mound and searches for food according to a foraging strategy selected at random (response thresholds determined by the ant’s genetics). Workers are either explorers or exploiters. Explorers leave the nest and follow a random walk in which 90% of the time they continue in the same direction and 10%

of the time they turn left or right (never backwards).

Workers leaving the nest are allocated an energy level determined by genetics. This level determines how long the ant can forage outside the mound before returning home and the level of pheromones dropped while foraging. Every tick the energy level is decremented by 5 so pheromone levels drop the longer an ant is foraging. Workers normally return home when they are at full carrying capacity with food or if they deplete a tile of food. They follow the outgoing pheromones home by travelling to the cell with the highest outgoing pheromones. A worker that never finds food and runs out of energy will return home empty handed.

On the way back to the mound the ant leaves incoming pheromones (as long as it is carrying food) following the same rules as outgoing pheromones. Exploiters leave the mound and follow incoming pheromones back to food sources by travelling to the cell with the highest incoming pheromone level. Like explorers exploiters leave outgoing and incoming pheromones. If there are no incoming pheromones to follow exploiters act as explorers. When a worker returns to the nest, with food or not, it enters the idle state and selects a new task.

When an ant selects the breeding task it is placed in a pool of queens. As long as the current food level is greater than twice the sum of the current ant population and current larvae population a new larvae is created (otherwise queens idle until food level increases or they have idled too long). The new larvae is created by selecting a queen from the pool at random. This queen is the parent of the new larvae. It is not the genetic parent of the new larvae but does get fitness credit for successfully producing a new larvae. When a queen creates a larvae the queen must rest for a period determined by its genetics before entering the idle state and selecting a new task. Larvae are created until either food is depleted or there are no active queens.

Since reproduction in our simulation is asexual if the parent's genetics are selected for the child then all children of specialized queens would be queens too. Instead we select a genetic parent from the colony at random using a fitness function. This type of reproduction may seem artificial but it was chosen to mimic ant reproduction in species where queens and workers are genetically different (as it is in our ants). For instance, in *Pogonomyrmex barbatus* workers and queens are genetically different and workers and queens result from different matings (within lineage vs inter-lineage respectively) (Gadau et al., 2012; Cahan et al., 2004).

Our fitness function combines the ant's success at foraging with the ant's success at breeding. Let w be the amount of food returned to the nest (per tick) and let q be the number of larvae created (per tick) then the fitness of an ant is $w_w \cdot w + w_q \cdot q$ where w_w and w_q are weights to balance the contribution of w and q . Because workers and queens have different advantages in our experimental setup we have had to vary the weights w_w and w_q in order to ensure that there

are always queens and workers among the most fit individuals in cases where specialization occurs (otherwise a colony of all workers or all breeders will die out).

Instead of sorting our ants by fitness we use a more efficient procedure. We use the mean fitness to filter out all ants less than the mean. Then of the ants above the mean we repeat this process to isolate the top quartile of ants by fitness. A genetic parent is selected from this group uniformly at random. The two genes used in our experiments are real valued genes in the range $[0, 1]$ and straightforward mutation mechanisms are used (increment by a random value in the range $[-0.1, 0.1]$). We use a mutation rate of 5%.

When a new larvae is created it eats a meal (10 units of food) from the food stored in the mound. A new larvae must mature for 100 ticks before becoming an adult ant. When it matures it eats a second meal and then selects its first task. If there is no food when it matures it dies.

After eating a meal an ant's hunger level is reset to 0. Every tick the hunger level increases by 1 and if the hunger level reaches the threshold (500) the ant becomes hungry (so ant's must eat after 500 ticks). If away from the mound the ant attempts to return. In any case when the ant is next idle it will eat a meal from the mound. If there is no food the ant dies. Ants over the age of 100 ticks can also die with a small chance every tick determined by the ant's genetics.

Experimental Setup

Our ants have a real valued gene $breed \in [0, 1]$ that determines the response thresholds for the breeding and foraging tasks. That is, an idle ant will select the breeding task with probability $breed$ and the foraging task with probability $1 - breed$. The initial ant population in all experimental runs are generalists in that their $breed$ genes are set to 0.5. This means an idle ant from the initial population decides to breed or forage with 50% probability.

In addition to determining the response thresholds for task selection the $breed$ gene also impacts ant behavior in other ways depending on experimental settings. The breed gene also impacts the lifespan of the ant, the rate larvae can be created, the amount of food that the ant can carry, and the amount of energy an ant receives when leaving the mound.

When the breed gene impacts life span an ant will live 100 ticks and then have a $0.01 + (1 - breed) * 0.99\%$ chance to die each tick. When the breed gene impacts the rate larvae can be created an ant selecting the breed task must wait $1 + (1 - breed) * 99$ tick(s) until they can select a new task. When the breed gene impacts the amount of food the ant can carry an ant can carry a max of $1 + (1 - breed) * 99$ units of food. When the breed gene impacts the amount of energy an ant receives any ant leaving the mound will have $5 + (1 - breed) * 495$ energy. When these benefits are not activated the value is calculated with $breed = 0.5$.

Our primary experiment is to vary the strength and impact of the $breed$ gene. In our control run the $breed$ gene

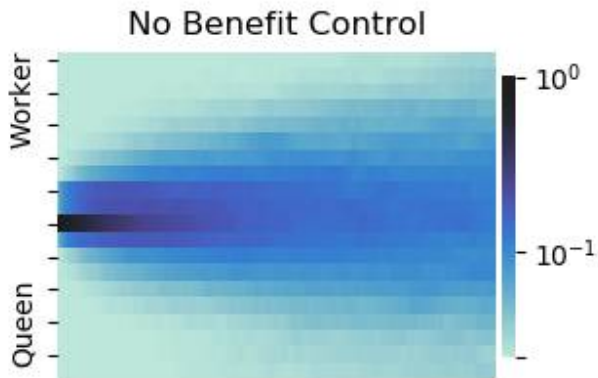


Figure 1: Histogram of the *breed* gene in the population over time for the control colony on a logarithmic plot.

has absolutely no effect. In its weakest state the gene merely impacts response thresholds. We also can activate each of the four additional morphological and physiological effects above. This leads to 16 different experimental settings of which we will focus on only 8. For each setting we have an initial population of 100 and we conducted 200 runs for 100000 ticks measuring the distribution of values in the *breed* gene.

Our ants have a second real valued gene *exploit* $\in [0, 1]$ that determines the response thresholds for the explorer and exploiter strategies for foragers. An idle ant that selects the foraging task will select the exploiter strategy with probability *exploit* and the explorer strategy with probability $1 - \textit{exploit}$. The initial ant population are explorers (*exploit* = 0) so that the initial ants mimic ancestral semisocial ants that use an explorer strategy.

In our primary experiment the colony is placed in a sparse environment in which explorers are adaptive. In our secondary experiment we take a population adapted to a sparse environment and expose it to a dense environment over a sequence of four seasons. In the dense environment exploiters are adaptive. We run our secondary experiment for 200,000 ticks to allow the colony first to evolve its RDoL strategy and then adapt to the new environment. Our second experiment reproduces some of the experiments in (Marriott and Gershenson, 2011) and tests the conditions under which a colony in stage three can move into stage four adapting from solo foragers to cooperative foragers.

Observations and Discussion

As a control run we have evolved a colony in which the *breed* gene has no effect on response thresholds at all. That is, idle ants in the control setting select the breed or forage task with equal probability. In this setting the *breed* gene is subject to genetic drift only so we should expect no selec-

tion pressure at all. Figure 1 shows the distribution of *breed* genes in the control run when subject only to genetic drift.

Our experimental colonies have evolved a number of different RDoL strategies. Some colonies remain generalists but their response thresholds face selection towards foraging in some cases and breeding in others. Other colonies specialize into two discrete queen and worker castes with very low variation among members of the castes. Finally in some cases continuous castes evolves in which members have more varied response thresholds.

In the first experiment the *breed* gene impacts response thresholds and may impact morphological and physiological differences that might benefit one caste or the other. We consider four cases: where no caste benefits, where queens benefit, where workers benefit and where both benefit. Figure 2 shows the distribution of the *breed* gene over time in these four experimental runs.

We can see that when neither caste has benefits (Figure 2 top left) the colony does not specialize. Instead the colony consists of a single continuous caste of generalists. The *breed* gene is subject to genetic drift and selection towards more reasonable response thresholds. In these settings the response thresholds settle around an average of 0.3 with a wide variation. This appears to be the optimal value given these settings as the colony can reach a higher population level than in the control settings where the response thresholds are fixed at 50%. Despite genetic drift and selection the behavior of this colony does not change much over the simulation. The single caste of generalists is similar to an ant species like *Harpegnathos saltator*.

Two genetic effects impact queen morphology and physiology. The first increases the lifespan of queens while decreasing the lifespan of workers. The second increases the rate that larvae can be created by queens while decreasing it for workers. When the benefits to queens are activated the colony does not specialize (Figure 2 top right). However, we can see that when the queens gain a benefit the response thresholds are pulled towards the queen end of the spectrum and settles around an average of 0.7. This means that the average ant selects the breed task more often (about 70% of the time) but also that they live longer and recover from breeding tasks quicker. Since the genetics do not impact foraging behaviors these generalists are as efficient foraging as in the default conditions and thus evolve to take advantage of the queen morphology but still forage enough to maintain the population. When queens benefit the single generalist caste is continuous but not as wide as in the last case.

The other two genetic effects impact worker morphology and physiology. The first increases the amount of food a worker can carry while decreasing it for queens. The second increases the energy level of a worker while decreasing it for queens. When workers benefit the colony specializes into discrete castes with little variation among its members (Figure 2 bottom left). The colony settles on about 40% queens.

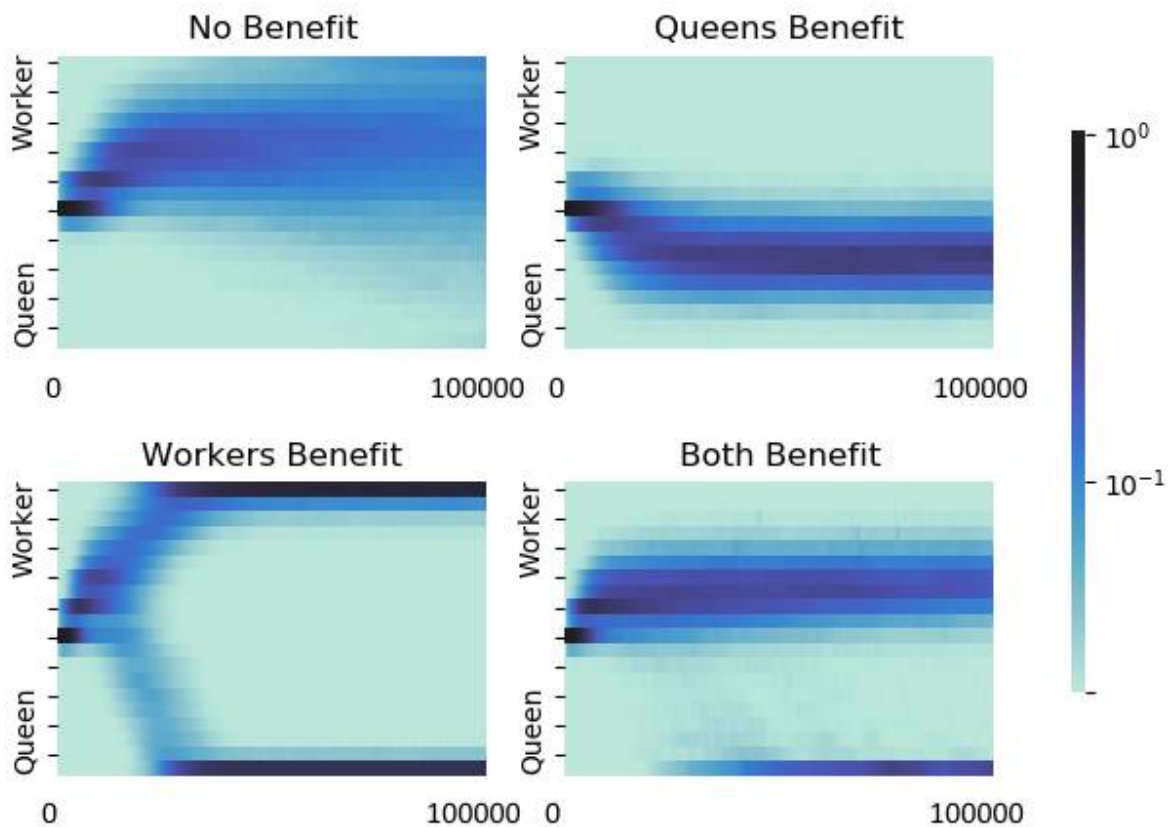


Figure 2: Histogram of the *breed* gene in the population over time for colonies in which the ants gain no benefit (top left), queens benefit (top right), workers benefit (bottom left) and both benefit (bottom right) on a logarithmic plot.

These discrete castes have morphological and physiological differences and more than one queen. This colony is similar to ant species like *Linepithema humile*.

Finally when both workers and queens benefit we get specialization but not as strong as when only workers benefit (Figure 2 bottom right). Under these conditions the queens still specialize into a discrete caste with little variation among its members. Queens make up about 20% of the population. The workers however form a continuous caste that is very similar to the generalist colony evolved when ants have no benefits (but with much less variation). Thus workers will still select the breed task from time to time, but when they do they are less efficient at it than the dedicated queens. This colony also displays castes but since its worker caste is continuous this colony is closer to an ant species like *Solenopsis invicta*.

In order to take a closer look at the benefits we have added we also look at cases where only one morphological or physiological benefit has been added. Figure 3 shows the distribution of the *breed* gene over time in runs when queens live longer, queens create larvae faster, workers can carry more

and workers travel further (but with only one benefit activated at a time).

When queens live longer no specialization occurs (Figure 3 top left). The colony evolves to a state very similar to when queens live longer and create larvae faster. The colony evolves to generalists with response thresholds favoring the breeding task. However when queens don't live longer but are able to create larvae faster the colony evolves to castes (Figure 3 top right). The highest concentration of ants are still specialized as queens or workers but the variation among the worker castes is greater than in the most specialized run (when workers benefit). Variation among the worker caste is greater than the queen caste so in this case we say that the queen caste is discrete while the worker caste has some continuity. The colony consists of approximately 30% queens.

When workers can carry more food specialization into castes occurs similar to when queens can create larvae quicker (Figure 3 bottom left). Variation among workers is greater than variation among queens so again we say the queen caste is discrete while the worker caste is continu-

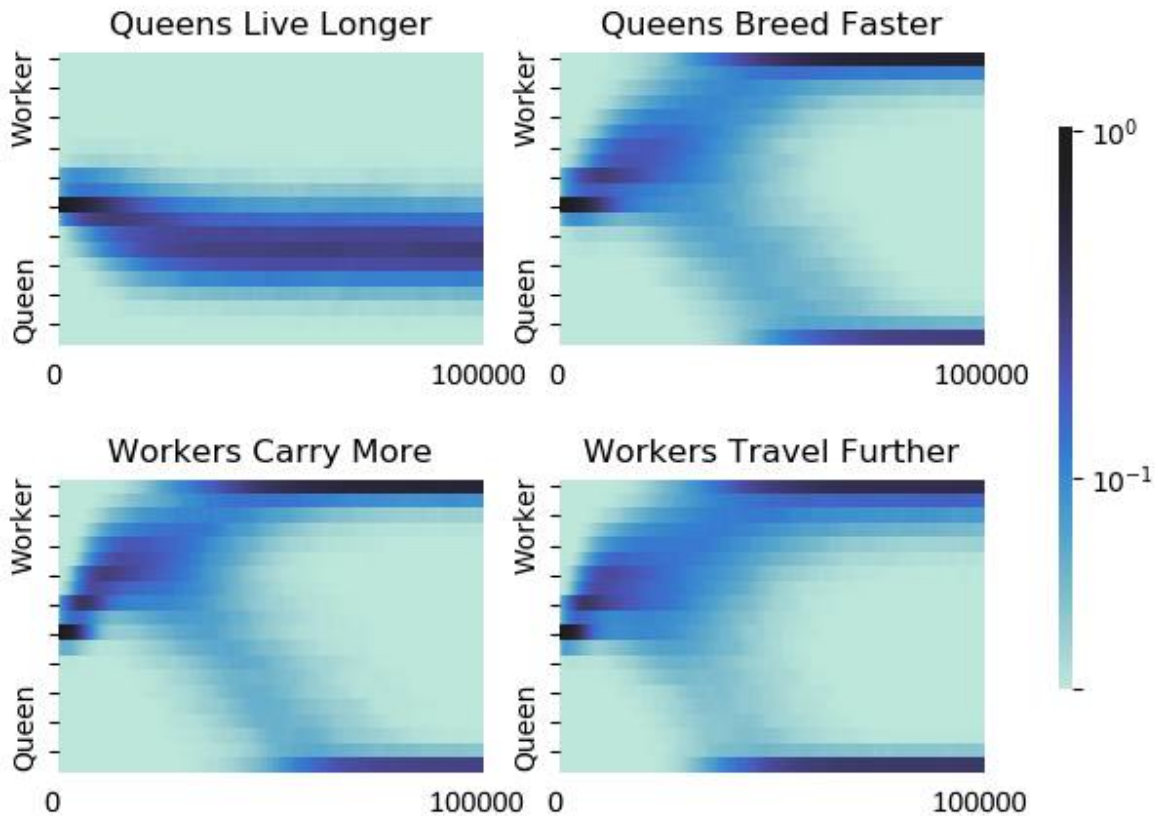


Figure 3: Histogram of the *breed* gene in the population over time for colonies in which the queens live longer (top left), queens breed faster (top right), workers carry more (bottom left) and workers travel further (bottom right) on a logarithmic plot.

ous. It is interesting that both when queens breed faster and when workers carry more we get similar specialized castes with a discrete queen caste and a continuous worker caste. These benefits, though benefiting different castes, have similar overall benefits to the colony. That is, as workers carrying more increases the rate of food brought home to the colony both of these benefits allow for larvae to be created faster. In the later case because queens have less downtime but in the prior case because with more food entering the mound more larvae can be created even with the same downtime. In the prior case this also means a larger population can be achieved (see Table 2). When workers can spend more time out of the mound we get a strong specialization in queens. Workers still remain in a varied continuous caste but the variation is greater than in the last two cases.

Table 2 summarizes the RDoL strategies evolved in each run. In addition it indicates the stable population level that the colony could achieve for each run. We can see that the control settings reaches a stable population of about 1100. The environment remains the same in each run (that is, the same amount of food is available) but the experimental colonies are subject to selection pressures that seek to

Run	Strategy	Pop.	w_q	w_w
Control	generalist	1100	1	1
FFFF	generalist workers	1300	5	2
FFTT	discrete castes	1700	7	2
TTFE	generalist queens	1450	5	2
TTTT	continuous workers discrete queens	1400	3	2
FFFT	continuous workers discrete queens	1500	5	2
FFTF	continuous workers discrete queens	1450	6	2
FTFF	continuous workers discrete queens	1300	5	2
TFFF	generalist queens	1400	5	2

Table 2: A summary of results from our eight primary runs. The run name corresponds to which benefits were activated in the run. The order of benefits is queens live longer, queens create larvae faster, workers carry more and workers travel further. We also indicate the stable population achieved in the run and the values of w_q and w_w used for the run.

increase population size. Even with no morphological or physiological benefits selection adjusts the response thresholds so that a population of about 1300 could be maintained.

The most efficient run was when workers benefit but queens do not. Recall in this case we have discrete queen and worker castes and we also reach a stable population of 1700. Most other benefits allowed for a higher population to be reached than the default settings except when queens create larvae faster. When queens create larvae faster but there are no other benefits there is no way to increase food into the mound so the population cannot grow larger. When queens live longer a higher population can be maintained because queens remain in the population longer.

We notice that in all runs in which castes evolve the queen caste is always discrete. We explain this due to an asynchronicity between queens performing the forage task and workers performing the breed task. A worker performing the breed task is no worse than a queen at creating a larvae (it must only wait longer before selecting a new task and may not live as long as queens). On the other hand a queen performing the forage task will have a very low energy level and cannot carry much. This means most queens that select the forage task return home empty handed or with a very small amount of food. This is detrimental to their fitness and thus selection is much stronger for queens to avoid foraging.

Another trend we notice in runs where specialization occurs is that specialization occurs towards foraging first. Workers receive immediate fitness benefit from specialization in cases where workers benefit. Carrying more and spending more time out of the mound means less failed foraging attempts and more food returned per trip. On the other hand queens gain no fitness bonus from living longer (since our fitness function considers number of larvae created *per tick* not over their lifetime). Secondly creating larvae more rapidly does impact the fitness value but only if you are selecting the breeding task reliably (i.e. a specialized queen). However, once workers begin to specialize (and are less efficient at breeding) selection for specialized queens becomes stronger and benefits from breeding faster are amplified.

This seems most clear in the run where all benefits are activated. In this run workers specialize only into a continuous generalist forager castes similar to when no benefits are activated. However, this is enough to amplify the selection pressure of even a single or a few specialized queens and a discrete caste of queens still evolves.

In our secondary experiment we took our colonies that were adapted to the sparse environment and exposed them to a dense environment over four additional seasons. As the seasons transition from sparse to dense the adaptive foraging strategy changes from solitary exploration to cooperative foraging using pheromone trails. Figure 4 shows the distribution of the *exploit* gene over time for populations exposed to the dense environment. The dense environment arrives at tick 125,000 of 200,000.

We can see that adaptation to exploiters occurs in all experimental conditions. However, complete specialization only occurs in runs where no queen caste exists. The reason for this is that only in these population do all members face selection for their foraging strategy. When there is a specialized queen class there is no selection pressure on the *exploit* gene for members of the queen class. As they never forage they're foraging strategy is irrelevant and is subject to genetic drift. In cases of exploiter specialization the exploiters are still continuous and in cases where exploiters don't fully specialize we still see some explorers. This is because explorers are still needed even when exploitation is adaptive. This is why exploiters act as explorers in absence of incoming pheromones.

Conclusions

In our primary experiment we considered an ant colony in which an ant's genetics determines its response thresholds to the breeding and foraging tasks. Therefore these ants have the potential to evolve into genetically distinct behavioral castes. In our experimental settings the ant's genetics determine response thresholds and morphological and physiological differences between specialist workers or queens.

When the ant's genetics only impact response thresholds our ant colony evolves to be a colony of generalists with a preference for foraging. Generalists with a preference for breeding evolve when queens live longer than workers and workers have no benefits. If queens do not live longer than workers but can create larvae quicker the colony evolves a discrete queen caste and a continuous worker caste.

A discrete queen caste and a continuous worker caste also evolve when workers can carry more. When workers travel further and when both workers and queens benefit a discrete queen caste with little variation evolves along side a continuous caste of generalist workers. Finally, two discrete castes with little variation evolve when workers can carry more and can spend more time out of the mound. These are the only settings in which two discrete castes evolve.

These varied RDoL strategies evolved under different genetic circumstances. This parallels the wide range of RDoL strategies observed among different ant species. Our experiment has explored how the presence of morphological and physiological differences among ants performing different tasks can lead to the evolution of different RDoL strategies.

Our secondary experiment examined how colonies with different RDoL strategies can adapt to a changing environment and evolve cooperative foraging strategies. We have shown that all types of RDoL strategies we've studied can evolve cooperative foraging strategies.

References

- Beshers, S. N. and Fewell, J. H. (2001). Models of division of labor in social insects. *Annual review of entomology*, 46(1):413–440.

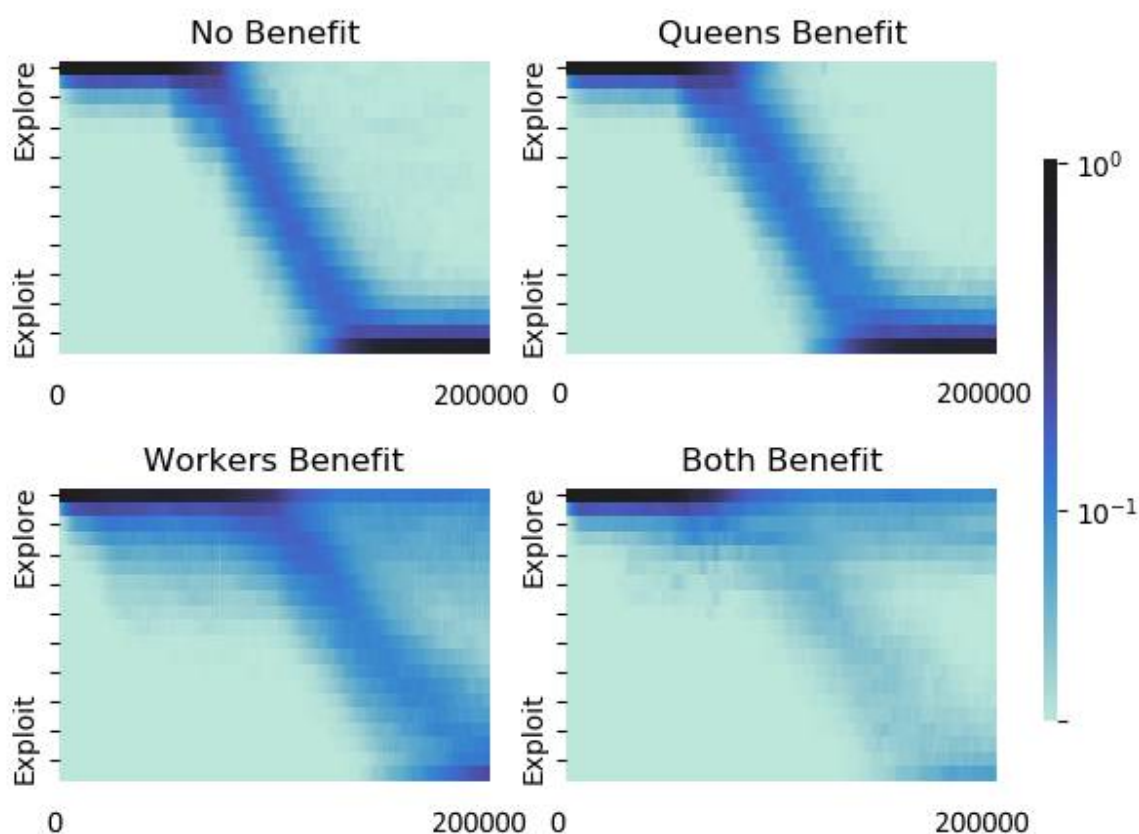


Figure 4: Histogram of the *exploit* gene in the population over time for colonies in which ants gain no benefit (top left), queens benefit (top right), workers benefit (bottom left) and both benefit (bottom right) on a logarithmic plot.

- Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L. (1998). Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60(4):753–807.
- Burda, H., Honeycutt, R. L., Begall, S., Locker-Grütjen, O., and Scharff, A. (2000). Are naked and common mole-rats eusocial and if so, why? *Behavioral Ecology and Sociobiology*, 47(5):293–303.
- Cahan, S. H., Julian, G. E., Rissing, S. W., Schwander, T., Parker, J. D., and Keller, L. (2004). Loss of phenotypic plasticity generates genotype-caste association in harvester ants. *Current Biology*, 14(24):2277–2282.
- Crespi, B. J. and Yanega, D. (1995). The definition of eusociality. *Behavioral Ecology*, 6(1):109–115.
- Gadau, J., Helmkampf, M., Nygaard, S., Roux, J., Simola, D. F., Smith, C. R., Suen, G., Wurm, Y., and Smith, C. D. (2012). The genomic impact of 100 million years of social evolution in seven ant species. *Trends in Genetics*, 28(1):14–21.
- Korb, J., Weil, T., Hoffmann, K., Foster, K. R., and Rehli, M. (2009). A gene necessary for reproductive suppression in termites. *Science*, 324(5928):758–758.
- Marriott, C. and Chebib, J. (2016). Finding a mate with eusocial skills. In *Proceedings of the Artificial Life Conference 16*, pages 298–305. MIT.
- Marriott, C. and Gershenson, C. (2011). Polyethism in a colony of artificial ants. In *ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 498–505. MIT.
- Nowak, M. A., Tarnita, C. E., and Wilson, E. O. (2010). The evolution of eusociality. *Nature*, 466(7310):1057–1062.
- Smith, J. M. and Szathmáry, E. (1997). *The major transitions in evolution*. Oxford University Press.
- Szathmáry, E. (2015). Toward major evolutionary transitions theory 2.0. *Proceedings of the National Academy of Sciences*, 112(33):10104–10111.
- Szathmáry, E. and Smith, J. M. (1995). The major evolutionary transitions. *Nature*, 374(6519):227–232.
- Wilson, E. O. and Hölldobler, B. (2005). Eusociality: origin and consequences. *Proceedings of the National Academy of Sciences of the United States of America*, 102(38):13367–13371.

Exogenous Rewards for Promoting Cooperation in Scale-Free Networks

Theodor Cimpanu¹, The Anh Han¹, Francisco C. Santos²

¹School of Computing and Digital Technologies, Teesside University

²INESC-ID and Instituto Superior Tecnico, Universidade de Lisboa

Emails: {T.Cimpanu,T.Han}@tees.ac.uk

Abstract

The design of mechanisms that encourage pro-social behaviours in populations of self-regarding agents is recognised as a major theoretical challenge within several areas of social, life and engineering sciences. When interference from external parties is considered, several heuristics have been identified as capable of engineering a desired collective behaviour at a minimal cost. However, these studies neglect the diverse nature of contexts and social structures that characterise real-world populations. Here we analyse the impact of diversity by means of scale-free interaction networks with high and low levels of clustering, and test various interference mechanisms using simulations of agents facing a cooperative dilemma. Our results show that interference on scale-free networks is not trivial and that distinct levels of clustering react differently to each interference mechanism. As such, we argue that no tailored response fits all scale-free networks and present which mechanisms are more efficient at fostering cooperation in both types of networks. Finally, we discuss the pitfalls of considering reckless interference mechanisms.

Introduction

The problem of explaining collective behaviours among self-interested individuals in evolving dynamical systems has fascinated researchers from many fields, and is a well studied research topic in evolutionary game theory (Hofbauer and Sigmund, 1998). It can be found in a variety of real-world situations, ranging from ecosystems to human organisations and technological innovations and social networks (Santos et al., 2006; Sigmund et al., 2001; Raghunandan and Subramanian, 2012; Han et al., 2019). It has been also investigated in various Artificial Life systems such as swarm-based systems and biologically inspired artificial social systems (Nitschke, 2005; Bonabeau et al., 1999).

In this context, cooperation is typically assumed to emerge from the combined actions of individuals within the system. However, in many scenarios, such behaviours are advocated and promoted by an external party, which is not part of the system, calling for a new set of heuristics capable of *engineering* a desired collective behaviour in a self-organised complex system Penn et al. (2010). For instance, if one considers a near future, where hybrid societies comprising humans and machines shall prevail, it is important to

identify the most effective incentives to be included to leveraging cooperation in such hybrid collectives (Paiva et al., 2018). In a different context, let us consider a wildlife management organisation (e.g., the WWF) that aims to maintain a desired level of biodiversity in a particular region. In order to do that, the organisation, not being part of the region's eco-system, has to decide whether to modify the current population of some species, and if so, then when, and in what degree to *interfere* in the eco-system (i.e., to modify the composition of the population) (Levin, 2000). Since a more impactful intervention typically implies larger costs in terms of human resources and equipment, the organisation has to achieve a balance between pregnant wildlife management and a low total investment cost. Moreover, due to the evolutionary dynamics of the eco-system (e.g., frequency and structure dependence) (Santos et al., 2006), undesired behaviours can reoccur over time, for example when the interference was not sufficiently strong in the past. Given this, the decision-maker also has to take into account the fact that it will have to repeatedly interfere in the eco-system in order to sustain the level of biodiversity over time. That is, it has to find an efficient interference mechanism that leads to its desired goals, while also minimising the its total cost.

This question has been studied previously in the context of populations distributed on regular graphs, namely the complete and the square lattice graphs (Han and Tran-Thanh, 2018; Han et al., 2018). In this type of network, every individual has the same degree of connectivity (i.e. the number of neighbours). However, in social graphs and real-world populations, individuals typically have a diverse social connectivity (Albert and Barabási, 2002; Santos et al., 2008). Hence, in this paper, we study cost-effective interference in heterogeneous networks, namely different types of scale-free networks, which have been shown to well capture real-world networks (such as the World Wide Web) (Newman, 2018). In particular, we consider populations of individuals distributed in a scale-free network, who interact with their neighbours via the one-shot Prisoner's Dilemma (PD), where uncooperative behaviour is preferred over cooperation (Sigmund et al., 2001; Santos et al., 2006). As an

outsider decision-maker, we aim to promote cooperation by interfering in the system, rewarding particular agents in the population at specific moments.

The research question here is to identify when and how much to invest (on individuals distributed in a network) at each time step, in order to achieve our desired ratio of cooperation within the system such that the total cost of interference is minimised, taking into account the fact that individuals might have different levels of social connectivity. For instance, we might wonder whether it is sufficient to focus the investment only on highly connected cooperators since they are more influential, thereby leading to cost-efficiency? Do we need to take into account a neighbourhood's cooperativeness level which was shown to play an important role in square lattice networks (Han et al., 2018)? Also, when local information is not available and only global statistics can be used in the decision making process, how different are the results in heterogeneous networks, in comparison to regular graphs?

To answer these questions, this paper will systematically investigate different general classes or approaches of interference mechanisms, which are based *i*) on the global population statistics such as its current composition, *ii*) a node's social connectivity in the network and *iii*) the neighbourhood properties such as the local cooperativeness level.

Our results show that interference in a heterogeneous network exhibits a significantly more complex challenge (to be cost-effective while ensuring high levels of cooperation) and much richer nonlinear dynamic behaviours, compared to regular graphs. For instance, in both well-mixed and square lattice graphs, a greater per-individual investment cost would ensure at least the same level of cooperation since it gives each cooperator a better fighting chance for survival against defectors. However, this is not the case in the context of heterogeneous networks as increasing the per-individual investment cost could actually be detrimental for cooperation.

The rest of the paper is structured as follows: the next section provides a brief overview of the related work, which is followed by a detailed description of our model, methods and its results. The paper ends with a final discussion.

Related Work

The problem of explaining the emergence and stability of cooperative behaviour has been studied intensively in many fields, from Social Sciences, Economics, Physics to Multi-agent Systems and Artificial Life (Hofbauer and Sigmund, 1998; Nowak, 2006; Han et al., 2012; Nitschke, 2005). Several mechanisms responsible for the evolution of cooperation have been identified, including direct and indirect reciprocity (Nowak and Sigmund, 2005), kin and group selections (Traulsen and Nowak, 2006), network reciprocity (Santos and Pacheco, 2005; Santos et al., 2006), punishment and rewarding (Sigmund et al., 2001), and cognitive mecha-

nisms (Han et al., 2011, 2012). However, these mechanisms do not consider how cooperation can be promoted by an external party. Instead, they are incorporated as part of individual strategic behaviours, in order to study how they evolve and whether their evolution promotes a better outcome for cooperative behaviour. In contrast, our interference mechanisms are external, i.e. they are not incorporated into the individual strategy.

In addition, the aim of our mechanisms is to minimise the cost of interference while guaranteeing high levels of cooperation, contrary to past literature where the cost optimisation is often omitted. In this respect, our work is also different from the modelling works of institutional incentives to encourage cooperation through costly reward and punishment (Sigmund et al., 2010; Vasconcelos et al., 2013) as well as through enforcing agreements (Han et al., 2017).

Similarly, our work also differs from EGT literature on optimal control in networked populations (Riehl and Cao, 2017; Ramazi and Cao, 2015), where cost-efficiency is not considered. Instead, these works on controllability focus on identifying which individuals or nodes are the most important to control (i.e. where individuals can be assigned strategies as control inputs), for different population structures.

Closely related to the current work are the analyses on well-mixed populations (i.e. having a fully connected graph structure) (Han and Tran-Thanh, 2018) and on square-lattice structured populations (Han et al., 2018), which study cost-efficient interference on the aforementioned types of networks, respectively. Moving to the more complex scenario of heterogeneous networks where individuals might have different degrees of connectivity (i.e. the number of neighbours), an interference mechanism might need to take this new dimension into account to be cost-efficient. As shown below, cost-efficient interference mechanisms that incorporate this information can outperform those who only consider global population statistics and neighbourhood cooperative properties as in previous works.

Also related to current work is the research of cooperation in social networks where changes are initiated from inside the system (Raghunandan and Subramanian, 2012; Franks et al., 2013, 2014). Among them, more relevant to our paper is the recent work by Franks *et al.* (Franks et al., 2014), which has explored the use of influencers on complex networks. However, these influencers are also part of the system and thus, similar to the cases mentioned above, this work does not consider external interference mechanisms. Given this, it does not address similar decision-making problems that we examine here.

Models and Methods

Prisoner's Dilemma on Scale Free Networks

We consider a population of agents on scale-free networks of contacts (SF NOCs)—a widely adopted heterogeneous population structure in population dynamics and evolutionary

games (for a survey, see (Szabó and Fath, 2007)). We focus our analysis on the efficiency of various interference mechanisms in spatial settings, adopting an agent-based model directly comparable with the setup of recent lab experiments on cooperation (Rand et al., 2014).

Initially each agent is designated either as a cooperator (C) or defector (D) with equal probability. Agents' interaction is modelled using the one-shot Prisoner's Dilemma game, where mutual cooperation (mutual defection) yields the reward R (penalty P) and unilateral cooperation gives the cooperator the sucker's payoff S and the defector the temptation T . As a popular interaction model of structured populations (Szabó and Fath, 2007), we adopt the following scaled payoff matrix of the PD: $T = b$, $R = 1$, $P = S = 0$. (with $1 < b \leq 2$). We adopt a weak version of the Prisoner's Dilemma in spite of cooperator prevalence shown in previous works on scale-free networks of contacts (Santos et al., 2008), so as to have a direct link of comparison with studies on the effects of rewarding mechanisms in different types of networks (Han et al., 2018).

For SF networks with low clustering we adopt the famous Barabási-Albert (BA) model (Albert and Barabási, 2002). Starting from a complete graph of m_0 nodes, at every time-step one adds new node with $m \leq m_0$ edges linking to existing nodes, which are chosen with a probability that is proportional to the number of links that the existing nodes already have. The new node always connects to m distinct nodes, and duplicate connections at each time step are not allowed. The average connectivity of the network is $z = 2m$.

To obtain a SF network with high clustering, we resort to the Dorogovtsev-Mendes-Samukhin (DMS) model (Dorogovtsev et al., 2001). Similarly to the BA model, we also have growth, yet each new node attaches to both ends of a randomly chosen edge. As a result, we favor the creation of triangular relations between individuals, thereby greatly enhancing the clustering coefficient of the final network. As in the BA model, the process of choosing the edge implicitly promotes the preferential choice of highly connected nodes, leading to the same degree distribution. The edges chosen at each time step are distinct and multiple connections between the same two nodes are not allowed. This network also has an average connectivity of $z = 2m$. Both types of SF NOCs are pre-generated, before the strategies of players are designated and before the first generation commences playing.

At each time step or generation, each agent plays the PD with its immediate neighbours. The score for each agent is the sum of the payoffs in these encounters. Before the start of the next generation, the conditions of interference are checked for each agent and, if they qualify, the external decision maker increases their payoff. Multiple mechanisms (i.e. multiple conditions) can be active at once. At the start of the next generation, each agent's strategy is changed to that of its highest scored neighbour (Nowak and May, 1992; Szabó and Fath, 2007). Our analysis will be primarily based

on this deterministic, standard evolutionary process in order to focus on understanding the cost-efficiency of different interference mechanisms.

We simulate this evolutionary process until a stationary state or a cyclic pattern is reached. The simulations converge quickly, with the exception of some cyclic patterns which do eventually reach a stationary state. Because this work studies cost effective intervention, these rarely-occurring patterns which inherently invite very large total costs are escaped early by running simulations for only 75 generations, at which point the accumulated costs are excessive enough for this mechanism to not be of interest. Moreover, the results are averaged for the last 25 generations of the simulations for a clear and fair comparison (e.g. due to cyclic patterns). In order to improve accuracy related to the randomness of network topology in scale-free networks, each set of parameter values is ran on 10 different graphs for both types of SF NOCs. Furthermore, the results for each combination of network and parameter values are obtained from averaging 30 independent realisations. It is important to note that the distribution of cooperators and defectors on the network is different for every realisation.

Note that we do not consider mutations or random explorations in this work. Thus, whenever the population reaches a homogeneous state (i.e. when the population consists of 100% of agents adopting the same strategy), it will remain in that state regardless of interference. Hence, whenever detecting such a state, no further interference will be made.

Cost-Efficient Interference in Networks

As already stated, we aim to study how one can efficiently interfere in a structured population to achieve high levels of cooperation while minimising the cost of interference. An investment in a cooperator consists of a cost $\theta > 0$ (to the external decision-maker/investor). This investment is added to the payoff of an agent if certain conditions are met. Each mechanism has different conditions for investment. In particular, we investigate whether global interference mechanisms (where investments are triggered based on network level information) or their local counterparts (where investments are based on local neighbourhood information) lead to successful behaviour with better cost efficiency. To do so, we consider three main classes of interference mechanisms based *i*) on the global composition of the population, *ii*) the node's connectivity in the network and *iii*) the neighbourhood cooperation level.

1. Population composition based (POP): In this class of mechanisms the decision to interfere (i.e. to invest on all cooperators in the population) is based on the current composition of the population (we denote x_C the number of cooperators currently in the population). Namely, they invest when the number of cooperators in the population is below a certain threshold, p_C (i.e. $x_C \leq p_C$), for $1 \leq p_C \leq z$. They do not invest otherwise ($x_C > p_C$). The value p_C describes

how widespread the defection strategy should be to trigger the support of cooperators' survival against defectors.

2. Node Influence (NI): For this mechanism, the decision to invest in a given cooperator is dependent on how influential its node is (i.e. how many connections end in that node). Whereas POP considered the composition of the population, NI looks at how connected a node is in the network. That is to say, the decision-maker invests in a cooperator node C when the number of its immediate neighbours ($|k_C|$) divided by the maximum connectivity ($\max |k_i|$) is above a threshold of influence c_I , for $0 \leq c_I \leq 1$. Otherwise, i.e. when $0 \leq c_I \leq \frac{|k_C|}{\max |k_i|}$, no investment is made. The value c_I describes how influential a cooperator node should be to trigger an investment into its survival.

3. Local cooperation based (LC): In this class of mechanisms, the decision to invest in a given cooperator is based on the cooperativeness level in that cooperator's neighbourhood. Namely, the decision-maker invests in a cooperator when the number of its cooperative neighbours is below a certain threshold, n_C , for $0 \leq n_C \leq |k_C|$; otherwise, no investment is made. By varying the local cooperation threshold n_C , we aim to provide an answer to the important question of how much cooperation is required in a neighbourhood before the investor can choose to withhold the intervention and save the interference cost and under which conditions this can happen. For instance, one can ask whether it is safe to withdraw action in a neighbourhood without affecting the outcome, therefore eliminating unnecessary interference.

Interestingly, these mechanisms require different levels of information which may or may not be readily available in the given network. In some cases, such as social networks, the connectivity (i.e. the number of friends) of a node is virtually free information which requires no effort on the part of the external decision maker to discern. On the other hand, other mechanisms such as POP, inherently require more information about the population and the level of cooperativeness in different parts of the network. POP is a broad mechanism which only requires knowledge about overall cooperativeness, but LC invites even more detailed observations, in order to determine the cooperativeness in each neighbourhood. Combining NI with LC generally does not require any more observation than LC by itself. Our study of neighbourhood based interference does not take into account the cost of gathering information, it is a direct comparison between perceived gains in cooperation and the associated per-individual cost of interference set out in the interference mechanisms.

Results

In contrast to the study on square lattice networks (Han et al., 2018), as detailed below for each interference mechanism, we found that performing cost-effective interventions

on SF NOCs presents multiple concerns. In a square lattice population, more detailed observations resulted in more effective intervention with a better outcome. On the other hand, more knowledge about the population in SF NOCs simply reduces the risk of interfering to the detriment of cooperators. In other words, interfering in SF NOCs without adequate knowledge should be approached cautiously or it could act to the benefit of defectors. This issue is prevalent in the BA model and is not representative of the DMS model.

Positive interference in BA models broadly requires very high θ values (often orders of magnitude higher than similar mechanisms performed on square lattice populations) or a blanketing mechanism that targets all or almost all cooperators, even those which are not necessarily in danger of converting to D. Converging to 100% C is very difficult unless both of these conditions are met and this introduces multiple concerns in the role of an exogenous interfering party. We avoid focusing on solutions where the per-generation cost is excessive, as it is unlikely for any institution to be able to produce such exorbitant sums in one generation, as required by these heterogeneous networks, instead we focus on effective intervention with manageable amounts of per-generation cost. In the following subsections we detail the results obtained for each interference mechanism.

Population Based

We compare population-based interference mechanisms, i.e. POP, on the two different types of SF NOCs, the BA model and the DMS model, namely how efficient the mechanisms are at promoting cooperation with minimal total cost (See Figure 1).

For SF NOCs with a large clustering coefficient, we found that it is very easy to escape cyclic patterns and a minimal amount of interference, enabling the population to quickly converge to 100% cooperation. Without any interference, the frequency of Cs is greatly dependent on the initial distribution of strategies in the network and there is a large probability that Ds will quickly overtake the Cs, if the oldest (i.e. the most connected) nodes are initially Ds. Conversely, applying even a minimal amount of interference to Cs, at any point in the rapid decline of C population, helps Cs in converging to 100% of cooperation. Because of this, investing any more than minimal amounts ($\approx \theta$), as well as interfering when the average cooperation is above 50%, increases the total cost with little to no benefit to the frequency of Cs. Note that the results are consistent for a larger cost θ . We plot up to $\theta \leq 5$ just for the sake of clear presentation.

In direct contrast with our findings for the DMS model, an external decision maker should only interfere in BA models with great care, as investing without discrimination could lead to a lower cooperation frequency when compared to no interference (See Figure 1). We observe that using certain values of θ negatively impacts average cooperation levels across a wide range of p_C values. For these undesired val-

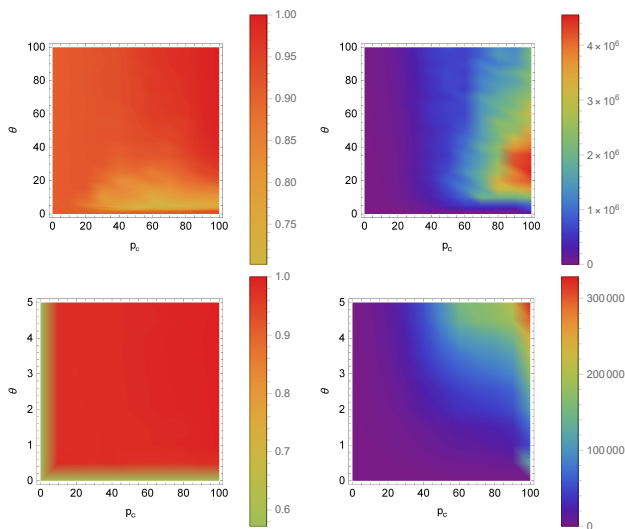


Figure 1: **Population-based (POP) interference for BA model (top row) vs DMS model (bottom row)**, for varying per-individual cost of investment θ , as well as the threshold of population cooperation p_C . The left column reports the frequency of cooperation while the right one reports the total cost required. We note in particular, the significant difference in total cost between the two models. Parameters: $b = 1.8$; $n = 5000$; $z = 4$ (average node connectivity).

ues of θ , cyclic patterns would form which ultimately help Ds by maintaining C players in clusters dominated by Ds (see Figure 2). This type of negative impact occurs when the θ value is not high enough for Cs to be able to convert a cluster to cooperation, but not low enough as to let the Ds converge to 100% D in that cluster. Many of these cyclic patterns eventually settle to 100% C if the simulation is ran for a sufficient number of generations (≈ 250). We note that the accumulated cost of interference at the end of the long-lasting cyclic patterns is prohibitively large, which make such values of θ undesirable for an external decision maker with limited resources.

Positive interference in BA models can be achieved by selecting very low or extremely high values for θ , with a high value for p_C . BA models converge to a high C frequency even without interference, so it is important to select a value for p_C that will allow interference after the system has reached a stable state (typically $p_C > 90$). In terms of total cost, it is more efficient to select very low values of θ , but the overall benefit to cooperation levels is much lower than with very high values of θ . Therefore, it is up to the external decision maker to decide if the increase in cooperation is worth the higher cost in resources.

Node Influence

When an exogenous decision maker takes into account only how connected a node is, see Figure 3, i.e. how influential it is in the network, it becomes very unlikely for interference to provide a meaningful improvement to levels of C in BA

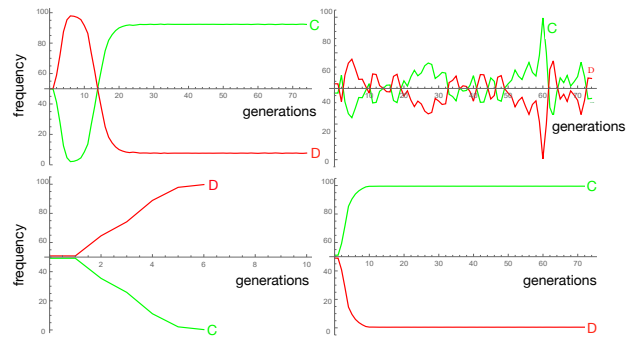


Figure 2: **Evolution of cooperation: BA Model (top) vs DMS model (bottom)**, for $\theta = 5$, $p_C = 80$. C and D frequencies are shown in green and red, respectively. The left column shows the network without interference, while the right one shows the same network after population-based (POP) interference. Other parameters: $b = 1.8$; $n = 5000$; $z = 4$ (average node connectivity).

models. Very low values of c_I (≈ 0.1), coupled with small to intermediate values of θ can cause a decrease in the average cooperation, forming previously discussed cyclic patterns (See Figure 2). For all other values, cooperation seems to be very inert. This phenomenon can be explained by the fact that clusters have already been decided in favour of cooperators and all that remains of defectors survives in a stable state around non-influential nodes. By targeting only the most influential cooperators, the external decision maker can have no impact on the less connected nodes, which enable the survival of defectors. Therefore, only a blanketing mechanism at very low c_I can reach the lowly connected cooperators and produce an increase in the average cooperation. This type of blanketing mechanism with low c_I quickly accumulates large amounts of total investment cost.

In SF networks with a high clustering coefficient, on the other hand, one can ensure convergence to 100% C in a cost effective way by selecting intermediary values of c_I (typically $\approx c_I = 0.6$) and low values for θ . An interesting observation is that contrary to POP, interference does not mean that the system will converge to 100% C. Anything more than minimal amounts of c_I show no increase to average of cooperation except at extremely high θ values. In other words, it appears that D clusters are very difficult to shift after the initial distribution of strategies on the network and the amounts of fitness they acquire are almost impossible to match except for a very large individual investment into the oldest and, implicitly, most influential, nodes. Similarly to the effect of NI on the BA model, a blanketing mechanism encourages the formation of C dominated neighbourhoods, which in turn generates a greater fitness than anything but very large values of individual investment θ . Interestingly, increasing individual investment at anything but low values of the influence threshold c_I actually promotes defection by enabling the temporary survival of cooperators connected to defectors which are centers of hubs. This, in turn, allows de-

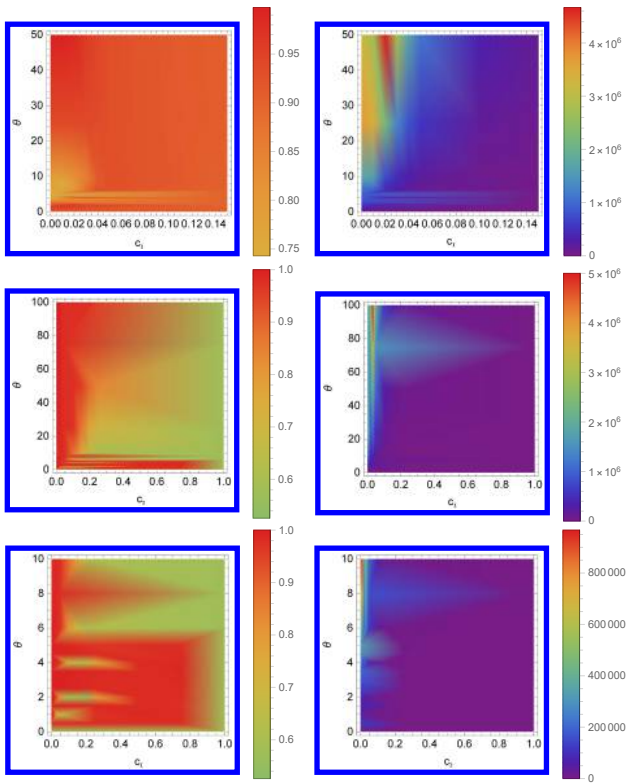


Figure 3: **Node influence based (NI) interference for BA model (top row) vs DMS model (middle row) and detailed view of DMS model (bottom row)**, for varying per-individual cost of investment θ , as well as the threshold for cooperator influence c_I . The left column reports the frequency of cooperation while the right one reports the total cost required. The ranges for θ and c_I are scaled for clear presentation. Parameters: $b = 1.8$; $n = 5000$; $z = 4$ (average node connectivity).

factor hubs to convert any remaining cooperator hubs. It is important to note that the initial distribution of players in the hubs is ultimately what determines which way the network will converge, so this type of interference does not produce any decrease in cooperation, as is the case of the BA model.

Local Cooperation

By the same token as earlier observations, interference on the BA model comes with the risk of reducing overall cooperation. What is more, LC based interference produces negative results for a wider range of parameters than any other mechanism (See Figure 4). That notwithstanding, investing smartly using the LC mechanism can lead to 100% cooperation, whereas the other mechanisms struggle. The key to such smart investments is in choosing a value for the threshold of local cooperation n_C which approaches the upper limit ($n_C \rightarrow 1$), with individual investment θ values high enough to convert defectors situated in cooperator clusters. As the value of n_C approaches 1, redundant investment decreases. With higher values of θ , the network converges

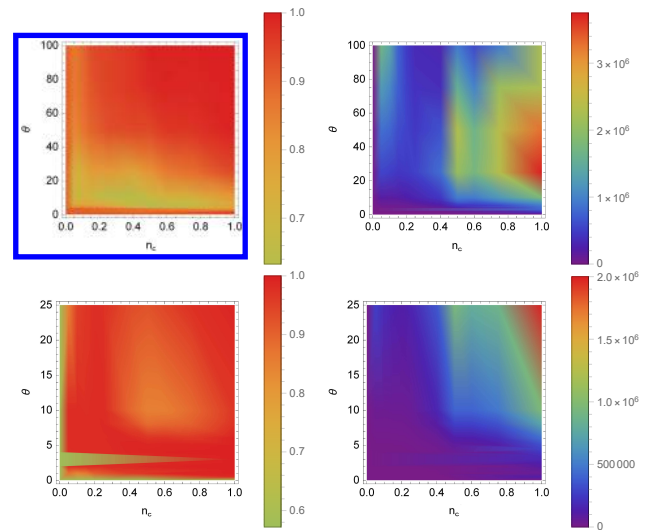


Figure 4: **Local cooperation based interference for BA model (top row) vs DMS model (bottom row)**, for varying per-individual cost of investment θ , scaled for clear presentation, as well as the threshold of neighbourhood cooperation n_C . The left column reports the frequency of cooperation while the right one reports the total cost required. Parameters: $b = 1.8$; $n = 5000$; $z = 4$ (average node connectivity).

more rapidly and therefore overall cost is reduced. Therefore, LC based interference can be regarded as the least risk averse, but potentially the most impactful given realistic values of per-individual investment θ .

With the exception of a small range of values for per-individual investment θ , the LC interference mechanism achieves a very high average cooperation for the clustering network. This reinforces the assumption that interference at any point in the decline of cooperation is enough to shift the scales and enables cooperators to overtake the defectors. On the basis thereof, an external decision maker can reduce the costs of interference by selecting very low values for n_C in combination with a high enough individual investment θ . For intermediate values of the local cooperation threshold n_C in combination with not high enough θ values, an interesting phenomenon is observed: the promotion of the survival of defectors by enabling the survival of cooperator nodes without actually giving them a chance of converting defectors in their neighbourhood, thereby allowing the defectors to exploit those cooperators.

Combining Node Influence and Local Cooperation

Due to the ease of acquiring information related to node connectivity in some types of networks, we test a combination of the two mechanisms where a cooperator node receives the individual investment only if both thresholds, local cooperation n_C and node influence c_I are met. Our results show that this is a risk-averse interference mechanism for low values of c_I (See Figure 5).

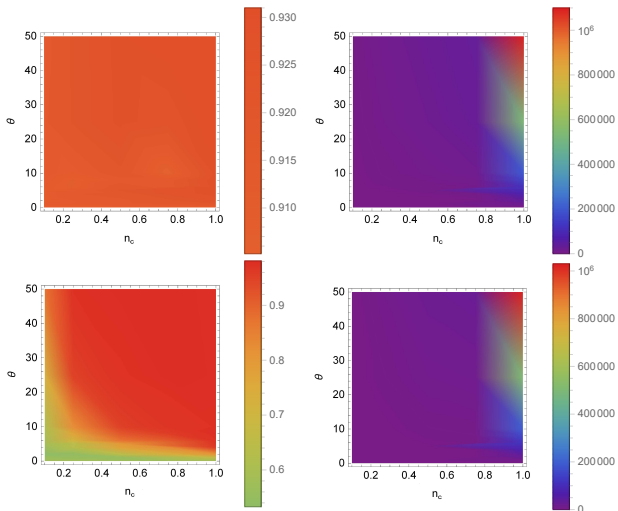


Figure 5: **Combination between node influence and local cooperation based interference for BA model (top row) vs DMS model (bottom row)**, for varying per-individual cost of investment θ and threshold of neighbourhood cooperation n_C , for $c_I = 0.05$. The left column reports the frequency of cooperation while the right one reports the total cost required. Parameters: $b = 1.8$; $n = 5000$; $z = 4$ (average node connectivity).

For the BA model, the possibility of inappropriate interference which leads to cyclic patterns is virtually eliminated even for very low values of c_I . We note that for very high values of per-individual investment θ , there is a marked increase in levels of cooperation while maintaining cost efficiency, if a high enough value for n_C is selected, similarly to our results for the LC-based mechanism. In that case, introducing the added parameter of node influence serves no purpose and reduces perceived gains to cooperation.

In the case of the SF NOC with a high prevalence of triangular motifs (DMS), the combination mechanism produces similar results to the ones observed for solely LC-based interference, but with slightly reduced costs across the range of parameter values and with a more predictable correlation between threshold values, per-individual investment cost and gains in cooperation. The maximum gains to levels of cooperation are reduced slightly when compared to the single two mechanisms, with the exception of very high θ .

Following these findings, we have shown that an integrated approach to interference would work best when the nature of the network is ambiguous. In that case, this type of interference would promote converge to cooperation in the case of the DMS model, without risking the decrease in cooperation seen in the BA Model for the two interference mechanisms applied independently.

Conclusions and Future Work

In summary, this paper aims to determine how best an external decision maker could incentivise a population of autonomous agents facing a cooperative dilemma to fulfil a

coveted collective state. We build on a previous account which identified the most effective mechanisms to foster cooperative scenarios in spatially distributed systems in regular graph structured populations of agents, but instead we consider two popular models of scale-free networks of contacts. In particular, we try to understand if the insights set out in the context of regular graphs remain applicable to heterogeneous models, as well as exploring an additional avenue of interference enabled by the variance in node connectivity. To address these issues, we have combined an evolutionary game theoretic model with several incentive mechanisms in two types of pre-generated networks characterised by preferential attachment, with different clustering coefficients. We argue that this problem cannot be solved trivially and we show that transitivity (i.e. the global clustering coefficient) should be the driving force behind the choice of an interference mechanism in promoting cooperation in heterogeneous network structures, as well as its application.

Our comparison between the two types of SF networks provides valuable insights regarding the importance of clustering in the outcome of cooperation. We found that a large clustering coefficient allows for successful, cost-effective interference, indeed even when disregarding a full comprehension of the population and its tendencies. These results are of particular interest, given that most SF networks portray high clustering, such as in the case of social ties where friends are likely to be friends of each other (Newman, 2018). Moreover, heterogeneous scenarios inhibited by spatial constraints (e.g. in highly urbanised areas or even the allotment of rangelands such as pastures) also impose some measure of clustering.

In the absence of clustering, we found that impetuously rewarding cooperators can lead to cyclic patterns which damage cooperation in the long run, and we show how this can be avoided when a decision maker lacks information about the level of clustering of the network. We observe a large negative impact on the cost of rewarding cooperators in the case of a low clustering coefficient, and provide insights on how it can be reduced. Moreover, we show that ignoring lowly connected individuals leads to unprofitable and even futile intervention irrespective of network transitivity.

Our future work aims to provide a comprehensive exploration of external interference on multiple types of networks while adopting different strategy update forms, such as stochastic learning (Szabó and Fath, 2007). We envisage that stochasticity will increase the overall cooperation and reduce the occurrence of cyclic patterns due to reckless interference, or eliminate them altogether. Furthermore, we plan to examine spatially-motivated interference mechanisms for heterogeneous networks, encouraging the formation of links between nodes or on the contrary, cutting off said links. The inherently high levels of cooperation in heterogeneous networks motivate us to experiment with a higher bias towards defection or mechanisms specifically

aimed at lowly connected nodes.

Acknowledgements

This work was supported by Future of Life Institute (grant RFP2-154) and by FCT-Portugal (grants UID/CEC/50021/2013, PTDC/EEI-SII/5081/2014, and PTDC/MAT/STA/3358/2014).

References

- Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA.
- Dorogovtsev, S. N., Mendes, J. F., and Samukhin, A. N. (2001). Size-dependent degree distribution of a scale-free growing network. *Physical Review E*, 63(6):062101.
- Franks, H., Griffiths, N., and Anand, S. S. (2014). Learning agent influence in mas with complex social networks. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 28(5):836–866.
- Franks, H., Griffiths, N., and Jhumka, A. (2013). Manipulating convention emergence using influencer agents. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 26(3):315–353.
- Han, T. A., Lynch, S., Tran-Thanh, L., and Santos, F. C. (2018). Fostering cooperation in structured populations through local and global interference strategies. In *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence '18*, pages 289–295. AAAI Press.
- Han, T. A., Pereira, L. M., and Lenaerts, T. (2019). Modelling and Influencing the AI Bidding War: A Research Agenda. In *AAAI/ACM conference AI, Ethics and Society*.
- Han, T. A., Pereira, L. M., Martinez-Vaquero, L. A., and Lenaerts, T. (2017). Centralized vs. personalized commitments and their influence on cooperation in group interactions. In *AAAI*, pages 2999–3005.
- Han, T. A., Pereira, L. M., and Santos, F. C. (2011). Intention recognition promotes the emergence of cooperation. *Adaptive Behavior*, 19(3):264–279.
- Han, T. A., Pereira, L. M., and Santos, F. C. (2012). Corpus-based intention recognition in cooperation dilemmas. *Artificial Life*, 18(4):365–383.
- Han, T. A. and Tran-Thanh, L. (2018). Cost-effective external interference for promoting the evolution of cooperation. *Scientific Reports*, 8:15997.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.
- Levin, S. A. (2000). Multiple scales and the maintenance of biodiversity. *Ecosystems*, 3(6):498–506.
- Newman, M. (2018). *Networks, 2nd edition*. Oxford university press.
- Nitschke, G. (2005). Emergence of cooperation: State of the art. *Artificial Life*, 11(3):367–396.
- Nowak, M. A. (2006). *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard University Press, Cambridge, MA.
- Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.
- Nowak, M. A. and Sigmund, K. (2005). Evolution of indirect reciprocity. *Nature*, 437(1291-1298).
- Paiva, A., Santos, F. P., and Santos, F. C. (2018). Engineering pro-sociality with autonomous agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7994–7999.
- Penn, A. S., Watson, R. A., Kraaijeveld, A., and Webb, J. (2010). Systems aikido—a novel approach to managing natural systems. In *in Proc. of the ALIFE XII Conference*, pages 577–580. MIT press.
- Raghunandan, M. A. and Subramanian, C. A. (2012). Sustaining cooperation on networks: an analytical study based on evolutionary game theory. In *AAMAS'12*, pages 913–920.
- Ramazi, P. and Cao, M. (2015). Analysis and control of strategic interactions in finite heterogeneous populations under best-response update rule. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4537–4542. IEEE.
- Rand, D. G., Nowak, M. A., Fowler, J. H., and Christakis, N. A. (2014). Static network structure can stabilize human cooperation. *Proc Natl Acad Sci USA*, 111(48):17093–17098.
- Riehl, J. R. and Cao, M. (2017). Towards optimal control of evolutionary games on networks. *IEEE Transactions on Automatic Control*, 62(1):458–462.
- Santos, F. C. and Pacheco, J. M. (2005). Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters*, 95:098104.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc Natl Acad Sci USA*, 103:3490–3494.
- Santos, F. C., Santos, M. D., and Pacheco, J. M. (2008). Social diversity promotes the emergence of cooperation in public goods games. *Nature*, 454(7201):213.
- Sigmund, K., De Silva, H., Traulsen, A., and Hauert, C. (2010). Social learning promotes institutions for governing the commons. *Nature*, 466(7308):861.
- Sigmund, K., Hauert, C., and Nowak, M. (2001). Reward and punishment. *Proc. Natl Acad Sci USA*, 98(19):10757–10762.
- Szabó, G. and Fath, G. (2007). Evolutionary games on graphs. *Physics Reports*, 446(4-6):97–216.
- Traulsen, A. and Nowak, M. A. (2006). Evolution of cooperation by multilevel selection. *Proc Natl Acad Sci USA*, 103(29):10952.
- Vasconcelos, V. V., Santos, F. C., and Pacheco, J. M. (2013). A bottom-up institutional approach to cooperative governance of risky commons. *Nature Climate Change*, 3(9):797–801.

Simplified Modeling of the Evolution of Skills in a Spatially Resolved Environment

Rudolf M. Fuchsli^{1,2}, Johannes J. Schneider¹, Thomas Ott³ and Richard Walker¹

¹Zurich University of Applied Sciences, School of Engineering, Winterthur, Switzerland

²European Centre for Living Technology, Venice, Italy

³Zurich University of Applied Sciences, School of Life Sciences, Wädenswil, Switzerland

rudolf.fuechslin@zhaw.ch

Abstract

We present a model for the spread, transmission and competition of skills with an emphasis on the role of spatial mobility of individuals. From a methodological point of view, we seek mathematical and computational simplicity in the sense of a minimal model. This minimalism lets us use an infinite dimensional simplex space and not a Euclidean space as underlying structure. Such a simplex captures the essentials of spatial heterogeneity without the mathematical difficulties of neighborhood structures.

In the presented model, individuals may have no skill or either skill A or B. Individuals are born unskilled and may acquire skills by learning from a skilled individual. Skill A results in a small reproductive advantage and is easy to transmit (teaching happens at high rate), whereas skill B is harder to teach but results in a high benefit. The model exhibits a rich behavior; after an initial transient, the system settles to a fix point (constant distribution of skills), whereby the distribution of skills depends on a mobility parameter m . We observe different regimes, and as the main result, we conclude that for some settings of the system parameters, the spread of the (harder to learn but more beneficial) skill B is only possible within a specific range of the mobility parameter.

From a technical point of view, this paper presents the application of the PRESS-method (probability reduced evolution of spatially resolved species) that enables the study of spatial effects in a very efficient manner. We analyze the consequences of spatial organization and argue that we can study aspects of social dynamics in an infinite dimensional simplex space. In spite of this maybe daunting name, the dynamics on such a structure is comparably easy to implement.

The model we present is far from reflecting all the details of human interaction. On the contrary, we deliberately tailored the model to be as simple as possible from a mathematical point of view (but still reflecting central properties of spatial organization). This approach is guided by physics, where seemingly simple models which obviously don't reflect the true physical behavior of a system (such as the Ising model) are nevertheless suited to reveal fundamental aspects and limiting cases of the real world.

Modeling Social Dynamics: Methodological Considerations

Studying social dynamics has to combine two core elements:

1. The interaction of individuals, once they meet.
2. The conditions and mechanisms that they meet.

Thereby, "to meet" implies the existence of an (implicit or explicit) binary function that determines whether two individuals are in contact. The notion of contact invokes aspects of space. At least, it requires a concept of "location" that enables to distinguish for a pair of individuals whether they are at the same location or not.

There are two main approaches for modeling and simulating social and societal dynamics: Agent-based [1-3] and density-based [4-6] simulations, the latter relying on systems of ordinary differential equations (ODE).

The agent-based simulation: The modeling consists in the precise determination of the interaction of individuals and the way, how and when this interaction takes place. The simulation platform translates these descriptions of interactions into actual encounters between individuals happening at rates resulting from the way how the individuals move. Thereby, "movements" have to be understood in a very general sense; the underlying space in which these movements take place can be an abstract one, for example a network.

Agent-based modeling exhibits a number of advantages.

- a. Simulating spatial effects is easy; agents just need a position attribute that represents their location in some form of spatial structure. Social interaction can easily be a function of these attributes.
- b. It is comparably easy to implement a combinatorial variety of interactions; an agent can be equipped with a table of attributes and interaction between two individuals can be determined as a function of the entries in the respective two attribute tables.
- c. From a technical perspective, it is no problem to introduce novel types of individuals (novel combinations of attributes) during a running simulation.
- d. Agent-based simulations exhibit fluctuations; this can be of advantage if one wants to include fluctuations into the simulation.

However, there are a number of challenges:

- a. Agent-based simulations scale quadratically (or with some other exponent larger than 1) with the number of individuals involved, e.g., if each individual gets the chance to interact with any other individual [7-9].
- b. Agent-based simulations exhibit fluctuations; this can be a disadvantage if one is interested in average values. Then, large numbers of agents are required, resulting in an according amount of computing time.

An alternative are approaches based on systems of ordinary differential equations: in such studies, one treats social dynamics in a way similar to chemical reaction kinetics; to be precise: chemical reaction kinetics in a well-stirred pot, means a homogeneous reaction environment.

Advantages of the ODE approach are:

- a. They work (at least implicitly) with continuous densities. No fluctuations due to discretization effects occur. In other words: One always works in the limit of infinitely many individuals.
- b. In rare cases (usually for the limits of some parameter values), analytical solutions can be obtained. To be precise, this happens very rarely for the study of dynamics but more often for the study of fixed points.

Some disadvantages balance these benefits:

- a. In a well-stirred reaction environment, there is no notion of space. As soon as the probability for the contact of two individuals requires a more involved conceptual framework than that of a collision frequency in chemical kinetics, ODE approaches become quite often somewhat contrived.
- b. ODE approaches exhibit unfavorable scaling behavior if one considers bigger numbers of different types of individuals.
- c. It is technically very hard to change the number of types during a running simulation (one has to construct a new set of ODE).

In what follows, we describe an approach that is based on systems of ordinary differential equations and nevertheless captures essential aspects of spatial structuring in a natural way, the so-called probability reduced evolution of spatially resolved species approach (PRESS). The method is based on an idea of John S. McCaskill and has been described in a series of publications with emphasis on evolutionary dynamics in molecular biology [10-12].

The PRESS Approach

The PRESS approach assumes a network of locations. Interactions happen in the locations. The locations are connected and migration may take place from one location to another one. The number of sites in a location is finite and fixed and the same for all locations. Also limited is the number of different types of individuals. In consequence, a location can only attain a finite number of states. The PRESS approach assumes the network of locations to be a simplex (each location has the same number of sites and is connected to all the other locations). This inherent symmetry implies that the probability for a location to be in a specific state is well-defined and the same for all locations. The goal of the PRESS approach is to compute the dynamics of these probabilities. From a mathematical perspective (since all the locations are fundamentally equal and each location is connected to all the other locations), migration can be expressed in terms of a mean field approach (as will be detailed in what follows).

Why using a simplex? The geometrical concept “space” carries a surprisingly rich variety of mathematical structure; most fundamentally, a notion of space means that one can distinguish between here and there. In addition, spaces such as our familiar three-dimensional Euclidean space bears structure that allows to quantify the “theres” (means “non-heres”) by a notion of

distance. The “theres” can be grouped in locations that are far or nearer to the “here”. Furthermore, the notion of space, especially if one considers nontrivial spaces, always contains connections, that means paths from a “here” to a “there”. These connections are trivial in the case of Euclidean spaces, but become more interesting in the spaces studied in higher mathematics as well as in those occurring in everyday life (e.g. the road system that connects different cities makes it a nontrivial decision of how to get from Lausanne to Zurich.)

We mentioned above that simulations based on systems of ordinary differential equations most often relate to models in which we make the assumption of a “well – stirred reaction environment”. This means nothing else then the complete absence of spatial structure. Despite its daunting name, an infinite-dimensional simplex is probably the smallest possible step from a completely homogeneous reaction environment towards a model that captures at least the most fundamental aspects of space. For our purposes, a simplex is nothing more than a set of n discrete locations which are all mutually connected in an identical manner (s. $n = 4, 5, 6$ – simplices in Fig. 1.) An infinite dimensional simplex is then the limiting case of large numbers of locations [13]. The question “Why a simplex?” is answered by noting that it is the simplest structure representing some sort of spatial heterogeneity but without all the computational challenges resulting from heterogeneous neighborhood structures (two locations may be in different states, but on simplex, they all have the same neighborhood, namely all the other locations).

In a social context, such a simplex can be interpreted as a densely packed village that consists of identical, small huts. These huts are the places where the interaction between individuals take place, see Fig. 2. Since we model only unary and binary interactions, locations which can host up to maximally two reaction partners constitute a minimal reaction environment. The goal of the model presented is then to compute the probability for each possible occupation state.

In our model, we make the following assumptions:

- a. A hut can host up to two individuals.
- b. Individuals only interact inside huts.
- c. There may be different types of individuals; the interaction between two individuals is a function of their respective types.
- d. The individuals commute freely between huts. If an individual leaves a hut, it chooses a different hut to enter at random.
- e. The time needed for the passage between huts is negligible.

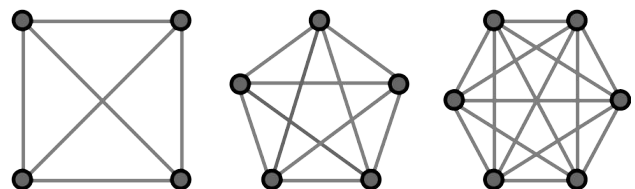


Fig. 1: Simplices for $n = 4, 5, 6$.

Besides the fact that in our model the village is highly symmetric, we introduce a further idealization. We model the village in the limit of a very large number of huts. The rational

for his approach is given by the fact that we are interested in the number of huts with a specific occupation, i.e. a specific combination of individuals (say, the number of huts occupied by only one individual of the blue type or the number of huts hosting a blue and a red individual). Knowing the numbers of huts for each possible occupation, we can easily compute the total number of individuals of a given type.

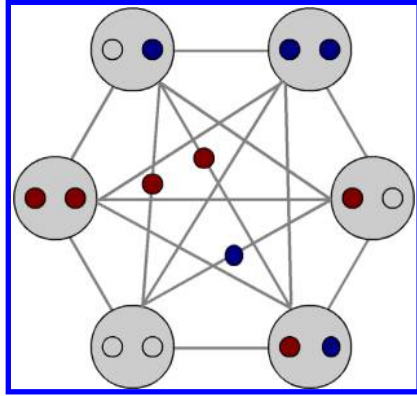


Fig. 2: A simplex built up from locations, which can host up to two individuals.

For a finite number n of huts, the number n_X of huts with a specific occupation X is a discrete value. Moreover, if the interaction between individuals and the passage between huts is driven by random events, n_X will be subject to fluctuations. Instead of n_X , we will compute the probability p_X for a hut to have an occupation X ; it simply holds

$$p_X = \frac{n_X}{n} \quad (1)$$

In the limit of large n , p_X will become a continuous variable. In addition, p_X will not exhibit fluctuations (without a detailed proof and referring to the central limit theorem, we assume the fluctuations Δn_X of n_X to scale in a sub-linear way with n_X .

The fraction $\frac{\Delta n_X}{n_X}$ will then vanish for large n).

In what follows, we will set up the dynamics of the probabilities for the various possible occupations of the huts.

This approach, which will be exemplified in the next section, has already been introduced in the context of chemical reaction kinetics [10-12]. The authors called it the PRESS – approach, for Probability Reduced Evolution of Spatially resolved Species. It mimics space in so far, as there is a distinction between different locations and as there is a parameter, namely a rate constant m for the passage from one hut to the other. The mobility m can be understood as the analogue to the diffusion constant in Euclidean space. It is also a parameter that can be interpreted in terms familiar to the social sciences as a type of migration frequency.

Obviously, our modeling framework represents a highly idealized situation. Whether the metaphor of a hut is appropriate may be discussed. Certainly, the restriction to a maximal occupation number of two is “unrealistic” in the sense that real huts can easily be larger. Understanding the huts as interaction sites enables a more abstract view. A restriction to size two can be justified in various ways. One way is the already mentioned perspective on the maximal number of partners in the modeled social interactions. As will be discussed below, this is two in our case. Another way of looking at this restriction is coarse graining. The interaction locations can be in a number of different states and we compute their respective probabilities based on a set of interactions. Assuming locations with two sites spans the minimal set of states necessary for representing the transitions we want to model. Further idealizations refer to the various symmetries invoked in our model. In a real village, huts will not be identical, interactions between individuals will not only take place inside huts, the huts will belong to somebody and not be chosen randomly and the distance between the huts will play a role for the probability of a transfer from hut A to hut B.

Furthermore, a simplex needs not necessarily to be understood as a village consisting of huts. One may understand the individual interaction sites as caricatures of villages themselves. The whole simplex then is a collection of interacting small villages. The occupation of a village may represent its internal state. Transport between interaction sites can be understood as information flows between isolated, but weakly interacting groups of human beings.

Modeling the Competition of Skills

Our goal is to study the question under which conditions (means for which parameter values) different skills are spread and maintained in populations. Thereby, a skill is a property of an individual that a) gives the individual some benefit and b) has to be transferred from a skilled to an unskilled individual by some form of teaching process. We will study a population in which individuals can learn one of two skills. These two skills differ first in the benefit they provide to dose individual having the particular skill and second in the effort (time) it takes to teach the skill to an unskilled individual U. In the model, there is an easy-to-learn skill A that yields only limited benefit and a skill B that is hard to learn but results in a higher benefit. The question we want to address is the following: What is the dependency of the average number of individuals with skill A, B or U (skill U means unskilled) one encounters in the randomly chosen hut as a function of the mobility parameter m ? This question arose from broader investigations about cultural evolution performed by one of the authors, Richard Walker. As it turns out, the dynamics of the model delivers a surprisingly faceted answer.

The Model

The Possible Occupations

We study huts that can host maximally two individuals. Larger huts are straightforward to implement, but we want to keep the mathematical formalism as simple as possible. The symbol “V” stands for void, “U” for unskilled, “A” for individual with skill A and “B” represents an individual with skill B. The occupation or state of a hut is described by an index consisting of three non-negative integers UAB , such that $U + A + B \leq 2$. The total number T of individuals in a state UAB is given by $T = U + A + B$ and the according number of voids by $V = 2 - U - A - B$. The index set $I_1 = \{000, 100, 010, 001, 200, 110, 101, 020, 011, 002\}$ contains all allowed indices for huts of size two.

Sometimes, an alternative indexation is useful, referring to the two available vacancies in a hut. We define an index set $I_2 = \{VV, VU, VA, VB, UU, UA, UB, AA, AB, BB\}$. In this notation, a potential problem becomes immediately apparent, namely the occurrence of symmetric states. In this work, we do not distinguish between occupations AB and BA.

For each occupation UAB , there is a time-dependent probability $p_{UAB}(t)$. We formulate a Master equation

$$\frac{d\vec{P}}{dt} = M(\vec{P}(t))\vec{P}, \quad \vec{P} = \begin{pmatrix} p_{000}(t) \\ \vdots \\ p_{002}(t) \end{pmatrix} \quad (2)$$

in order to determine the time development of the occupations. Thereby, $M(\vec{P}(t))$ is a matrix that depends on $\vec{P}(t)$ (and therefore implicitly on time) and represents the different interactions transforming one state into another. In what follows, we will discuss these interactions and the present their contribution to $M(\vec{P}(t))$.

Interactions and Processes

In our model, four types of processes can occur:

- Individuals may give birth to children. Children are always unskilled.
- An individual with skill X can teach an unskilled individual.
- Individuals can die.
- Individuals can leave a hut and enter another one.

In what follows, we provide the rate equations for the probabilities $p_{UAB}(t)$ for each of these processes. We will formulate template formulas. In these template formulas, we assume that $p_{UAB}(t) \equiv 0$ if the index UAB does not satisfy the requirement that all the integers of which the index is composed are bigger or equal to zero or that the sum $U + A + B$ is bigger than two. To facilitate the notation further, we introduce the function ε :

$$\varepsilon(U, A, B) = \begin{cases} 1, & U, A, B \geq 0, U + A + B \leq 2 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Birth

The birth rate of individuals with skill X is given by β_X , $X \in \{U, A, B\}$ (we don't model the details of sexual reproduction). Different skills may lead to different birthrates, which in turn constitute evolutionary advantages or disadvantages.

The probability $p_{UAB}(t)$ will change by births according to the following template formula:

$$\left. \frac{dp_{UAB}}{dt} \right|_{\text{birth}} = (U-1)\beta_U p_{(U-1)AB} + A\beta_A p_{(U-1)AB} + B\beta_B p_{(U-1)AB} - \sum_{X \in \{U, A, B\}} X \varepsilon(U+1, A, B) \beta_X p_{UAB} \quad (4)$$

Let us analyze this formula. The term p_{UAB} gives the probability for the state with U unskilled inhabitants, A inhabitants with skill A and B inhabitants with skill B. The first term on the right hand side, $(U-1)\beta_U p_{(U-1)AB}$ models the birth of an unskilled individual from an unskilled parent with a birth rate β_U . Because birth increments the number of unskilled individuals by one, the original state is $p_{(U-1)AB}$. The factor $(U-1)$ takes into account the possibility of many potential parents (which is not important if one restricts the maximal occupation number to two, but the template can be applied for larger huts, too.) The second term $A\beta_A p_{(U-1)AB}$ represents the birth of an unskilled inhabitant by a parent with skill A, and accordingly for the third term and skill B. The last term models the decrease of the probability p_{UAB} caused by birth processes (Note that in our model, we could replace $\varepsilon(U+1, A, B)$ by $(2-U-A-B)$, but the variant we use is also valid, if we run the model with larger huts.) A remark: the fact that the number of vacancies in a hut is restricted imposes an implicit resource that limits growth.

Death

The template formula for the death of individuals is structurally identical to the one of birth. It reads:

$$\left. \frac{dp_{UAB}}{dt} \right|_{\text{death}} = (U+1)\delta_U p_{(U+1)AB} + (A+1)\delta_A p_{U(A+1)B} + (B+1)\delta_B p_{UA(B+1)} - \sum_{X \in \{U, A, B\}} X \delta_X p_{UAB} \quad (5)$$

The death rates, which may be different for differently skilled individuals, are given by δ_X .

Teaching

We assume that if an unskilled individual resides together with an individual with skill X in a hut, a teaching process may occur. The according template formula reads:

$$\begin{aligned} \left. \frac{dp_{UAB}}{dt} \right|_{teach} &= (U+1)(A-1)\tau_A p_{(U+1)(A-1)B} \\ &+ (U+1)(B-1)\tau_B p_{(U+1)A(B-1)} \\ &- UA\tau_A p_{UAB} - UB\tau_B p_{UAB} \end{aligned} \quad (6)$$

The first line represents the transformation of an unskilled individual into one with skill A (the unskilled individual is taught by an individual with skill A). This transformation, the “teaching rate”, is given by τ_A . Accordingly, the second line models the teaching of skill B with teaching rate τ_B . Finally, the last line represents transformations out of state UAB . Remark: this template is a bit more complicated than it needs to be, at least as long as we restrict the number of vacancies in a hut to two. However, the formula as it stands can be applicable for larger huts.

Diffusion

Modeling diffusion is conceptually a little bit more difficult. First, we have to note that we assume an individual leaving a hut and entering immediately into another one. One can think of an individual choosing a target hut at random and make a passage to this hut as long as this hut is not already full (because we work on an infinite dimensional simplex, the case of leaving a hut and entering it again does not require specific care). The an individual with skill X fills a vacancy in a given hut is then proportional to the average number of individuals with that skill in all the other huts. As a consequence, the chance of an individual with skill X in a given hut to leave this hut is proportional to the average number of vacancies in all the other huts.

These averages are easy to compute; the advantage of an infinite dimensional simplex as underlying space is that the occupation probabilities in all the hearts are identical. Therefore, we have:

$$\begin{aligned} \bar{U} &= \sum_{i \in I_1} u(i) p_i \\ \bar{A} &= \sum_{i \in I_1} a(i) p_i \\ \bar{B} &= \sum_{i \in I_1} b(i) p_i \\ \bar{V} &= \sum_{i \in I_1} v(i) p_i \end{aligned} \quad (7)$$

Thereby, i is an index of the form UAB and the functions u, a, b are defined as $u(i = UAB) = U$, $a(i = UAB) = A$, $b(i = UAB) = B$ and $v(i = UAB) = 2 - U - A - B$.

The template formula for in – diffusion (state transitions that increase the number of inhabitants) reads:

$$\begin{aligned} \left. \frac{dp_{UAB}}{dt} \right|_{in-diff} &= (2 - (U - 1) - A - B)\bar{U} m p_{(U-1)AB} \\ &+ (2 - U - (A - 1) - B)\bar{A} m p_{U(A-1)B} \\ &+ (2 - U - A - (B - 1))\bar{B} m p_{UA(B-1)} \\ &- \sum_{X \in \{U, A, B\}} (2 - U - A - B)\bar{X} m p_{UAB} \end{aligned} \quad (8)$$

Thereby, the parameter m models the mobility and the last line takes into account that the influx of an inhabitant into a state UAB reduces p_{UAB} .

Accordingly, for out-diffusion (state transitions that decrease the number of inhabitants), we have:

$$\begin{aligned} \left. \frac{dp_{UAB}}{dt} \right|_{out-diff} &= (U+1)\bar{V} m p_{(U+1)AB} + (A+1)\bar{V} m p_{U(A+1)B} \\ &+ (B+1)\bar{V} m p_{UA(B+1)} - \sum_{X \in \{U, A, B\}} X \bar{V} m p_{UAB} \end{aligned} \quad (9)$$

The complete Master equation is given by:

$$\begin{aligned} \frac{dp_{UAB}}{dt} &= \left. \frac{dp_{UAB}}{dt} \right|_{birth} + \left. \frac{dp_{UAB}}{dt} \right|_{death} + \left. \frac{dp_{UAB}}{dt} \right|_{teach} \\ &+ \left. \frac{dp_{UAB}}{dt} \right|_{in-diff} + \left. \frac{dp_{UAB}}{dt} \right|_{out-diff} \end{aligned} \quad (10)$$

Results

We solved the Master equation Eq. (10) by standard numerical procedures. As it turned out, the solutions settled down to a fixed point after a sufficiently long time development. Of course, many parameter values lead to rather uninteresting behavior. However, some showed a rather rich behavior. In what follows, we analyze the system for the following set of parameters:

$$\begin{aligned}
\beta_U &= 1/10 \\
\beta_A &= 1/5 \\
\beta_B &= 1/2.5 \\
\delta_U &= \delta_A = \delta_B = 1/80 \\
\tau_A &= 1/5 \\
\tau_B &= 1/10
\end{aligned}
\tag{11}$$

If one takes the unit of time as one year, one sees that the average life span of an individual is 80 yrs. Learning a skill takes 5 years (skill A) or ten years (skill B). The birth rates look much larger as they really are; take into account that the number of births is limited (space is a limited resource).

For this choice of parameters, we observe a rich variety of possible fix points. In Fig. 3, we distinguish seven different patterns of behavior with respect to the mobility parameter m . Shown are the values the averages of skills U, A, B attained after 10^6 time units. The initial conditions are set to $\bar{V}(0) = 1, \bar{U}(0) = 0.05, \bar{A}(0) = \bar{B}(0) = 0.225$. The system turned out to be insusceptible towards the initial conditions, provided that there is at least some density of the skills A and B.

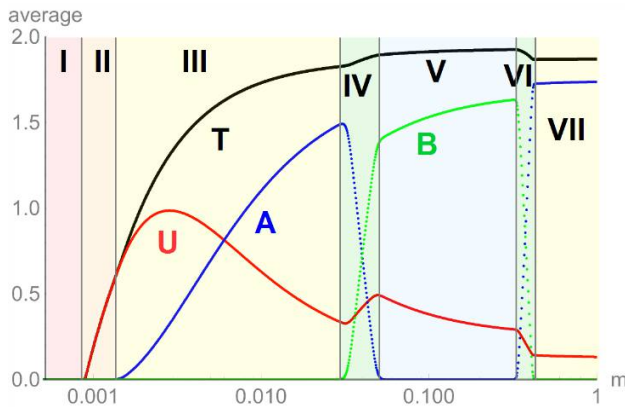


Fig. 3: Average number of skilled individuals per location of interaction as a function of the migration parameter m .

In Fig. 3, the y-axis reflects the expected average density of a given skill as a function of the mobility parameter m . U represents the density of unskilled individuals, A and B those of individuals with the respective skill. T stands for the total of individuals. At very low m (region I), the population can't survive. For low m (region II), no skill can be maintained. For higher values of m , we observe either that one skill dominates the other or co-existence of skills. As a main result, we observe that a population can maintain a complex skill B only in a window of the mobility parameter. In order to uphold a complex skill, a certain mobility is necessary, but too much mobility favors simpler skills that are easier to transmit.

Discussion

The outcome of our simulations shows that maintaining skills in a population may depend on migration rates in a rather subtle manner. However, we are very reluctant giving these simulations (or the parameter values at which changes in behavior of the system occur) a lot of direct relevance. And of course, we are far from modeling the real processes in prehistoric societies. But physics teaches us that seemingly simple models which obviously don't reflect the true physical behavior of a system (such as the Ising model) are nevertheless suited to reveal fundamental aspects and limiting cases of the real world. In that sense, we are convinced that a PRESS model based approach to social dynamics can help us understand and guide our search for interactions, which lead to interesting system behavior.

An example for such a support is a question that results from the observations in Fig. 3. Basically, Fig. 3 states a complex dependence of the system behavior on the migration rate. But in the presented model, a migration process only takes place if a randomly chosen hut has at least one empty site. With other words: migration is limited by the number of free sites and there is no spread of information different from the migration of individuals. However, one can easily imagine a situation in which the flow of information is much faster than migration of individuals. This could e.g. lead to a migration pattern where potential students (unskilled individuals) know in advance which hut harbors empty sites. Technically, this means that migration stays maximal until the system is full.

Investigations based on the PRESS model should (and will) be complemented by agent based simulations. In an agent-based simulation, it is very easy to study the effect of a fast flow of information.

Acknowledgements

This work has been partially supported by the European Horizon 2020 project ACDC – Artificial Cells with Distributed Cores to Decipher Protein Function “, grant agreement No 824060”. Relevant parts of the mathematical framework developed in ACDC have been used in this work.

References

1. S. Moss de Oliveira, P. M. C. de Oliveira, and D. Stauffer, *Evolution, Money, War, and Computers*, Teubner, Stuttgart, Leipzig, 1999, ISBN 978-3-519-00279-6.
2. J. J. Schneider and C. Hirtreiter, The Impact of Election Results on the Member Numbers of the Large Parties in Bavaria and Germany, *Int. J. Mod. Phys. C* Vol. 16, 1165-1215, 2005.
3. S. Kirkpatrick and J. J. Schneider, How Smart Does an Agent Need to Be?, *Int. J. Mod. Phys. C* Vol. 16, 139-155, 2005.
4. R. Hesani, A. Ghorbani, and V. Dignum, Using Difference Equation to Model Discrete-time Behavior in System Dynamics Modeling, 32nd International Conference of the System Dynamics Society, Delft, The Netherlands, 2014, <https://pdfs.semanticscholar.org/0832/553cfa88aa069fb3335cb040aac77997e1f0.pdf>.

5. C. M. Brown, *Differential Equations – A Modeling Approach*, Series Quantitative Applications in the Social Sciences Vol. 150, Sage Publishing, Los Angeles, 2007, ISBN 9781412941082.
6. H. Van Dyke Parunak, R. Savit, and R. L. Riolo, *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*, International Workshop on Multi-Agent Systems and Agent-Based Simulation, MABS, Lecture Notes in Artificial Intelligence Vol. 1534, 10-25, Springer, 1998.
7. A. Ebersbach, J. Schneider, I. Morgenstern, and R. Hammwöhner, *The Influence of Trucks on Traffic Flow – An Investigation on the Nagel-Schreckenberg-model*, Int. J. Mod. Phys. C Vol. 11, 837-842, 2000.
8. J. J. Schneider and C. Hirtreiter, *Scaling Laws for the Lifetimes of Governments in the Sznajd Democracy*, Int. J. Mod. Phys. C Vol. 16, 157-165, 2005.
9. J. J. Schneider and C. Hirtreiter, *The Democracy-Ochlocracy-Dictatorship Transition in the Sznajd Model and in the Ising Model*, Physica A Vol 353, 539-554, 2005.
10. J. S. McCaskill, R. M. Fuchsli, and S. Altmeyer, *The stochastic evolution of catalysts in spatially resolved molecular systems*, Biol. Chem. Vol. 382, No. 9, 1343-1363, 2001.
11. R. M. Fuchsli, S. Altmeyer, and J. S. McCaskill, *Evolutionary stabilization of generous replicases by complex formation*, Europ. Phys J. B Vol. 1, 103-110, 2004.
12. S. Altmeyer, R. M. Fuchsli, and J. S. McCaskill, *Folding stabilizes the evolution of catalysts*, Artificial Life Vol. 10, 23-38, 2004.
13. <https://mathworld.wolfram.com/CompleteGraph.html>

Promoting Cooperation through External Interference

The Anh Han¹, Long Tran-Thanh², Simon Lynch¹, Theodor Cimpanu¹, Francisco C. Santos³

¹School of Computing, Media and the Arts, Teesside University

²School of Electronics and Computer Science, University of Southampton

³INESC-ID and Instituto Superior Tecnico, Universidade de Lisboa

Email: t.han@tees.ac.uk

The problem of promoting the evolution of cooperative behaviour within populations of self-regarding individuals has been intensively investigated across diverse fields of behavioural, social and computational sciences (Nowak, 2006; Perc et al., 2017). In most studies, cooperation is assumed to emerge from the combined actions of participating individuals within the populations, without taking into account the possibility of external interference and how it can be performed in a cost-efficient way. However, in many scenarios, cooperative behaviours are advocated and promoted by an exogenous decision maker, who is not part of the system (e.g. the United Nation interferes in political systems for conflict resolution or the World Wildlife Fund organisation interferes in ecosystems to maintain biodiversity). Thus, a new set of heuristics capable of engineering a desired collective behaviour in a self-organised multiagent system is required. Here we summarize our recent works to bridge this gap, in which we employ theoretical analysis and computer simulations based on evolutionary game theory (Nowak, 2006), to study cost-efficient interference strategies for enhancing cooperation in the context of cooperation dilemma games, for both well-mixed (Han and Tran-Thanh, 2018) and square-lattice structured populations (Han et al., 2018).

We consider finite populations (of size N) of individuals who interact with each other through the one-shot Prisoner's Dilemma game (PD) (Nowak, 2006), where in each interaction two players simultaneously choose either to cooperate (C) or defect (D). Mutual cooperation (mutual defection) yields the reward R (penalty P) and unilateral cooperation gives the cooperator the sucker's payoff S while the defector the temptation T . A PD is characterized by the ordering $T > R > P > S$. In a well-mixed population, each player interacts with all others in the population while in a square lattice the player interacts with its four immediate neighbors. A player's fitness is its averaged payoff over all its interactions, which is then used for strategy update. Namely, a player A with fitness f_A chooses to copy the strategy of a randomly selected player in the population (well-mixed) or randomly selected neighbor (structured) with a probability

given by the Fermi function, $(1 + e^{\beta(f_A - f_B)})^{-1}$, where β represents the intensity of selection (Traulsen et al., 2006). When $\beta = 0$ corresponds to neutral drift while $\beta \rightarrow \infty$ leads to increasingly deterministic selection. Weak or even close to neutral selections (small β) are abundant in nature, while the strong selection regime has been reported as predominant in social settings. As an alternative to this stochastic update rule, one can also consider a *deterministic update* in which agents copy, if advantageous, the most successful player in their neighbourhood.

An interference strategy or scheme can be generally defined as a sequence of decisions about which C players in the population to invest in (i.e. reward the player an amount, denoted by θ), in order to achieve the highest level of cooperation while minimising the total cost of investment. These decisions can be made by considering different aspects of the population such as its global statistics and/or its structural properties. In the context of a well-mixed population, an interference scheme solely depends on its composition (i.e. how many C and D players there are at the time of decision making). In this case, we have derived analytical conditions for which a general interference scheme can guarantee a given level of cooperation while at the same time minimising the total cost of investment (for rewarding cooperative behaviours), and show that the results are highly sensitive to the intensity of selection by interference. Moreover, we have studied a specific class of interference strategies that make investments whenever the number of C players reaches a certain threshold, denoted by t ($\forall t \in \{1, \dots, N-1\}$), showing that there is a wide range of t that it outperforms standard institutional incentive strategies—which unconditionally interfere into the system regardless of its composition, corresponding to $t = N-1$ (Chen et al., 2015). Figure 1a shows the optimal threshold of t for varying the intensity of selection, β , where the minimal expected cost of interference is obtained while guaranteeing at least ω -frequency ($0 \leq \omega \leq 1$) of cooperation. We can observe that, when β is sufficiently small (weak selection), an intermediate value of t would lead to most cost-efficient interference strategies, while for a sufficiently strong selection, it is best to

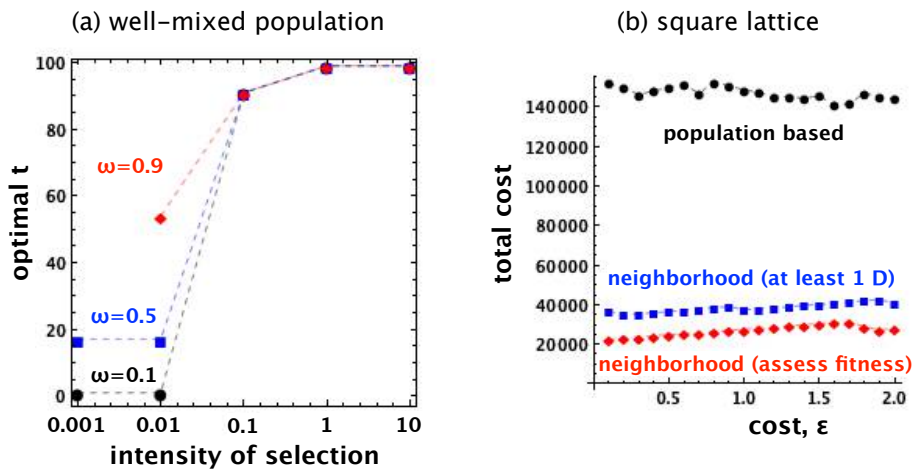


Figure 1: **a)** In a well-mixed population (size $N = 100$), the optimal threshold of t where a minimal expected cost of interference is obtained while guaranteeing at least ω -frequency of cooperation, which increases with the intensity of selection (Other parameters: $T = 2$; $R = 1$; $P = 0$; $S = -1$; $\theta = 5$); **b)** In a square lattice population (size $N = 100^2$), neighborhood-based (local) interference strategies are more cost efficient in ensuring high cooperation ($\approx 100\%$) than population-based (population cooperation level) ones. Other parameters: $T = 1.8$; $R = 1$; $P = 0$; $S = 0$ (weak PD); deterministic update was used but results are robust for stochastic update (Han et al., 2018).

always interfere, meaning that standard institutional incentive strategies (i.e. $t = N - 1$) would be most effective.

With a structured population, individuals (even of the same strategy) might reside in different kinds of neighborhood (with different cooperativeness levels), and therein local information might be useful to enhance cost-efficiency and cooperation. To this end, we test several interference paradigms (Han et al., 2018) that make investment decisions based on a player’s current cooperativeness level (the number of C players in the neighborhood), and compared their efficiency with the population-based strategies (as in the well-mixed case). Our systematic analysis reveals a simple strategy that invests when there is at least one D player in the neighborhood and does not invest otherwise, is highly cost-efficient in promoting cooperation (see Figure 1b). Furthermore, when additional information regarding the fitness levels (i.e. individual income information) of players in a neighbourhood is accessible, further improvement can be made by more accurately influencing D neighbours for behavioural change (to become cooperators).

Future works include analysis of other types of population structures such as the scale-free ones and their variants (Cimpeanu et al., 2019) and more complex interference strategies such as those vary the cost of investment over time or combine different forms of incentives (Chen et al., 2015; Han, 2016).

In short, we have studied how cooperation can be promoted in a cost-efficient way from an external decision maker’s perspective. It provides new insights regarding heuristics capable of engineering a desired collective behaviour in a self-organised complex system, not only in social and biological contexts, but also Artificial Life scenarios

such as swarm-based and multi-robots systems (Bonabeau et al., 1999; Han et al., 2012).

References

- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford U. Press.
- Chen, X., Sasaki, T., Brännström, Å., and Dieckmann, U. (2015). First carrot, then stick: how the adaptive hybridization of incentives promotes cooperation. *Journal of The Royal Society Interface*, 12(102):20140935.
- Cimpeanu, T., Han, T. A., and Santos, F. C. (2019). Emergence of coordination with asymmetric benefits via prior commitment. In *ALIFE (in press)*.
- Han, T. A. (2016). Emergence of social punishment and cooperation through prior commitments. In *AAAI*, pages 2494–2500.
- Han, T. A., Lynch, S., Tran-Thanh, L., and Santos, F. C. (2018). Fostering cooperation in structured populations through local and global interference strategies. In *IJCAI-ECAL*, pages 289–295. AAAI Press.
- Han, T. A., Pereira, L. M., and Santos, F. C. (2012). Corpus-based intention recognition in cooperation dilemmas. *Artificial Life journal*, 18(4):365–383.
- Han, T. A. and Tran-Thanh, L. (2018). Cost-effective external interference for promoting the evolution of cooperation. *Scientific reports*, 8.
- Nowak, M. A. (2006). *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard U. Press, Cambridge.
- Perc, M., Jordan, J. J., Rand, D. G., Wang, Z., Boccaletti, S., and Szolnoki, A. (2017). Statistical physics of human cooperation. *Physics Reports*, 687:1–51.
- Traulsen, A., Nowak, M. A., and Pacheco, J. M. (2006). Stochastic dynamics of invasion and fixation. *Phys. Rev. E*, 74:11909.

Factors that Affect the Evolution of Complex Cooperative Behavior

Padmini Rajagopalan¹, Kay E. Holekamp² and Risto Miikkulainen¹

¹The University of Texas at Austin, Austin, TX 78712

²Michigan State University, East Lansing, MI 48824
padmini@cs.utexas.edu

Abstract

Collaboration in order to perform various tasks such as herding or hunting is frequently seen in nature. Cooperative behaviors benefit the group by helping them achieve rewards that would not be possible for an individual to achieve alone. In addition to cooperative hunting, spotted hyenas also participate in coordinated mobbing of lions, which is a complex behavior that is still believed to be genetic. Lions are larger and stronger than hyenas, and therefore the hyenas need to cooperate in large numbers to overcome their fear and attack the lions. Individualistic hyena traits and other factors that may affect the frequency or success of lion-mobbing have not been studied in simulation before. Furthermore, multiple emotions, such as fear and affiliation towards teammates, affect the willingness of hyenas to attack lions. The computational model of lion-hyena interaction developed in this work can help understand the evolution of mobbing behaviors. It may be used in the future to evolve strategies in video game characters to overcome powerful adversaries or solve problems that involve high risk.

Introduction

Complex cooperative behaviors are hard to model in simulation, whether they be hard-coded or learned. In previous work, collaboration had to be evolved through coevolution, or other such means (Uchibe and Asada, 2006; Yong and Miikkulainen, 2009; Rawal et al., 2010). It is, therefore, helpful to examine how cooperative behaviors emerge in nature, and what factors influence their successful evolution.

In nature, spotted hyenas frequently cooperate in teams in order to hunt for prey that is difficult to kill (Kruuk, 1972; Holekamp et al., 1997). This behavior has previously been modeled in simulation (Rajagopalan et al., 2011).

Less frequently, hyenas also gather in large numbers to attack lions and drive them away in order to gain possession of a kill. This lion-mobbing behavior is very complex and requires precise coordination to succeed. This is because lions are larger and stronger than hyenas and, therefore, are expected to emerge the winners in any lion-hyena interaction.

There are some limitations to the study of lion-hyena interactions in nature. The path that hyena behavior evolution took to reach its current state of sophisticated mobbing

cannot be studied in real-life hyenas. This problem can be solved by developing a computational model that faithfully reproduces lion-hyena interactions and mobbing behaviors from nature. It can then be used to study the evolution of such behaviors as well as to make predictions about them. This paper describes how such a model was built.

Neuroevolutionary techniques were used to control a team of hyenas that were placed in various situations along with simulated lions in a field. The simulations showed that the factors observed in nature as affecting the evolution and success of mobbing behaviors also emerged in the computational model. In the future, these principles can also be used to build teams of artificial agents with complex cooperative behaviors.

Related Work

This section will first describe the biological background of lion-mobbing, after which the modeling of cooperation and the various neuroevolutionary techniques that are used to build such models will be reviewed.

Biological Background

Since spotted hyenas (*Crocuta crocuta*) and lions are both apex predators that compete for prey, resources and habitat, they come into conflict very frequently. Lions are much larger and more powerful than hyenas and are expected to win most such interspecific competitions. Hyenas are generally reluctant to engage with lions. Nevertheless, hyenas have sometimes been observed to exhibit a curious cooperative action where they band together to attack a group of lions in order to gain or retain access to a kill (Watts and Holekamp, 2008). Hyenas display other cooperative behaviors for hunting and for defense (Holekamp et al., 2012), but lion-mobbing is much more complex than these, and can be considered a novel evolutionary step. Mobbing is very dangerous for the hyenas (Trinkel and Kastberger, 2005; Kruuk, 1972). In fact, lions are the leading cause of death in many hyena populations (Cooper, 1991; Hofer and East, 1995; Trinkel and Kastberger, 2005). Consequently, hyenas can rarely displace lions from food unless the odds ratio (i.e. the

ratio of hyenas to lions) is at least four to one (Kruuk, 1972).

Dr. Holekamp and her colleagues have been continuously monitoring spotted hyena clans in the Masai Mara National Reserve and Amboseli National Park in Kenya since 1988. They have made direct observations of seven different hyena clans and recorded over 500 hours of videos and detailed notes about more than 900 lion-hyena encounters (Lehmann et al., 2016). Dr. Holekamp's group used this data to construct a table that characterizes each such encounter along dimensions such as the number of hyenas present, the number of lions, whether mobbing occurred, and whether it was successful. Using this dataset, they then characterized all the lion-hyena encounters and assessed mobbing probabilities in Lehmann et al. (2016). Some of the conclusions they reached were:

1. Lions and hyenas interacted more frequently at fresh kill sites than at sites with older carcasses. Mobbing rates were also highest at a fresh kill.
2. Lion-hyena interaction probability increased with increasing prey size.
3. The presence of adult male lions at the kill site increased the probability of interactions but decreased the probability of successful mobbing.
4. The interaction probability increased with number of hyenas present.
5. Local prey availability did not significantly impact the probability of interaction.
6. Mobbing increased the probability that hyenas would acquire food from a lion-controlled kill site. Thus, the evolution of cooperation in hyenas has increased their overall fitness.

The goal of this work was to understand the cognitive processes that result in mobbing behavior using a computational model to simulate lion-hyena interactions. All the conclusions from the observational data listed above were tested in simulation.

Simulations of Cooperative Behavior

A significant body of work exists on computational modeling of cooperation in nature. For instance, flocking behaviors of birds and shoaling of fish have been modeled extensively using rule-based approaches (Reynolds, 1987; Seno, 1990), while cooperative behavior of micro-organisms has been modeled with genetic algorithms (Kubota et al., 1996; Roeva et al., 2007). Ant and bee colonies have been the subject of many studies involving evolutionary computation as well (Dorigo et al., 1996; Waibel et al., 2006).

More complex cooperative behaviors in teams have also been studied in computation before. Yong and Miikkulainen

(2009) used neural networks to control and evolve the behaviors of three predators cooperating to catch a prey. Simultaneous cooperative and competitive coevolution was implemented in teams of predators and prey by Rawal et al. (2010), while dynamically changing hunting behaviors of hyenas were modeled in Rajagopalan et al. (2011).

Previous computational work also studied the effect of different communication strategies in mobbing, evolving the behaviors as a set of rules (Solomon et al., 2012; Fairey and Soule, 2014). The results showed that having a single leader to make all mobbing decisions for the hyena team resulted in the most effective coordination. But this result has not been observed in nature, and therefore, this will not be an assumption in this work.

Neuroevolution of Behavior

Neural networks and evolutionary computation may be combined into a learning algorithm that can be used to solve difficult sequential decision tasks with continuous state and action spaces, and partially observable states. Neuroevolution has previously been used to discover dynamic and intelligent behavior in autonomous agents. For example, it has been used in simulated robot soccer (Whiteson et al., 2005), and Ms. Pac-Man (Burrow and Lucas, 2009; Schrum and Miikkulainen, 2014). Neuroevolution has also been used in previous modeling of predators and prey (Yong and Miikkulainen, 2009; Rawal et al., 2010) as well as in the evolution of cooperative hunting behaviors in simulated hyenas (Rajagopalan et al., 2011). Thus, neuroevolution is a natural choice for modeling the complex cooperative behavior of lion-mobbing.

NeuroEvolution of Augmenting Topologies, or NEAT (Stanley and Miikkulainen, 2002), is a neuroevolution technique that optimizes not only the connection weights, but also the topology of a neural network. This technique was shown to be more effective than traditional neuroevolution methods that modify only the connection weights (Stanley and Miikkulainen, 2002). Speciation is also used to nurture new innovations in network structure that might otherwise be lost due to their low initial fitnesses. NEAT was used in this work when building a computational model to study lion-hyena interactions.

Experimental Setup

The hyena agents were placed on a 100×100 toroidal grid without any obstacles, where they could move east, west, north or south. A group of non-evolving lions already in possession of the kill were fixed at a location, and had the deterministic behavior of killing any hyena that came within a certain number of steps from them, i.e. the *interaction radius*, with a certain *kill probability*. Whenever a hyena moved closer than the interaction radius, it was said to be interacting with the lions. Then, it could either be killed or

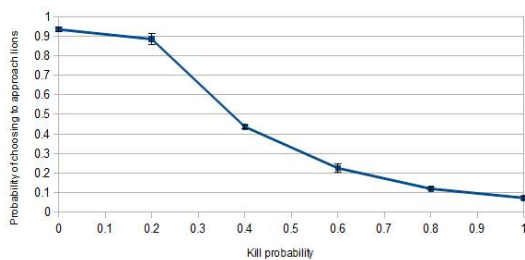


Figure 1: Interaction probability when kill probability is fixed. The x-axis has kill probability and the y-axis is the probability that the hyena approaches the lions, averaged over ten runs. The fixed value of the kill probability during each run had to be discovered by the hyena through evolution. The probability of approaching lions decreased as the kill probability increased.

be part of a successful mobbing event. The goal of the simulated hyenas was to mob the lions with enough teammates to drive them away and obtain the kill for themselves. An example of a successful mobbing behavior in simulation has been uploaded at <http://nn.cs.utexas.edu/?mobbingfactors>.

The hyena population was evolved using the NEAT algorithm (Stanley and Miikkulainen, 2002). For each simulation, a hyena was picked from the population and cloned to create the team members. Each hyena in the population was evaluated five times, and each experiment was run ten times. The fitness, mobbing probability and lion-hyena interaction probability were averaged across these ten runs.

A hyena:lion ratio of 4:1 is necessary for a successful mobbing event to take place (Kruuk, 1972). The kill probability in the simulation depended on the number of hyenas and lions, but it came into play only when a hyena entered the interaction circle of the lions.

In addition to mobbing reward given to successful mobbers, a survival reward was given to those hyenas that survived until the end of the simulation regardless of whether they participated in a mobbing event. This represented reward from hunting in real-life hyenas, and provided a fitness gradient for the evolution of the hyena neural networks.

In the following section, several experiments were designed to build and test a computational model for lion-mobbing. Representations of various parameters from the real world were gradually introduced and tested.

Using a Computational Model to Characterize Lion-Hyena Interactions

In the following experiments, various parameters were carefully and systematically tested in order to reproduce the hyena behaviors seen in nature.

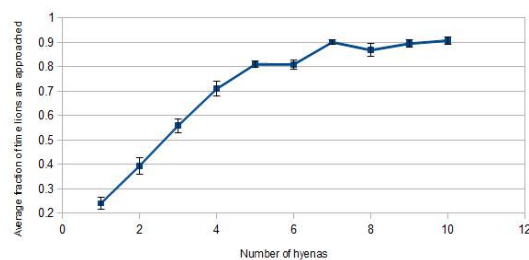


Figure 2: Interaction probability when number of hyenas varies dynamically. The number of hyenas is on the x-axis, and the y-axis is the probability of the hyena approaching the lions, averaged over ten runs. The number of lions was kept fixed at two. The numbers of hyenas and lions were given as input to the neural network, but the kill probability had to be calculated. The probability of approaching lions increased with number of hyenas.

Preliminary Experiments to Calculate Kill Probability

Initial experiments consisted of a single time step in which hyena agents decided whether to approach the lions, and were immediately rewarded. The number of clones created for each simulation run was chosen at random from [1, 10]. If the number of hyenas was more than four times the number of lions, the kill probability decreased to 0 and they could successfully mob the lions. Otherwise, it was equal to the normalized ratio of number of lions to number of hyenas.

More specifically,

$$K = \begin{cases} 0 & \text{if } H \leq 4L \\ \frac{\frac{L}{H} - 0.25}{0.75} & \text{if } H > 4L \end{cases}$$

where K was the kill probability, L was the number of lions, and H was the number of hyenas.

The numbers of lions and hyenas were not known to the hyena. However, both these numbers (and hence, kill probability) were kept fixed during an experiment run, so the hyena population could discover the kill probability through evolution. If the team of hyena clones chose to approach lions and were killed, they received a reward of $-10,000$ points. If they approached lions but were not killed, they got 1000 points. If they chose to stay away from the lions, a survival reward of 100 points was given to them. As expected, the fraction of time they chose to approach the lions decreased as the kill probability increased (Figure 1).

In the second experiment, the hyena neural network received as input the numbers of hyenas and lions, and had to calculate the current kill probability. The actual number of

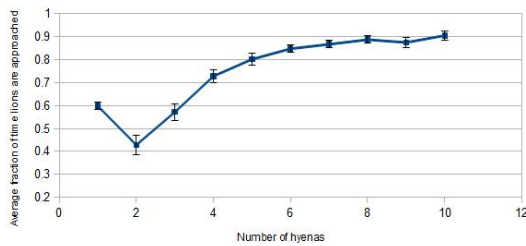


Figure 3: Interaction probability when lion-hyena distances are given. The y-axis shows the average probability of approaching the lions. The number of lions was kept fixed at 2. The absolute lion-hyena distances were given as input, but the number of hyenas was not. The probability of approaching lions generally increased with number of hyenas except for a small bump when number of hyenas was 1.

lions was kept fixed at 2. The number of hyenas in each run was chosen randomly between 1 and 10.

While the probability of approaching lions decreased with decreasing numbers of hyenas, it never reached 0 (Figure 2). Fewer hyenas in the environment lead to lower odds for a successful mob, and therefore the hyenas evolved to avoid the lions instead of getting killed. This behavior has also been observed in real-life hyenas (Lehmann et al., 2016).

In a third exploratory experiment, the hyena neural network received as inputs the distances of all other hyenas from the lions in addition to the number of lions (L) and its own x - and y -distances from the lions. The number of other hyenas was not an input. The other hyenas were virtual for now, so their distances were generated at random. Kill probability still depended only on the number of hyenas and lions, so the values of distances did not matter. However, the hyena had to evolve to count the number of distance inputs that were switched on and thus find out the number of other hyenas.

The probability of approaching the lion increased with increasing number of hyenas on the field with a small bump when there was only one hyena (Figure 3). When there were fewer hyenas than the *mob minimum*, it was unable to evolve to avoid the lions completely.

The next experiment had H non-virtual hyena clones instead of just one. Based on its inputs, each clone had to decide whether to attack the lions, which could lead to death, or avoid them, which gave smaller reward. The probability of interaction increased with increasing number of hyenas, just as stated in Conclusion 4 in the Biological Background section (Figure 4). The average maximum fitness reached for $H = 1$ and 2 was exactly equal to the survival reward, because H was below the mob minimum. But when $H = 3$ or 4, they chose to approach lions more often.

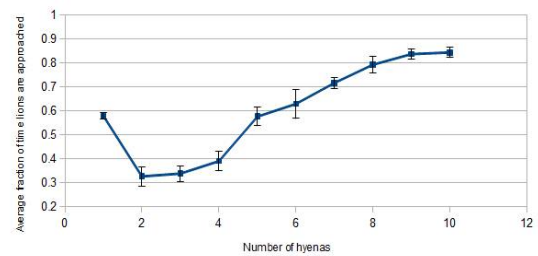


Figure 4: Interaction probability for multiple hyenas when lion-hyena distances are given. The number of lions was kept fixed at 2. The absolute lion-hyena distances were given as input to all the hyenas, but the number of hyenas was not. Each hyena had to make a decision whether to approach the lions. The average probability of approaching lions generally increased with number of hyenas, except for a small rise when there is a single hyena present.

Realistic Modeling of Lion-Hyena Encounters

The previous subsection paved the way for a multi-step simulation where the hyena clones could move around. Therefore, the number of time steps was increased to 500, and the hyenas could move east, west, north and south, or remain idle, represented by five output nodes. Each hyena neural network received a continuous input of the distances of itself and the other hyenas from the lions. The number of lions was fixed at 1. At every time step, if a hyena was within 10 steps (interaction radius) of the lion, kill probability came into play, which depended on how many hyenas were within the interaction circle (mob count). If a hyena was killed, it got a fitness penalty of $-10,000$ points and disappeared from the environment. For each time step that a hyena was within the interaction circle but did not die, it received a reward of five points, which represented the hyena feeding on the kill alongside the lions. If the mob count was greater than the mob minimum, the simulation was terminated with a mobbing reward of 10,000 points per hyena. At the end of the simulation, a reward of 100 points per hyena was given to all surviving hyenas. The inputs to the neural networks were the same as in the previous exploratory experiments, but now their values changed at every time step. The initial neural networks had 15 input-output neurons, 65 links and no hidden layer. The final evolved networks had around 130 neurons and around 400 links.

The results showed that the hyenas did not mob the lion successfully in most cases, but even the rare mobbing successes raised their average highest fitness above survival reward. More hyenas approached the lion even if they did not end up mobbing it. When the number of hyenas was less than the mob minimum, the best hyena teams evolved to avoid the lions altogether to stay alive and collect their survival reward. In general, the interaction probability in-

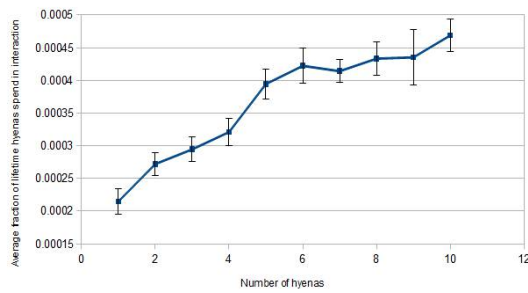


Figure 5: Interaction probability for hyenas in a 500-time-step simulation. The x -axis has number of hyenas, and the y -axis represents the average interaction probability, i.e. fraction of time the hyenas were within the interaction circle. The number of hyenas was not given as input, but the absolute lion-hyena distances were. The average interaction probability increased with number of hyenas.

creased with increasing number of hyenas (Figure 5). Overall, this probability was low because when a hyena entered the interaction circle, it would either soon get killed, or successful mobbing would occur, ending the simulation.

While the mobbing frequency was very low, the interaction probability increased with increasing number of hyenas in the environment, which is in line with Conclusion 4 from the Biological Background section. Successful mobbing did evolve and, therefore, can be productive for the hyena team and increase its fitness as long as mobbing gives a net gain to the hyenas. This agrees with Conclusion 6.

Increasing the Frequency of Successful Mobbing

While the hyenas in the previous subsection did evolve to successfully mob the lions, they did so very rarely. To increase the frequency of mobbing, various parameter values were tested carefully and systematically. The reward for remaining alive within the interaction circle needed to be higher to encourage hyenas to approach the lions. The interaction radius also needed to be increased in order to allow hyenas to drive away the lions from a greater distance. But this meant that the probability of hyenas dying also increased, since the kill probability came into play once a hyena was within the interaction circle. The mobbing reward per hyena was increased, while the survival reward as well as the mobbing reward were given only to those hyenas that survived to the end of the simulation, unlike in previous experiments. This change helped hyenas evolve to coordinate their attack on the lions instead of charging them blindly. The survival reward needed to be low so that the hyenas did not avoid the lions altogether.

The result of these parameter changes was that the frequency of mobbing increased when compared to previous experiments (see Figure 6). In all cases where the number of

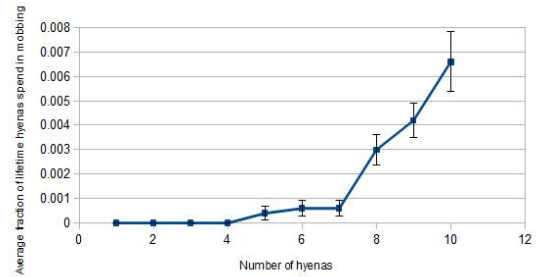


Figure 6: Successful mobbing probability for increased mobbing frequency. The y -axis represents the average mobbing probability. The interaction radius, mobbing reward and reward from feeding on the kill were all increased. A small survival reward was given to all surviving hyenas at the end of the simulation. The average mobbing probability was very low, but successful mobbing occurred more often than in previous experiments.

hyenas was five or more (the mob minimum), the mobbing probability was non-zero. The average mobbing probability increased with increasing number of hyenas on the field. The fraction of time the hyenas spent within the interaction circle, also increased (compare Figures 5 and 7). The average interaction probability also increased with increasing number of hyenas. A surprising development, which is also observed in nature, was that even when they did not have the numbers to mob the lion, they still obtained some reward from moving into the interaction circle and feeding on the kill. But this meant that they did not all stay alive until the end of the simulation, and thus the team did not get the maximum possible survival reward.

Presence of Adult Male Lions

Dr. Holekamp's group found that the presence of adult male lions in the lion group led to an increase in the probability of the hyenas and lions interacting (Lehmann et al., 2016). Male lions are more likely to initiate the interaction themselves (Elliott and Cowan, 1978), but are also better able to protect their kill, leading to lower mobbing frequency (Cooper, 1991; Kissui and Packer, 2004). The computational model matched Conclusion 3 from the Biological Background section in the following experiments.

Since male lions instigate interspecific interactions with the hyenas, the presence of male lions could be represented by a larger interaction circle in the computational model. The male lions also end up killing or injuring more mobbing hyenas due to their strength. That fact would also be true in the model when using a larger interaction circle, because any hyenas are more likely to step into the interaction circle if it is larger, and thus, they would be more likely to die.

In these experiments, the interaction radius was varied

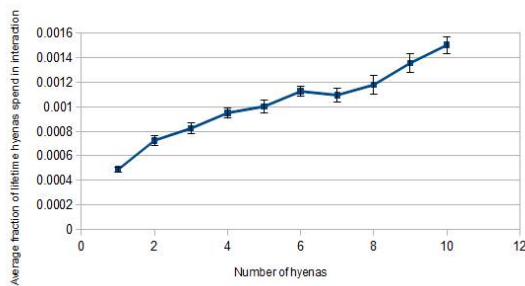


Figure 7: Interaction probability with increased mobbing frequency. The y-axis represents the average interaction probability, i.e. fraction of time the hyenas were within the interaction circle. The interaction radius, mobbing reward, and reward from feeding on the kill were all increased. There was also a small survival reward. The average interaction probability was higher than in previous experiments, and increased with number of hyenas.

randomly in the range 0 to 30 during the experiment run. This value was then given as input to the hyena neural network. The number of hyenas was fixed at 10 in all the experiments so that the comparisons between results would reflect only the changes in interaction radius.

Just like in nature, the simulated hyenas spent more of their time interacting with lions as the interaction radius increased, but they also got killed more frequently (see Figures 8 and 9). This result is in line with Conclusion 3 from the Biological Background section, which states that presence of male lions increases the interaction probability, but also increases the number of hyena deaths.

Prey Desirability

It makes intuitive sense that the desirability of the prey at the kill site should dictate whether the hyenas mob the lions to gain the kill, as concluded by Lehmann et al. (2016). Both interaction and mobbing rates were highest at a fresh kill site when compared to a kill site with an old carcass (Conclusion 1 from the Biological Background section). They also observed that the propensity of the hyenas for interspecific interactions with the lions increased with increasing prey size (Conclusion 2). The fitness boost from a successful mobbing had to be large enough to overcome the cost of injury or death while mobbing. In the experiments in this subsection, the freshness and the size of the prey were combined into one component, prey desirability. This component was represented by the mobbing reward in the simulations.

The simulated hyenas did not always behave in the same way as their real-life counterparts. If the initial mobbing reward was too low, the hyena team had a large overhead cost for evolving mobbing strategies and they simply avoided the lions altogether, preferring to collect the survival reward in-

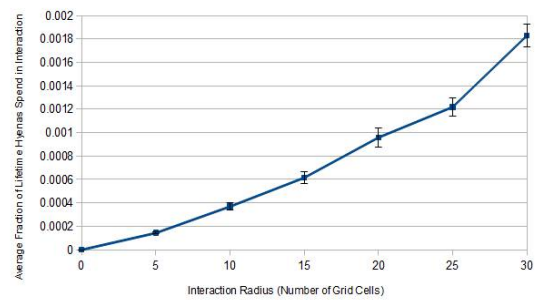


Figure 8: Interaction probability comparisons when adult male lions are present. The x-axis shows the interaction radius, and the y-axis represents the interaction probability, i.e. fraction of time the hyenas were within the interaction circle, averaged over ten runs. The numbers of hyenas and lions were kept fixed at 10 and 1, respectively. The interaction radius was varied dynamically between 0 and 30 steps. The average interaction probability increased as the interaction radius increased. Therefore, the frequency of interaction increased when there were male lions present.

stead. On the other hand, if the initial mobbing reward was too high and the survival reward too low, they evolved successful mobbing behaviors that they could execute with minimal cost even if the mobbing reward decreased later in the experiment run.

The survival reward values for these experiments had to be chosen very carefully to come up with a situation where hyenas could dynamically choose to mob or avoid lions based on the prey desirability. Different values of survival reward were tested systematically with varying success. When the survival reward was 5 fitness points, there was a trend of successful mobbing probability increasing with increasing prey desirability (see Figure 10). This result is in line with Conclusions 1 and 2 from the Biological Background section.

Discussion and Future Work

The computational model developed in this work to study lion-hyena interactions used neural networks for the hyenas. One challenge was that neural networks do not fear the lions in the same way that real hyenas do. If they evolve a good mobbing strategy, they always use it. If the net return from mobbing is very low, they evolve to never mob the lions instead. In order to replicate mobbing behaviors from nature, various parameters such as mobbing and survival rewards, and probability of injury or death had to be fine-tuned very carefully and systematically. However, the resulting successful settings suggested principles that make such behaviors possible.

It can hence be concluded that mobbing can be success-

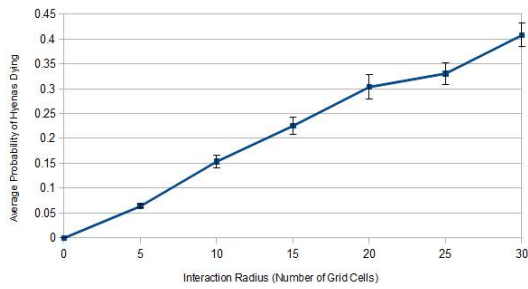


Figure 9: Hyena death probability comparisons when adult male lions are present. The x -axis shows the interaction radius, and the y -axis shows the probability of the hyenas dying, averaged over ten runs. The numbers of hyenas and lions were kept fixed at 10 and 1, respectively. The interaction radius was varied dynamically between 0 and 30 steps. The average death probability increased as the number of male lions increased, which is represented by increase in interaction radius.

ful without being counterproductive. Carefully coordinated mobbing leads to better overall fitness for hyenas because once they drive the lions away, there is no more danger. They also get a big fitness boost from eating the kill they wrested from the lions. These observations from computational simulation suggest that mobbing is possible and successful in specific circumstances, but not a very general and common ability. This may be the reason why mobbing is indeed rare, i.e. the spotted hyenas seem to do it, and not their closest relatives, the striped and brown hyenas.

The role of emotions such as fear and affiliation, as well the importance of individualistic traits in lion-mobbing has not been studied before. It is not clear exactly what information emotions provide to the hyenas and how they regulate behavior. As such, it is difficult to simulate emotion inputs to hyena neural networks. Similarly, a heterogeneous team will behave very differently from the homogeneous team employed in this work. The different roles of individual hyenas are hard to replicate in simulation when not much about these roles has been observed in nature. Therefore, computational modeling of emotions and individualistic traits in the context of lion-hyena interactions is still future work.

Conclusion

The computational model built in this paper was used to study lion-hyena interactions and the evolution of successful mobbing strategies. Several factors affected the evolution of realistic behaviors on the part of the hyenas, including interaction radius, mobbing reward, survival reward and reward from feeding on the kill gradually when mobbing has not occurred. In order to replicate frequent mobbing behaviors as seen in nature, the values of these factors had to be very

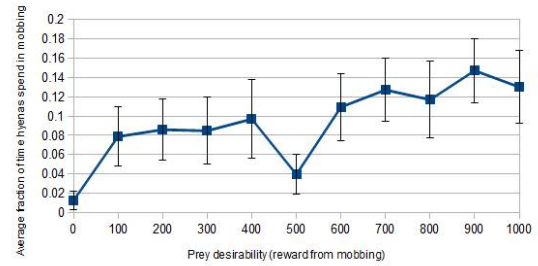


Figure 10: Successful mobbing probability comparisons for different prey desirability values. The x -axis shows the prey desirability value, which is equal to the mobbing reward, and the y -axis shows the mobbing probability, i.e. fraction of time the hyenas successfully mobbed the lion, averaged over ten runs. The survival reward was 5 points. The average mobbing probability showed a trend of increasing with increasing prey desirability.

carefully and systematically varied. From the simulation experiments in this work, it could be concluded that mobbing and interaction frequencies increased with increase in interaction radius, mobbing reward, and reward from feeding on kill gradually inside the interaction circle. These frequencies also increased when survival reward was reduced, and when mobbing and survival rewards were only given to survivors.

These parameters represented environmental factors in the real world that affect interspecific interaction probabilities and mobbing rates in hyenas. Out of the six conclusions from observational data listed in the Biological Background section, five were modeled successfully in simulation. The following are the conclusions that matched perfectly:

1. The probability of lion-hyena interaction increased with number of hyenas present. This result matches Conclusion 4.
2. Successful mobbing contributed positively to the overall fitness of the hyena team, although a fine balance of parameter values was necessary to bring about mobbing behaviors. This result matches Conclusion 6.
3. Interaction probability was higher when adult male lions were present, but the probability of death and injury for the hyenas was also greater. In simulation, the presence of male lions was represented by a larger interaction radius. This result matches Conclusion 3.
4. Interaction probability was higher when prey desirability was higher. In the computational model, prey desirability was represented by the mobbing reward. This result matches Conclusions 1 and 2.

Since the results discovered through simulation and those observed in nature were congruent, the computational model

was deemed to be a faithful representation of real-life lion-hyena encounters. The next step then is to use this model to make predictions about hyenas in nature, which could be tested in the field in future. The behaviors simulated here can also be used to create complex cooperative behaviors in artificial agents in the future.

Acknowledgements

This research was supported in part by NSF grant DBI-0939454, and in part by NIH grant R01-GM105042.

References

- Burrow, P. and Lucas, S. M. (2009). Evolution versus temporal difference learning for learning to play Ms. Pac-man. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2009)*, pages 53–60.
- Cooper, S. (1991). Optimal hunting group size: the need for lions to defend their kills against loss to spotted hyaenas. *African Journal of Ecology*, 29(2):130–136.
- Dorigo, M., Maniezzo, V., and Colnari, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41.
- Elliott, J. P. and Cowan, I. M. (1978). Territoriality, density, and prey of the lion in ngorongoro crater, tanzania. *Canadian Journal of Zoology*, 56(8):1726–1734.
- Fairey, J. and Soule, T. (2014). Evolution of communication and cooperation. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 169–176. ACM.
- Hofer, H. and East, M. (1995). Population dynamics, population size, and the commuting system of serengeti spotted hyenas. *Serengeti II: dynamics, management, and conservation of an ecosystem*, 2:332.
- Holekamp, K. E., Smale, L., Berg, R., and Cooper, S. M. (1997). Hunting rates and hunting success in the spotted hyena (*crocuta crocuta*). *Journal of Zoology*, 242(1):1–15.
- Holekamp, K. E., Smith, J. E., Strelhoff, C. C., Horn, R. C. V., and Watts, H. E. (2012). Society, demography and genetic structure in the spotted hyena. *Molec. Ecol.*, 21(3):613–632.
- Kissui, B. M. and Packer, C. (2004). Top-down population regulation of a top predator: lions in the ngorongoro crater. *Proceedings of the Royal Society of London B: Biological Sciences*, 271(1550):1867–1874.
- Kruuk, H. (1972). The spotted hyena: a study of predation and social behavior. *Wildlife Behavior and Ecology*, pages 315–325.
- Kubota, N., Shimojima, K., and Fukuda, T. (1996). Virus-evolutionary genetic algorithm-coevolution of planar grid model. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 1, pages 232–238. IEEE.
- Lehmann, K. D. S., Montgomery, T. M., MacLachlan, S. M., Parker, J. M., Spagnuolo, O. S., VandeWetering, K. J., Bills, P. S., and Holekamp, K. E. (2016). Lions, hyenas and mobs (oh my!). *Current Zoology*.
- Rajagopalan, P., Rawal, A., Miikkulainen, R., Wiseman, M. A., and Holekamp, K. E. (2011). The role of reward structure, coordination mechanism and net return in the evolution of cooperation. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2011)*, pages 258–265.
- Rawal, A., Rajagopalan, P., and Miikkulainen, R. (2010). Constructing competitive and cooperative agent behavior using coevolution. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2010)*, pages 107–114.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34.
- Roeva, O., Pencheva, T., Tzonkov, S., Arndt, M., Hitzmann, B., Kleist, S., Miksch, G., Friehs, K., and Flaschel, E. (2007). Multiple model approach to modelling of escherichia coli fed-batch cultivation extracellular production of bacterial phytase. *Electronic Journal of Biotechnology*, 10(4):592–603.
- Schrump, J. and Miikkulainen, R. (2014). Evolving multimodal behavior with modular neural networks in ms. pac-man. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 325–332. ACM.
- Seno, H. (1990). A density-dependent diffusion model of shoaling of nesting fish. *Ecological Modelling*, 51(3):217–226.
- Solomon, M., Soule, T., and Heckendorn, R. B. (2012). A comparison of a communication strategies in cooperative learning. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 153–160. ACM.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Trinkel, M. and Kastberger, G. (2005). Competitive interactions between spotted hyenas and lions in the etosha national park, namibia. *African Journal of Ecology*, 43(3):220–224.
- Uchibe, E. and Asada, M. (2006). Incremental coevolution with competitive and cooperative tasks in a multirobot environment. *Proceedings of the IEEE*, 94(7):1412–1424.
- Waibel, M., Floreano, D., Magnenat, S., and Keller, L. (2006). Division of labour and colony efficiency in social insects: effects of interactions between genetic architecture, colony kin structure and rate of perturbations. *Proceedings of the Royal Society B: Biological Sciences*, 273(1595):1815–1823.
- Watts, H. and Holekamp, K. E. (2008). Interspecific competition influences reproduction in spotted hyenas. *Journal of Zoology*, 276(4):402–410.
- Whiteson, S., Kohl, N., Miikkulainen, R., and Stone, P. (2005). Evolving soccer keepaway players through task decomposition. *Machine Learning*, 59(1-2):5–30.
- Yong, C. H. and Miikkulainen, R. (2009). Coevolution of role-based cooperation in multiagent systems. *IEEE Transactions on Autonomous Mental Development*, 1(3):170–186.

Towards modelling social habits: an organismically inspired evolutionary robotics approach

Manuel G. Bedia¹, Manuel Heras-Escribano², Diego Cajal¹, Miguel Aguilera^{1,2}, and Xabier E. Barandiaran^{2,3}

¹ ISAAC Lab, Aragón Institute of Engineering Research, University of Zaragoza, Zaragoza, Spain.

² IAS-Research Center for Life, Mind and, Society, UPV/EHU, University of the Basque Country, Spain.

³ Department of Philosophy, Faculty of Labor Relations and Social Work, UPV/EHU, University of the Basque Country, Spain.
mgbedia@unizar.es

Abstract

There has been a revival of the notion of habit in the embodied and situated cognitive sciences. A habit can be understood as ‘a self-sustaining pattern of sensorimotor coordination that is formed when the stability of a particular mode of sensorimotor engagement is dynamically coupled with the stability of the mechanisms generating it’ (Barandiaran, 2008, p. 281). This view has inspired models of biologically-inspired homeostatic agents capable of establishing their own habits (Di Paolo and Iizuka, 2008). Despite recent achievements in this field, there is little written about how social habits can be established from this modelling perspective. We hypothesize that, when the stability of internal behavioural mechanisms is coupled to the stability of a behaviour and other agents are present during this behaviour, a social interdependence of behaviour takes place: a social habit is established. We provide evidence for our hypothesis with an evolutionary robotics simulation model of homeostatic plasticity in a phototactic behaviour. Agents evolved to couple internal homeostasis to behavioural fitness display social interdependencies in their behaviour. The social habit of these agents was not interrupted when blindness to phototactic stimuli was introduced as long as social perception remained active. This did not happen when internal homeostasis was not coupled to the fitness of the agent. The results allow us to propose a possible conjecture about the character of social habits and to offer a potential theoretical framework to understand how habits develop from neurodynamics to the level of social interaction.

From individual to social habits: philosophy, psychology, neural and social sciences

Habits in the history of philosophy

The concept of habit has been key for making sense of our cognitive abilities. Philosophers, psychologists, and cognitive scientists from the most diverse traditions have used this concept to make sense of behavior and cognition from Ancient Greece to the 20th Century (Sparrow and Hutchinson, 2013). The advent of computationalism and information processing prompted a decline of the notion of habit in the last Century (Barandiaran and Di Paolo, 2014), although the notion has gained momentum recently with the rise of the embodied and situated cognitive sciences (Di Paolo et al., 2017; Egbert and Barandiaran, 2014; Barandiaran, 2017).

Despite its popularity, there has not been a unified notion of habit through the ages. However, we can differentiate two main branches: (1) The associationist view, that defines habits as automatic responses, and (2) the organicist view, that defines habits as self-organizing structures linked to environmental aspects through which agents establish tendencies with positive or negative balance (Di Paolo et al., 2014). This second branch, which is the one we will focus on, was received by pragmatism, and it was combined with the idea of habit as mentioned in the works of Charles Darwin (James 1890: Chapter 4). On the other side, the first branch became relevant with the scientific revolution, and developed a view that was partially mechanistic and focused on subpersonal processes of automatization. The associationist conceptualization of habits was inherited by the contemporary neuroscientific approach (Wood and Rüniger, 2016).

The organicist view goes back to Aristotle, and it was based on three main ideas: habits are dispositions or tendencies (active capacities of the agent), they are agential (they belong to the agent as a whole) and they are relational (those tendencies of the agent are directed towards an environmental element). The term ‘habit’ comes from the Greek words ‘ethos’ and ‘hexis’ and Aristotle understood hexis as the capacity of the agent for being in possession of something in an active way, as ‘a kind of activity of the haver and of what he has something like an action or movement’ (Aristotle, 2007, 5.20, 1022b1214). This is, a kind of disposition towards through which someone established a tendency, which can be positive or negative (Faucher and Roques, 2018, 2). In this sense, a habit is an active capacity possessed by the agent, who establishes a structure that gives rise to a particular behavior. This is tightly related to the meaning of ‘ethos’, which means ‘custom’ or ‘habit’: that particular behavior is established to be repeated. Habits are also relative to something external, as Aristotle himself defines them (Aristotle, 2004, 8, 11a2032), because these behavioral patterns or tendencies are always related to an object or being of the outer world. This Aristotelian view of habits inaugurates an understanding of habits as self-structured patterns of action that depend on environmental elements, something

that was inherited by traditions as diverse as pragmatism or phenomenology.

The relevance of the organicist view of habits declined with the raise of mechanicism and natural philosophy after the achievements of the scientific revolution. Natural philosophers often described the body in mechanical terms, and this picture has been present thorough history until nowadays. This is the basis of the asociacionist view of habits: habits are unconscious, rigid and automatic behaviors that are formed thanks to the association of different stimuli to certain behavioral outputs. The gap between stimuli and response is offered in subpersonal mechanistic terms. This mechanistic view of the body governed the main developments in psychology and neuroscience, explaining the connection between stimuli and response as a bunch of discrete subpersonal steps that are mechanically linked to each other from perception to action. This asociacionist view was at the basis of a wide variety of theories, from behaviorism to cognitivism (Reed, 1996) and also inspired the main current views of neuroscience (Bennett and Hacker, 2003, 2008; Kandel et al., 2000)

In the 20th Century, pragmatism and phenomenology inherited the organicist view of habits. First of all, Jamesian psychology included the idea of habit as a cognitive capacity inspired by the work of Darwin. James claimed that there are habits or tendencies that are innate (instincts) or educated (acts of reason) (James, 1890). In any case, these tendencies belong to the agent as a whole and, given the anti-structuralist and anti-elementarist approach of James, habits cannot be reduced to a series of mechanical and discrete subpersonal steps. James claimed that habits are an example of the plasticity of the behavior of organic life that helps us compensating the perturbations of the environment. At the same time, habits help us to automatize our behavior, relieving us from excessive conscious attention (Blanco, 2014). In this sense, James' notion of habit combines the agential, active and dispositional advantages of organicism with the emphasis on automatization and unreflectiveness of the asociacionist view.

Habits are understood in Merleau-Ponty's phenomenology as the bunch of ways in which our body establishes a meaningful dialogue with our surroundings by means of our action, allowing us to experience the world as a network of attractions and repulsions (Moya, 2014). Merleau-Ponty claimed that our understanding (this is, our capacity for making sense of the environment) is not based on abstract rationality but on our bodily and unreflective understanding with the world (Kaufer and Chemero, 2015). In this sense, Merleau-Ponty served as an inspiration for those cognitive scientists that endorsed an embodied and situated approach (Moya, 2014; Kaufer and Chemero, 2015; Gallagher, 2017; Heras-Escribano, 2019). Taking this, the embodied and situated approach to cognitive sciences is a direct heir of the organismic view of habits, although passed through the lens

of the pragmatist and phenomenological views offered here; this is, habits are self-structured dispositions or tendencies that allow us to related to the world, but in an unreflective and embodied way. The tendencies of our body towards environmental elements allow us to navigate it and compensate its perturbations in a skillful but totally unreflective way.

From individual to social habits

The term 'ethos', which is at the basis of the idea of habit in the Aristotelian tradition, is also understood in moral terms, such as the ethos of a person or a community (this is why 'ethics' and 'morals' derive from this word). Thus, individual habits also have a moral dimension and, more importantly, a social dimension. In fact, the work of some pragmatist philosophers, such as Dewey, relies on the constitutive aspect of the social environment for establishing both individual and shared habits.

Dewey proposed an ontology of mind based on habits in order to outcompete the subpersonal, mechanistic view of the reflex arc concept in psychology (Dewey, 1896). This reflex arc concept endorsed the idea that perception was passive and separated from action. According to the reflex arc view, the senses received the impingements from the environment and, thanks to a series of discrete and mechanic steps, it was delivered a behavioral output that was automatic and dis-embedded. Dewey claimed that this is not a reliable picture of perception and action: in his view, perception was continuous with action and context-dependent. First of all, perception starts with the exploration of the environment, so stimuli are informed by the previous activity of the organism. Also, the response is based on the particular situation in a given spatiotemporal context that includes the particularities of the natural and social environment that the organism is exploring. In this sense, stimulus and response are two abstractions from the continuous and cyclic process of exploration of the environment. This is why Dewey discards the mechanistic and subpersonal approach of behaviorism and develops an ontology of mind based, among other things, on the organicist idea of habit.

Habits in the Deweyan view are understood as the constitutive dispositional patterns that organize human behavior (Dewey, 1958, pp. 20-6, 40-1), but the key point is that they are socially acquired or established. This is because, according to Dewey, our individual habits are built upon previous collective habits in the sense that there is a priority of the society over the individual (Dewey, 2002, p. 58). This priority is not based on metaphysical assumptions, but on the natural origins of humans: every individual is born in a particular society, which means that individuals develop their own action patterns in accordance with the social ones and thanks to the feedback of the community (Dewey, 2002, p. 59). So, following Dewey, habits are always evaluated or assessed by the individual's social environment, and this assessment comes in the form of a reinforcements and sanc-

tions (Dewey, 2002, pp. 16-17). For this reason, Dewey claimed that habits are ‘ways of incorporating the environment’ (Dewey, 2002, p. 15) and this includes, of course, the social environment. The social constitution of habits is so crucial for Dewey that he claimed that it is practically impossible for humans to form habits that are not socially-established, so our social environment plays a constitutive role in the shaping and establishment of individual habits (Dewey, 2002, p. 16).

The notion of habit has also been vindicated as foundational for social sciences. One such line of development departs from Norbert Elias and Marcel Mauss and finds its most prominent exponent with Pierre Bourdieu (1977). They all share in common the view that habits are not only social constructions shaping individual behaviour, but the most important building block or structuring process of society (in contrast with intellectualist views on how ideology, rational thinking or individual goal-oriented strategic planning shapes society).

Bourdieu defines habitus as:

systems of durable, transposable *dispositions*, structured structures predisposed to function as structuring structures, that is, as principles of the generation and structuring of practices and representations which can be objectively ‘regulated’ and ‘regular’ without in any way being the product of obedience to rules, objectively adapted to their goals without presupposing a conscious aiming at ends or an express mastery of the operations necessary to attain them and, being all this, collectively orchestrated without being the product of the orchestrating action of a conductor. (Bourdieu, 1977, p. 72)

We aim to enrich recent contributions in which habits have been modelled (Di Paolo and Iizuka, 2008) by introducing a social dimension of habits inspired by the main insights of social habits as defined by Dewey and social theorists. These social habits, then, satisfy both the organismically-inspired homeostatic model of habit-formation (because we rely on processes of homeostasis as a key for the establishment of social habits, offering a naturalistic continuum between individual and social habits) as well as the Deweyan demands of understanding habits as ‘ways of incorporating the [social] environment’ (Dewey, 2002, p. 15). At the same time, Bourdieu’s notion of structuring structures resembles a classical theme of Artificial Life, that of self-organizing structures, where a process produces and re-produces its conditions for self-perpetuation or self-structuring and it affords the opportunity to model social life using artificial life modelling techniques. Artificial Life should thus be ready to face both challenges: the social dimension of habit and the habitual dimension of society.

In the next subsection we will explain how a recent modelling version of individual habits have been proposed in the

embodied and situated cognitive sciences. This version includes homeostasis as a key aspect for offering a naturalization of habits as understood within the organicist tradition.

Modelling habits for organismically-inspired homeostatic agents

An organismically inspired approach to habit formation in evolutionary robotics and artificial life is not new. Ezequiel Di Paolo pioneered the modelling of homeostatic behaviour generating mechanisms to simulate re-adaptation to visual inversion (Di Paolo, 2000), a paradigmatic experimental setup aimed at discovering the deep entanglement of sensorimotor habits and processes of re-habituation (Kohler, 1963). Variations of these homeostatic mechanisms have been many (Aguilera et al., 2016, 2015; Iizuka and Di Paolo, 2008; Di Paolo, 2003; Williams and Noble, 2007; Iizuka and Di Paolo, 2007), but perhaps the most relevant for our purpose here is that of Barandiaran and Di Paolo (2010), where the authors develop a model of habit development, maintenance and switching for an operant conditioning task and the spontaneous emergence of new associative habits not present during artificial evolution. The internal mechanism capable of supporting this behaviour was homeostatic plasticity in a continuous time recurrent neural network and a sensorimotor embodied task. A behavioural sensorimotor pattern is linked to an emergent region of stable synaptic plasticity. Being plasticity activity dependent, variations on sensorimotor history are capable to generate, reshape those regions or to switch between them. Long terms disruption of sensorimotor correlations can also destroy the underlying neurosynaptic structures supporting them. However, the high dimensionality of the system makes it hard to analyse. In an attempt to simplify the model Egbert and Barandiaran (2014) chose a mesoscopic level of modelling, avoiding neural mechanisms, and proposing instead an iterant deformable sensorimotor medium (IDSM) a mode of plastic sensorimotor mapping that became structured through repetition. The spontaneous emergence of an ecology of habits was displayed on that model (see Egbert, 2018, for further developments). Furthermore, other works have used minimal models to analyze the relation between agency and agency and social interaction Di Paolo et al. (2008), although they focus on interactive dynamics in a social context without considering the capacity of agents of developing plastic habits.

From an organismic perspective a habit can be defined as ‘a self-sustaining pattern of sensorimotor coordination that is formed when the stability of a particular mode of sensorimotor engagement is dynamically coupled with the stability of the mechanisms generating it’ (Barandiaran, 2008, p. 281). As we can see, this model offers a naturalization of habits that satisfies the organicist view shown here, as self-organizing structures that govern the action patterns of an agent. In particular, it satisfies the two main requirements: they are active and relational, because the modelled agent

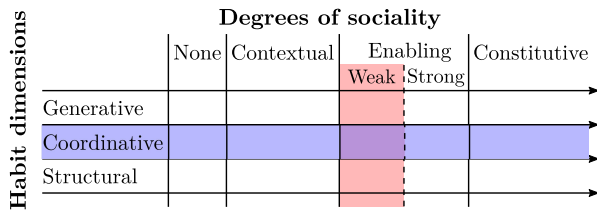


Figure 1: Types of sociality of a habit

displays a behavior based on dispositions and tendencies actively and in relation to the source of stimulation in the environment. The introduction of a social dimension to this definition of habits has to do with the social or interactive dependence of sensorimotor coordinations, their generation and stability.

What is a social habit? We can establish the sociality of a habit by degrees along three axes or *dimensions* of sociality: the behavioural or coordinative dimension (whether the enactment itself in action demands coordination with other agents), the structural or stabilizing dimension (whether the stability of the action demands social interaction) and the generative dimension (whether the habit can be acquired without social interaction). Building upon De Jaegher et al. (2010) we can distinguish three *degrees* of sociality or interactivity: contextual, enabling and constitutive. Contextual sociality means that other agents' behavior *S* simply affect or produce variations on habit *H*. Enabling sociality means that habit *H* cannot happen without *S* and constitutive sociality means that *S* is inherently part of *H*. We want to add a further distinction within the enabling degree, which can be *strong* when *S* has to happen, *ceteris-paribus*, for *H* to occur, and *weak* when *S* has to happen for *H* to occur only in cases in which conditions change (e.g. an impairment has taken place). So, for example, dressing up might be something that is done individually, without coordination with others. But it is certainly generatively social (wild infants have not been seen to dress up) and is probably a structurally social habit (meaning that out of a social context, the care and disposition to dress up fades away). However dressing up *fashionably* is constitutively social at the coordinative level. Similarly if your partner has often a say on how you should dress up, whether you follow the advice or directly and systematically oppose it, the habit will be contextually social at the coordinative dimension.

We hypothesize that, when the stability of internal behavioural mechanisms is coupled to the stability of a behaviour, and other agents are present during this behaviour, a social interdependence of behaviour takes place: a social habit is established, even if the task is not coordinatively social. It follows that no specific 'social mechanism' is required for a social habit to emerge. We provide evidence for our hypothesis with an evolutionary robotics sim-

ulation model of homeostatic plasticity in a phototactic behaviour. Agents evolved to couple internal homeostasis to behavioural fitness display social interdependencies in their behaviour so that blindness to the very object of taxis does not disrupt the habit, provided that social perception is still active. This constitutes a mode of *weak social enabling* for habit enactment (see Figure 1).

In order to explore this hypothesis we defend a phototactic task is a sufficiently complex task so as to discover some relevant sociality phenomena. Pierre Bourdieu often conceives social phenomena as occurring in a social field where agents (individuals) move trying to maximize certain values (e.g. social status). Interestingly for Bourdieu, both the perception of the field is the result of habit and, at the same time, moving through the field creates the field itself. In this paper we are not going to develop such a deep conception of social *habitus*, but it suffices to note that moving toward a light source can potentially mimick the structure and phenomenology of more complex forms of social behaviour.

Model

In Ashby's seminal work (Ashby, 1952), the author states that the adaptation of organisms can be understood as the maintenance of the internal stability through the homeostatic regulation of some essential variables that ensure survival. By doing so, Ashby proposes a conceptual framework for the development of artificial agents in which, rather than assuming that organisms have behaviors that seek intentional goals, they are able to generate self-induced adaptive behaviors exclusively through the maintenance of their internal stability. Based on Ashby's ideas, Di Paolo (2000) applied notions of homeostatic regulation to the synaptic activity of agents involved in phototaxis, demonstrating that they could adapt to sensory inversion despite not having been developed specifically for this task. Di Paolo explains how that adaptation arises in this model: although agents do not evolve to adapt to inversion of the visual field, it is suggested that evolution has created a link between structural stability and desired behavior. This link appears because the process shapes a stable attractor when a certain pattern of sensorimotor activity is present.

We use the same kind of minimal agent guided by homeostasis to search stable configurations where the plasticity effects generate the behaviour and behaviour affects plasticity, leading to the creation of invariants and habits in a social scenario. This allows us to investigate how ultrastability can lead to adaptation in a social dimension.

Model: We simulated a pair of minimal agents evolved to exhibit phototaxis on a series of 1000 light sources. Agents are evaluated in a two-dimensional environment with only a single light source at a time. During each trial, agents are encouraged to remain close to the light sources. Light sources are placed randomly at a distance between 10 and 25 times the agent's radius from the agent position, for a random pe-

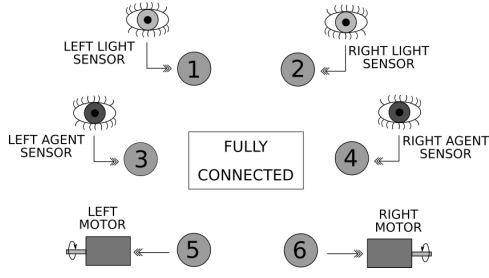


Figure 2: Agent's controller

riod of time T , chosen from the interval $[2000, 4000]$ time steps and with a random intensity.

Agent: Each agent is modelled as a simple circular agent with two light sensors and two motors. If we name V_l and V_r as the motors' velocities and r as the agent's radius (for the experiments, the body's radius is set at 4), we calculate the translational movement as $(V_r + V_l)/2$ and the rotational movement by calculating the angular speed as $(V_r - V_l)/2r$. Collisions are not considered.

Sensors: Each agent has two light sensors, separated by 120° , symmetrically placed at the front of each agent and each sensor has a viewing cone of 80° . At each time step, the angle from the agent to the light source is first calculated and then adjusted by the agent's rotation. This angle is then used to determine if the agent's sensors are active. Light intensity on each sensor can be calculated as inversely proportional to the square of the distance to the light source. In addition, agents have two agent sensors, placed on the same positions and with the same cone of vision that light sensors. Each agent can be seen as a mobile source with constant intensity, so another agent can see it as it see lights. Intensity values on sensors are calculated in the same way as light case.

Controller: A 6-neuron fully connected Continuous Time Recurrent Neural Network is used as the agent's controller. Each neuron is a CTRNN node described by

$$\dot{y}_i = \frac{1}{\tau_i} \left(-y_i + \sum_{j=1}^N w_{ji} \sigma(y_j + \theta_j) + I_i \right) \quad (1)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

where $i = 1, 2, \dots, 6$. Each neuron's state is controlled by y_i , representing the cell's potential, τ_i is the decay constant, b_i the bias, z_j the firing rate of the j_{th} neuron, w_{ij} the strength of synaptic connection from node i to node j , and I_i the input from sensors. Inputs and outputs are connected as depicted in Figure 2.

Plasticity: Controllers are able to regulate synaptic activity. Plastic changes in the recurrent network occur locally on each connection governed by both the synaptic activity and

a plasticity rule encoded genetically. The plasticity rule is given by one of these hebbian learning rules

$$\Delta w_{ij} = \delta_i n_{ij} p_j z_i z_j \quad (3)$$

$$\Delta w_{ij} = \delta_i n_{ij} p_j (z_i - z_{ij}^0) z_j \quad (4)$$

$$\Delta w_{ij} = \delta_i n_{ij} p_j z_i (z_j - z_{ij}^0) \quad (5)$$

$$\Delta w_{ij} = 0 \quad (6)$$

where Δw_{ij} is the change per unit of time to a synaptic weight w_{ij} , z_i and z_j are the firing rates of the presynaptic and postsynaptic neurons, n_{ij} is the learning rate factor, and p_j is the degree of local plasticity. δ_i models a damping factor, and z_{ij}^0 a weakening of a synapse.

Genetic Algorithm: Agent's genotypes are divided into a real component and an integer component. The real component has 87 genes: 3 for motor and sensor gains, one per each type; 6 bias and 6 decay constants, one per neuron; 36 weights and 36 learning rates, one per synapse. Finally, 36 integer values are used to select a hebbian rule from (3) to (6) for each synapse. A genetic algorithm is used to test agents against a sequence of five lights. Two agents with the same genotype are selected at random to test their aggregate phototaxis in every step. Fitness scores of each pair of agents are assessed as the averaged individual fitness. Then, selected agents, using crossover and mutation, form a new offspring, which replaces the lowest scoring pairs. A mutation probability of 0.5 is used for all real components and a probability of 0.1 is used for integer components. The GA is repeated until no perceivable fitness increase is observed.

Homeostatic fitness function: The agent's behaviour is assessed using a fitness function with three terms where the first, F_d , is a measure of how near to the source an agent is (measuring the reduction in final and initial starting positions for the agent), F_p is a measure of time spent near the source (proportion of time over the evaluation period that the agent is within 4 radius units of the light source) and F_h is a measure of the homeostatic behaviour of an agent's neurons (the time-averaged proportion of neurons that act homeostatically, without inducing plasticity in the controller). The results of all three sub-fitness functions are ranged between $[0, 1]$. $F_H = 0.2F_d + 0.64F_p + 0.16F_h$. Each component has got a weight to adjust the contribution of each term of the fitness function in order to have the sum equal to 1.

Non-homeostatic fitness function: In order to evaluate how homeostasis affects the behaviour of agents, two different types of fitness are used: one taking into account the term for homeostatic stability (F_H), and the other $F = 0.2F_d + 0.8F_p$ not taking it into account. Evolving populations of pairs of robots with these two possible fitness functions, we obtain two types of agents: homeostatic and non-homeostatic. From each population, we select the best pair of agents with best performance for the subsequent analysis.

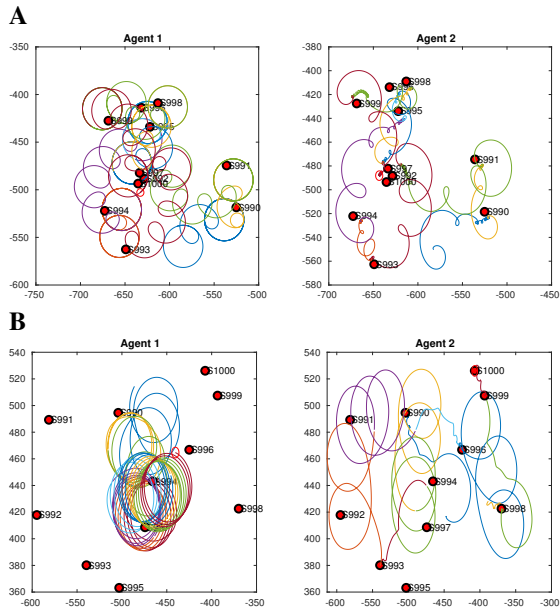


Figure 3: Behaviour of two agents performing the task in homeostatic (A) and non-homeostatic (B) conditions. Red dots are the light sources, numbered as they appears. Agent 1 is blinded, so it can perceive the other agent but not the light. We observe that, in the homeostatic case, both agents are able to behave in coordination and reaching the lights, suggesting that a collective habit is displayed.

Results

We want to explore the relation between individual capacities and social interaction in the maintenance of habits. For doing so, we artificially ‘blind’ one of the agents and compare the case of the robots with synaptic plasticity with agents not evolved for homeostatic stability. We only blind light sensors. Thus, blinded agents can keep the sight of other agents, which are the only clue to reach light sources. We then observe how the behaviour of the agents is reconfigured after synaptic plasticity stabilizes the activity of agents.

In Figure 3, we show the behaviour of the two agents in the case when Agent 1 is blinded, for the case with homeostatic stability (A) and without (B). In the first case, we observe that, even when Agent 1 is not able to perceive the light anymore, the agents are able to collectively achieve the task after homeostatic reconfiguration. In the case without homeostatic stability, this is no longer the case, and the behaviour of the blind agent is disrupted. This suggests that synaptic plasticity maintains a collective pattern of behaviour.

In order to test this change in a more systematic way, we simulate the behaviour of a (normal or blinded) Agent 1 and a (normal) Agent 2 reaching a series of 20 lights. We repeat each experiment 40 times and measure the fitness F (without considering homeostatic stabilization terms F_h). In Figure 4

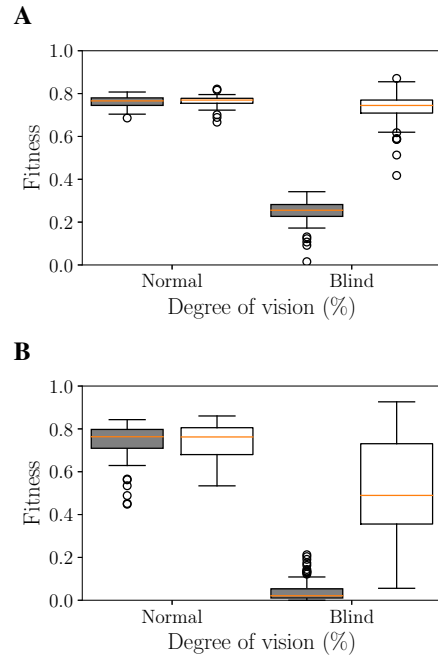


Figure 4: Fitness values of Agent 1 (gray) and Agent 2 (white) when homeostatic plasticity is (A) included (B) not included in evolution. Fitness is compared between normal conditions and a situation in which Agent 1 is blind to the light. Fitness of Agent 1 only maintained to some degree in the case of homeostatic stability.

we observe how fitness F change when Agent 1 is blinded to the light. In the case with homeostatic stabilization, the fitness of agents in the normal case is of 0.77, and when Agent 1 is blinded its fitness is reduced to 0.26, while the fitness of Agent 2 is reduced to 0.74. This means that, while Agent 2 is practically unaffected, the fitness of Agent 1 is reduced to one third, although it is still able to reach a number of lights thanks to its coordination to Agent 2. Thus, even when Agent 1 has lost its individual abilities for phototaxis, it is able to maintain its phototactic habit to some extent thanks to social interaction.

In the case without homeostatic stability, the situation is different. Now the mean fitness is reduced from around 0.76 to 0.02 for the blinded agent and 0.49 for Agent 2. In this case, losing its individual vision means that Agent 1 is no longer able to maintain its phototactic habit. Furthermore, the phototactic habit of Agent 2 is disrupted when Agent 1 can no longer reach the light, and its performance is reduced. In further tests, we tested the effect of ‘blinding’ the agent to social interaction instead of the light, without practically no effect in the fitness of the agents, so results are not reported.

We tried to increase the number of agents working together on a step-wise manner. Good performances are achieved in populations of 3 ($F = 0.71$) and 4 ($F = 0.67$)

agents with only a blind agent per group. However, the GA did not provide any successful case of more than four agents. An element to consider to explain this is that agents can't see the position of each other agent but a sum of their contributions to an agent sensor.

Discussion, conclusions and future work

In this article, we have explored the social dimension of habits understood under the lens of homeostatic neurodynamic models of habit generation. These models have been mostly focused in the emergence of individual sensorimotor habits and we have shown how a richer picture should incorporate the social dimension of habits.

But... what is the nature of social habits? Is it a merely aggregation of individual habits? Or does the social dimension play a constitutive role for generating emergent collective habits? How can we attempt a gradual path towards social habits? A social account of habits should clarify these aspects and offer a sound theoretical framework of how habits are developed from neural and sensorimotor dynamics to the level of social interaction and back.

We have provided a theoretical framework that classifies the types and degrees of sociality of habits, and we have illustrated a weak enabling sociality of the coordinative dimension of habits with a model of evolutionary robotics: a minimal model of agents controlled by a dynamical neural network incorporating mechanisms of homeostatic synaptic plasticity evolved to solve a task in a social scenario. A pair of agents have to perform a phototactic task. Each agent has sensors that perceive the presence of light sources, and another set of sensors that perceive the presence of the other agent. These agents are compared with a similar population adapted to solve the task without a pressure for stabilizing its synaptic plasticity, therefore reducing drastically the capacity for self-regulation of emerging patterns or habits.

In order to investigate the degree of sociality of the phototactic habits in our robotic model we perform an experiment over the resulting agents in which we artificially 'blind' one of the agents of the group. Thus, the capacity to express an individual habit is removed, yet its capacity for social interaction remains present. In this case, we can see that agents with homeostatic stabilization are able to maintain to some extent the performance of the task, therefore illustrating that the social dimension has a role in the maintenance of a phototactic habit. In contrast, agents that are not selected for homeostatic stabilization lose their capacity for phototactic behaviour entirely when blinded, suggesting that in this case the phototactic habit cannot be maintained by social interaction when direct photosensitivity is precluded. Still, performance of non-blinded agents is reduced when an agent is blinded, suggesting that the phototactic habit also has some social component. This suggests that the relation between habits and its individual and social components can be intricate, and further studies should clarify these relations. The

modeling results presented here are tentative and just seek to illustrate the relation between some ideas presented in this article. Further work could explore in more depth different possibilities to study different ways in which a habits can be socially constituted at different levels and degrees.

As we have seen, the dependence between the maintenance of habits, the presence of individual abilities and social coordination is complex. A possibility to unveil the complexities of these relations could be to introduce different scenarios in which noise or perturbations selectively disturb the individual or the social behaviour of the agents. This should force the agents to rely more on their individual capacities or in one another, depending on the case. This could allow the exploration of different kinds of the habits in our classification (Figure 1).

A natural next step on the evolution of social habits is to apply this framework to a constitutively social coordination task. We have focused on a simple phototactic behaviour that can be carried out individually. On the other hand, it is typical of collective behavior to include a necessary or constitutive coordination between agents: e.g. dancing, moving as a whole, etc. Long term studies of the structural stability of the habits out of the social dimension would also be interesting as future work.

Another research line that could be explored is the social constitution of the behavioural field. For instance, habits could be formed not (or not only) by exploiting the synaptic plasticity of an agent's neural controller, but by plastically modifying its own environment. For example, the activation of lights could be the result of the behaviour of the agents (e.g. light intensity being reinforced or consumed by the proximity of an agent). This will help approaching the concept of social field by Bourdieu.

Acknowledgements

Heras-Escribano was supported by the BBVA Foundation 2018 Leonardo Grant¹, the Project FFI2016-80088-P funded by the Spanish Ministry of Science, and the FiloLab Group of Excellence funded by the University of Granada, Spain. Miguel Aguilera was supported by the UPV/EHU post-doctoral training program ESPDOC17/17 and partially by projects TIN2016-80347-R and FFI2014-52173-P funded by the Spanish Ministry of Economy and Competitiveness. Xabier E. Barandiaran acknowledges funding from project FFI2014-52173-P by the Spanish Ministry of Economy and Competitiveness.

References

- Aguilera, M., Barandiaran, X. E., Bedia, M. G., and Seron, F. (2015). Self-Organized Criticality, Plasticity and Sensorimotor Coupling. Explorations with a Neurobotic Model in a Behavioural Preference Task. *PLoS ONE*, 10(2):e0117465.
- Aguilera, M., Bedia, M. G., and Barandiaran, X. E. (2016). Extended Neural Metastability in an Embodied Model of Sensorimotor Coupling. *Frontiers in Systems Neuroscience*, 10.

¹The Foundation accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors

- Aristotle (2004). *The Categories*. Kessinger Publishing.
- Aristotle (2007). *Metaphysics*. Mineola, NY: Dover.
- Ashby, W. R. (1952). *Design for a Brain*. J. Wiley, 2 edition.
- Barandiaran, X. E. (2008). *Mental Life: a naturalized approach to the autonomy of cognitive agents*. PhD Thesis, University of the Basque Country (UPV-EHU), Donostia - San Sebastian, Gipuzkoa, Spain. 17th June 2008.
- Barandiaran, X. E. (2017). Autonomy and Enactivism: Towards a Theory of Sensorimotor Autonomous Agency. *Topoi*, 36(3):409–430.
- Barandiaran, X. E. and Di Paolo, E. A. (2010). Homeostatic Plasticity in Robots: from development to operant conditioning to habit formation. In *Proceedings of CogSys2010*, Zurich.
- Barandiaran, X. E. and Di Paolo, E. A. (2014). A genealogical map of the concept of habit. *Frontiers in Human Neuroscience*, 8:522.
- Bennett, M. R. and Hacker, P. M. S. (2003). *Philosophical Foundations of Neuroscience*. Wiley-Blackwell.
- Bennett, M. R. and Hacker, P. M. S. (2008). *History of cognitive neuroscience*. John Wiley & Sons.
- Blanco, C. A. (2014). The principal sources of William James' idea of habit. *Frontiers in Human Neuroscience*, 8.
- Bourdieu, P. (1977). *Outline of a Theory of Practice*, volume 16. Cambridge university press.
- De Jaegher, H., Di Paolo, E., and Gallagher, S. (2010). Can social interaction constitute social cognition? *Trends in Cognitive Sciences*, 14(10):441–447.
- Dewey, J. (1896). The reflex arc concept in psychology. *Psychological review*, 3(4):357.
- Dewey, J. (1958). *Experience and nature*, volume 471. Courier Corporation.
- Dewey, J. (2002). *Human nature and conduct*. Courier Corporation.
- Di Paolo, E., Buhrmann, T., and Barandiaran, X. E. (2017). *Sensorimotor Life: An enactive proposal*. Oxford University Press.
- Di Paolo, E. A. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, Cambridge, MA. MIT Press.
- Di Paolo, E. A. (2003). Organismically-inspired robotics: homeostatic adaptation and teleology beyond the closed sensorimotor loop. In Murase, K. and Asakura, editors, *Dynamical systems approaches to embodiment and sociality*, pages 19–42. Advanced Knowledge International, Adelaide.
- Di Paolo, E. A., Barandiaran, X. E., Beaton, M., and Buhrmann, T. (2014). Learning to perceive in the sensorimotor approach: Piaget's theory of equilibration interpreted dynamically. *Frontiers in Human Neuroscience*, 8.
- Di Paolo, E. A. and Iizuka, H. (2008). How (not) to model autonomous behaviour. *Biosystems*, 91(2):409–423.
- Di Paolo, E. A., Rohde, M., and Iizuka, H. (2008). Sensitivity to social contingency or stability of interaction? Modelling the dynamics of perceptual crossing. *New Ideas in Psychology*, 26(2):278–294.
- Egbert, M. (2018). Investigations of an Adaptive and Autonomous Sensorimotor Individual. *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 343–350.
- Egbert, M. D. and Barandiaran, X. E. (2014). Modeling habits as self-sustaining patterns of sensorimotor behavior. *Frontiers in Human Neuroscience*, 8:590.
- Faucher, N. and Roques, M., editors (2018). *The Ontology, Psychology and Axiology of Habits (Habitus) in Medieval Philosophy*. Historical-Analytical Studies on Nature, Mind and Action. Springer International Publishing.
- Gallagher, S. (2017). *Enactivist interventions: Rethinking the mind*. Oxford University Press.
- Heras-Escribano, M. (2019). *The Philosophy of Affordances*. Palgrave Macmillan, New York, NY, 1st ed. 2019 edition edition.
- Iizuka, H. and Di Paolo, E. A. (2007). Toward Spinozist Robotics: Exploring the Minimal Dynamics of Behavioral Preference. *Adaptive Behavior*, 15(4):359–376.
- Iizuka, H. and Di Paolo, E. A. (2008). Extended Homeostatic Adaptation: Improving the Link between Internal and Behavioural Stability. In *From Animals to Animats 10. 10th International Conference on Simulation of Adaptive Behavior*, pages 1–11.
- James, W. (1890). *The Principles of Psychology*. Henry Holt and Company.
- Kandel, E. R., Schwartz, J. H., Jessell, T. M., Department of Biochemistry and Molecular Biophysics Thomas Jessell, Siegelbaum, S., and Hudspeth, A. (2000). *Principles of neural science*, volume 4. McGraw-hill New York.
- Kaufer, S. and Chemero, A. (2015). *Phenomenology: an introduction*. John Wiley & Sons.
- Kohler, I. (1963). The formation and transformation of the perceptual world. *Psychological issues*.
- Moya, P. (2014). Habit and embodiment in Merleau-Ponty. *Frontiers in Human Neuroscience*, 8.
- Reed, E. S. (1996). *Encountering the world: Toward an ecological psychology*. Oxford University Press.
- Sparrow, T. and Hutchinson, A. (2013). *A History of Habit: From Aristotle to Bourdieu*. Lexington Books.
- Williams, H. and Noble, J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems*, 87(23):252–259.
- Wood, W. and Runger, D. (2016). Psychology of Habit. *Annual Review of Psychology*, 67:289–314.

Speciation under changing environments

Kevin Godin-Dubois, Sylvain Cussat-Blanc and Yves Duthen

University of Toulouse, IRIT - CNRS UMR 5505, 2 rue du Doyen Gabriel Marty, 31042 Toulouse, France
{kevin.dubois, sylvain.cussat-blanc, yves.duthen}@irit.fr

Abstract

Progress in molecular genetics allowed taxonomists to better understand the relationships between species without the bias of morphological similarities. However, access to data from times past is limited to the fossil archives which, being far from complete, can only provide limited information. To address this problem through the field of Artificial Life, we devised a polyvalent sexual reproduction scheme and an automated phylogenetic tool capable of producing, from a stream of genomes, hierarchical species trees with relatively low memory footprint. We assert that these apparatus perform well under reasonable stress by embedding them into 2D simulations of unsupervised plant evolution in textbook cases of geographical speciation. After thousands of generations and millions of plants, the extracted phylogenetic data not only showed the expected results in terms of branching pattern (anagenesis, cladogenesis) but also exhibited complex interactions between species both in space and time.

Introduction

Phylogenetic trees of our world's species display the overwhelming amount of variations, adaptations and bifurcations generated by unbridled evolution over the course of a few billion years. Despite being visualization tools, designed to classify a continuum into more easily manageable chunks, they can provide further insight into the underlying mechanisms of natural selection be it in biological or artificial systems.

Before getting to point where phylogeny is relevant, one first needs the basic block of any evolutionary process: individuals. Models of genotype-phenotype mapping abound in the literature with a specific emphasis on the class of morphologies they can generate. One of the first contributions is the Lindenmayer Systems (Prusinkiewicz et al., 1995), heavily inspired by the branching patterns exhibited by plants, which encode, in a very compact form, recursive derivation and can reproduce life-like instances in both 2D and 3D (Bornhofen, 2008). The directed graphs designed in (Sims, 1994) follow a similar approach by defining body segments and the, potentially recurrent, relationships between them and, though originally designed to model motile creatures, were successfully applied to plant morphologies (Dubois

et al., 2017), as well. Further generalization led to the biologically inspired Genetic Regulatory Networks which, by defining the cell as the elementary unit, emulate its internal chemistry through self-interacting 'proteins' controlling its life-cycle. Using such a low-resolution building block allowed for the generation of specific shapes (Joachimczak and Wróbel, 2008) and organ emergence for creatures embedded in a virtual environment (Disset et al., 2016).

Natural selection, however, is not a genetic algorithm and the metaphor fails as soon as one tailors a fitness function to drive evolution into solving an optimization problem. Indeed, when left unchecked, even the simplest of rule set such as (Gardner, 1970) can create such diversity that they are still investigated almost fifty years later. Thus, it ensues that simulations in which individuals roam free have been designed: ranging from (Adami and Brown, 1994), with its computer programs fighting one-another for memory space, to (Metivier et al., 2002; Ventrella, 2005) where motile creatures are required to actively look for mating partners.

Despite the preponderant place given to the living parts of an ecosystem, its abiotic component is equally important given that models of the biosphere using only water and temperature as variables were found to account for most of the vegetal biodiversity observed in Nature (Woodward and Williams, 1987). The same holds true for artificial simulations as disruptiveness, whether sudden or diffused, promotes different strategies and leads to diversity (Bornhofen et al., 2011). Unforeseeable environmental dynamics add another layer of complexity in the generated individuals by selecting those that exhibit better adaptability (Canino-Koning et al., 2016).

Circling back to the biotic component, we see its self-driving force in numerous examples of co-evolution, co-adaptation, competition, whether in the natural world or artificial systems (Miconi, 2008): arming race is a striking case of an inter-species conflict able to quickly promote divergence of character or optimization. But the concept of species is a blurred one: though one can argue that for individuals there is no such thing as a species, only mates and non-mates, we refer to the definition of *biolog-*

ical species given in (Singh, 2012) as a “group of potentially interbreeding natural population reproductively isolated from other such groups”. Similarly, the process of speciation, by which species are created, has been described in a number of ways without the emergence of global consensus. (Butlin et al., 2008) argues that attempts at categorizing a continuous multi-dimensional phenomenon by discrete topology-centered methods makes one lose sight of the adjacent equally important factors.

This paper comes as a proof-of-concept of how environmental conditions can be used as the sole driving force of an evolutionary process. To this end, we hereby describe the model for autonomously reproducing individuals and the phylogenetic tool used to monitor species dynamics.

Self-reproducing vegetals

Here, individuals are not only expected to adapt to an unknown, and potentially precarious, environment but also to thrive by self-reproducing to the utmost limit. As our objective is to study evolutionary dynamics instead of individual development, we decided upon using L-Systems as our morphological controller, as opposed to more complex plant growth models, thanks to their intelligibility and computational lightness.

L-System

Each plant comes with a pair of L-System: the shoot and root. These are deterministic, context-free and share the same set of control instruction: $+/-$ for left/right rotations, [...] branching, $A...F$ non-terminals and S the initial non-terminal. The shoot manages the above-ground portion of the plants’ structure and uses the terminals s (stem), l (leaf), f (flower) while the below-ground compartment instead relies on t (root trunk) and h (root hair).

Individual rules can mutate through duplication, replacement or suppression of an existing symbol, extraction into its own branch (e.g. slf becoming $s[l]f$) and swapping adjacent (e.g. slf giving sfl). In addition, rules can be added (e.g. $S \rightarrow slf$ giving $S \rightarrow sAf; A \rightarrow l$) or removed (e.g. $S \rightarrow sAf; A \rightarrow l$ reducing to $S \rightarrow sf$).

Some restrictions apply to these operators. There is always at least one rule, in which case it must be the one derivating the initial non-terminal (S which can be seen as a seed) so that the plant can germinate. As these L-Systems are deterministic the maximal number of rules is the size of the non-terminals set. Rules cannot be longer than $M = 4$ non-control ($\notin \{+, -, [,]\}$) characters long so that derivations must occur when aiming for complex morphologies. Finally, the number of replacements a plant can perform for a given compartment is limited to a small value $D \leq 5$, itself subject to mutations, which bounds the number of symbols in the derived phenotype to M^D .

Constants	
k	assimilation rate
J	saturation rate
f	resource cost
l	life cost
m_{Tr}, s_{Tr}	temperature range regulation
Genetic fields	
g_s	Growth speed
m_T, s_T	Plant’s temperature parameters
R_E	Resistor for transportation of element E
Environmental conditions	
P	plant’s position
T	temperature at P
X^L	Biomass for layer L
R_E^L	Reserve in layer L of element E
C_E^L	Concentration in layer L of element E
T^-	1 if $T < m_T$, 0 otherwise
T^+	1 if $T > m_T$, 0 otherwise
w_h	Water around root hair h
s_h	Surface of root hair h
l_l	Length of leaf l exposed to the sun

Table 1: Metabolic variables

Metabolism

Similar to (Bornhofen et al., 2011), plants in this model have three ‘reservoirs’ per compartment: one for water, which is extracted by root hairs h below the surface, one for glucose, produced by photosynthesis from leaves l , and dry biomass generated by converting these nutrients.

In addition, the effects of external temperature are taken into account at multiple stages of the metabolic dynamics whose control parameters are detailed in table 1. Given the bell curve function of mean m and standard deviation s

$$gauss(x, m, s) = exp^{-\frac{(x-m)^2}{2s^2}} \quad (1)$$

a plant’s heat efficiency at temperature T is defined as

$$h_{eff}(T) = gauss(T, m_T, s_T) gauss(s_T, m_{Tr}, s_{Tr}) \quad (2)$$

The left-hand part of the equation impedes the metabolism as T goes further from the plant’s optimal temperature m_T while the right-hand part regulates the tolerance range s_T so that it cannot grow unchecked. Indeed, the individuals must strike a balance between resilience to greatly varying temperatures (at the cost of average efficiency) and optimization for specific environmental conditions (at the risk of extinction should these change too much). This impacts water uptake as, the lower the temperature is below m_T , the less a plant can absorb water through its root hairs:

$$U_W(T) = \frac{T^-(h_{eff}(T) - 1) + 1}{1 + C_w^{rt} J} \sum_{h, \text{root hair}} k w_h s_h \quad (3)$$

Nonetheless, the root compartment shares a portion of its water reserve to the shoot, according to the relative concen-

trations and transport resistors:

$$T_W = \frac{C_W^{rt} - C_W^{sh}}{\frac{R_W}{X^{rt}} + \frac{R_W}{X^{sh}}} \quad (4)$$

Leaves in the upper layer with direct access to sunlight then produce glucose and similarly to (4) transports part of it to the lower layer.

$$U_G = \frac{1}{1 + C_G^{sh} J} \sum_{l, leaf} kl_l \quad (5)$$

$$T_G = \frac{C_G^{sh} - C_G^{rt}}{\frac{R_G}{X^{rt}} + \frac{R_G}{X^{sh}}} \quad (6)$$

When placed under too hot environmental conditions, plants will additionally experience water loss through transpiration.

$$R_W^{sh}(T) = (1 - T^+ h_{eff}(T)) R_W^{sh} \quad (7)$$

Extreme temperatures can lead to a complete drain of their shoot water reserves in a day. Plant tissue turnover is modeled by continuously transforming part of the biomass in to wastes:

$$W^L(T) = l(2 - h_{eff}(T)) X^L \quad (8)$$

External conditions influence this as well by inflicting upon plants under uncomfortable temperatures up to 200% the rate of cellular decaying experienced by siblings under a more favorable climate. Finally, both glucose and water reserves are consumed to generate new biomass which is allocated to the various sinks (flowers, fruits, stems and root trunks) in the plant.

$$\dot{X}^L(T) = g_s X^L C_W^L C_G^L - W^L(T) \quad (9)$$

One should note, however, that, whenever wastes production exceeds dry biomass renewal, $\dot{X}^L(T)$ will be negative. That is, sinks will *lose* biomass causing them to shrink. This leads to their death as soon as their individual biomass is completely depleted, removing them and their subtrees from the plant. Starvation is, thus, one of the possible cause of death for an individual: when all of its sinks are destroyed the plant itself is considered dead. Senescence is the other one, as determined by an evolved genetic field, thus preventing immortal phenotypes from monopolizing the environment.

Self-reproduction

One of the most powerful tools available to Life is its ability to adapt through the process of natural selection. Over the course of history, numerous propagation schemes have been developed. We chose to focus, in this work, on sexual reproduction because of its greater degree of interactions and inter-species diversity.

The subset of a plant's genotype devoted to reproduction includes its gender, compatibility metrics $CM = \{\mu, \sigma_i, \sigma_o\}$ and sexual organs. These interact with one another according to the algorithm defined in previous work (Godin-Dubois et al., 2019).

The genotypic distance is defined recursively: given A, B two genomes, e an elementary field with range $[e_{min}, e_{max}]$ (e.g. the growth speed g_s) and r a compound field composed of subfields $f^1 \dots f^n$ with weights $w^1 \dots w^n$

$$dist(e_A, e_B) = \frac{|e_A - e_B|}{e_{max} - e_{min}} \quad (10a)$$

$$dist(r_A, r_B) = \sum_{i=1}^n w^i dist(f_A^i, f_B^i) \quad (10b)$$

This metric is *objective* in the sense that it provides information on the amount of genetic divergence between a pair of individuals but makes no hypothesis as to their capability to mate. The compatibility value, on the other hand, is *subjective* and asymmetric as different sets of the reproduction parameters CM may give very different results for an identical genetic distance d .

This crossover operator differs from those commonly found in the literature (Sims, 1994; Bornhofen, 2008; Disset et al., 2016) on three points: 1) it can fail early on, 2) is biased by the *female* genome and 3) has low resistance to large structural differences. The rationale behind point 3 is that, instead of devising a robust operator that can produce a somewhat viable offspring from two completely unrelated individuals, a minimalist alignment procedure is better suited to sexual reproduction of same species creatures in which the population is mostly homogeneous. Indeed, point 1 guarantees that the more both genomes are different the less likely it is that crossing will be attempted at all.

Embedding the compatibility function into the genome allows for the emergence of species-specific segregation schemes which is of utmost importance as our interest lies in obtaining speciation as a by-product of reproduction at the individual level. Furthermore having both in-/out-breeding coefficients makes specification of the search spaces possible, with adaptive plants accepting a broader range of incoming genetic material while more conservative ones could instead focus on controlled inbreeding to solidify their alleles.

Automated phylogeny tool (APOGeT)

Studying long term evolutionary dynamics generates a massive amount of data which precludes observation at an individual level. To this end, we devised a tool for automated phylogeny which only relies on genomes possessing both a distance metric and a compatibility function (e.g. as defined above).

In APOGeT, species are modeled by a fixed-size collection of "representative" points that form an envelope in the

genetic space. This allows for a compact, yet diverse, description of a species without resorting to centroids which, when applicable, would shrink individual differences.

The procedure for inserting a genome g in the tree is two-fold: first, determine the correct species for g and update the envelope if need. Given S , the species of g 's parents and $env(S)$ the collection of representatives, we test whether $match(g, S) \geq T$, given that:

$$xcompat(g, e) = \min(g.compat(d), e.compat(d)) \quad (11)$$

with $d = dist(g, e)$

$$match(g, S) = \frac{1}{|env(S)|} \sum_{e \in env(S)} xcompat(g, e) \quad (12)$$

If the result is positive we can assign g to S . Otherwise, the procedure is performed for each direct subspecies of S until either a match is found or a new species is created with g as its sole representative.

Then the envelope E of the modified species' is checked for update. If it is not yet full (less than K are recorded) then the genome is simply appended. Otherwise, g 's contribution is confronted against that of the $e_i \in E$ according to:

$$C(E, g) = \max_{e_i \in E} \left(- \min_{e_j \neq e_i} dist(e_i, e_j) \right. \\ \left. + \min_{e_j \neq e_i} dist(g, e_j) \right) \quad (13)$$

If $C(E, g) > 0$, then g is more different than a current envelope point and will be inserted in its place. In this manner, the envelope for a given species is a set of those individuals, while still capable of inter-crossing, that are the most different.

Hybridization

Earlier work on this tool faced us with the problem of hybridization between species, whether occasional or deliberate. Indeed the algorithm presented above makes the assumption that both parents belong to the same species which, under unrestricted genome flow conditions, is not necessarily the case. This implies that for a given species I individuals may come from a, potentially large, set of candidate parent species $H_1 \dots H_n$ thus changing this tool from a tree to a graph and losing much intelligibility in the process.

To solve this issue we introduced the concept of *major contributor* as follow: Given g with parents p_m, p_f belonging to different species S_f, S_m , the algorithm presented earlier makes the assumption that $S_f = S_m$. The sufficient extension managing multiple parent species is to check against both and resume the rest of the procedure for:

$$S = \arg \max_{S_f, S_m} match(g, S_i) \quad (14)$$

In order to keep track of these hybridizations, each species S maintains a list of contributions $C^S = \{\{S_i, c_i\} \dots\}$

which records for any species S_i how many times c_i it provided genetic material. This allowed us to the redefine the notion of parent species as the *major contributor* i.e. $S_i \in C^S / S_i \neq S, \forall_j c_i > c_j$. It also implies that whole subtrees can be reparented to reflect the change in genetic material source.

The environment

So that individuals can be subjected to a large range of dynamical abiotic conditions, the environment can produce changes along three dimensions:

Topological y , with seeds being much harder to disseminate onto higher ground

Hygrometric w , water availability has a direct impact on the plants' ability to thrive

Temperature t , equations (2-8) show how deleterious this can be on the metabolism

The system is designed as a closed one, so that one can easily plug any kind of controller between the input D, Y, x, y, w, t and output $\dot{y}, \dot{w}, \dot{t}$ variables where D is the relative time in the current year ($\in [0 : 1]$) and Y the relative time in the planned simulation duration (same range) and x the position in the environment. All other values have range $[-1 : 1]$. For this article, we resorted to a simple expression parser to easily define straightforward experimental validations.

A pair of constraints $C0, C1$ (controlled by the genomic coefficients c_0, c_1) is used to post-process the outputs of the environmental controller so as to provide more plausible correlations between physical dynamics.

$$C0 : \dot{t} = -c_0 \mathbb{1}_{\mathbb{R}_{\geq 0}} y + (1 - c_0) \dot{t} \quad (15)$$

$$C1 : \dot{w} = -c_1 \mathbb{1}_{\mathbb{R}_{\leq 0}} t + (1 - c_1) \dot{w} \quad (16)$$

That is temperature decreases linearly with an increase in altitude and water evaporates more (and thus also decreases) as temperature raises.

Additionally, a bare-bone physics engine is embedded in the system to prevent plant-plant collisions, manage light availability and perform mates detection. All of our simulations generate the initial population from a single primordial genome, which is cloned 100 times and disseminated regularly around the center of the environment. These are then left to their own devices for a number of years where days and years have durations of 10 ticks and 100 days, respectively.

Experiments

In order to validate both our autonomous reproduction scheme and phylogeny extraction tool we devised simple scenarios to test our system on. Namely, we explore allopatric, parapatric and a form of peripatric speciations.

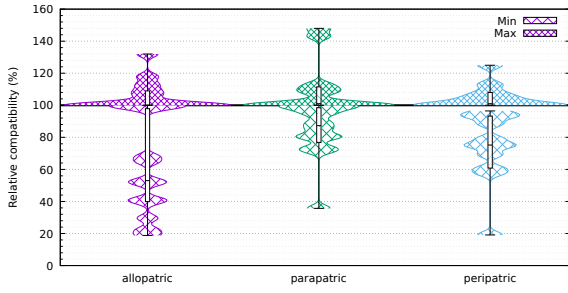


Figure 1: Speciation results for the three experiments

Parameters not subjected to variation are the environments' width (100m) and height (50m), simulation time of 100 years and identical primordial plant genome.

In all of these experiments, we are interested in whether or not *strong* speciation occurred, that is we are more focused on the apparition of reproductively isolated species than of varieties. To this end, we defined the following metrics:

The absolute compatibility between species A and B at a given timestep with $P^A = \{P_1^A \dots P_n^A\}$, the female plants of species A , and $P^B = \{P_1^B \dots P_m^B\}$, the male plants of species B , is:

$$c_a(A, B) = \frac{1}{|P^A||P^B|} \sum_{f \in P^A} \sum_{m \in P^B} f.compat(m) \quad (17)$$

That is the average compatibility between possible mating pairs of each considered species. We then derived from $c_a(A, B)$ the relative compatibility as follow:

$$c_r(A, B) = \frac{c_a(A, B)}{c_a(A, A)} \quad (18)$$

which provides a normalized metric whose comparison between different reproductive trends or even simulations is more straightforward. Results across all three experiments are summarized in figure 1 with an uneven number of repeats: 13, 12 and 11 for the allopatric, parapatric and peripatric, respectively. This corresponds to the subset, from 20 runs per protocol, that neither immediately go extinct nor failed to reach the 100th years, in the allotted 10 hours timeframe. Note that, given the definition of $c_r(A, B)$, the minimal worse and maximal best relative compatibility is 100%. Indeed, the worst case scenario would be having all values clustered at, or very close to, 100% which would show a striking lack of speciation. Given that this is not the case, we can safely conclude that *some* did occur, which will be explored in the following sections.

Allopatric speciation

$$\begin{aligned} c_0 &= c_1 = 1 \\ \dot{t} &= .75 \sin(.5Y\pi) \text{ gauss}(x, .5, .05) \\ \dot{y} &= \dot{w} = 0 \end{aligned}$$

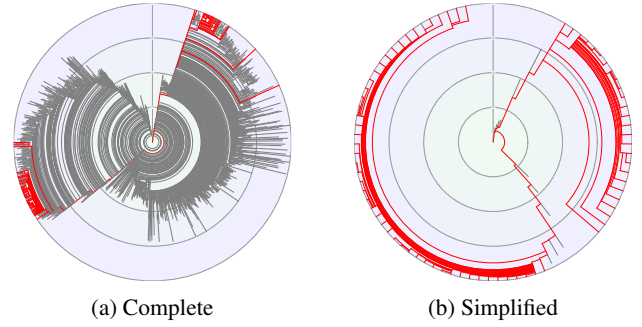


Figure 2: Phylogenetic tree for the lowest c_r at the 100th year

Our first test case is focused on the most simple mode of speciation: complete geographical isolation. To this end, our environment, otherwise uniform, slowly grows a mountain in its center according to the parameters described previously. This gradual process produces, at the end of the simulation, a topological barrier 37.5 meters high and 20 meters large. As seeds have difficulty reaching higher places this effectively prevents cross-reproduction between individuals from either side.

As seen in 1, speciation did occur in this experiment, however aggregated data can only show a coarse picture. To this end, we extracted the phylogenetic tree produced during the most successful run (minimal $c_r = 18.8\%$, maximal = 100%) which can be seen in figure 2a.

For a given species the number of available information is limited to the minimum of what can be easily processed at a glance: an arced path connects it to its parent species with the distance to center providing the date of the first appearance of this species. The timeline pointing outward stops as soon as no more individuals can be found in the simulation. Additionally, paths in red denotes species on a 'survivor path' i.e. those that left living descendants at the end of the simulation.

One can clearly distinguish the two species clusters stemming from the geographical separation with the lower part of the leftmost one failing to provide viable species past the 75th year. Unfortunately, however complete this graph may be, it is too densely packed with extinct species to provide

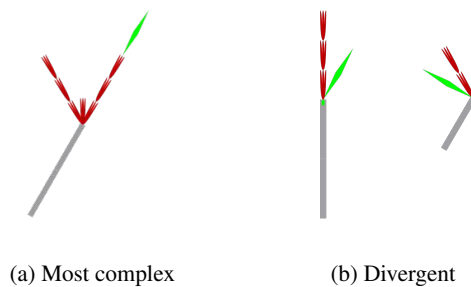


Figure 3: Morphologies show limited complexity

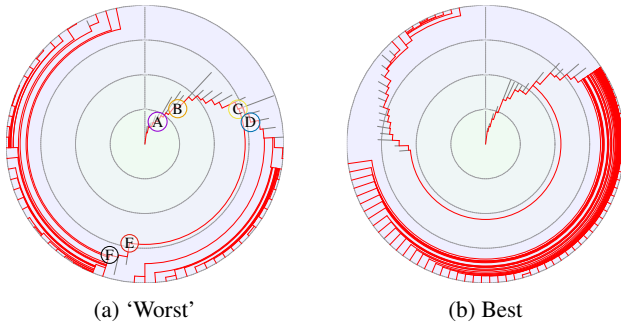


Figure 4: Phylogenetic trees for the parapatric runs with the most extreme speciations at $y=100$.

much insight on when speciation did occur.

To this end we use, instead, the simplified version in figure 2b which only shows the species on the survivor paths. Then we can easily see that very early in the simulation, around the 10th year, two species branched off from the main branch and, due to the harsh topological barrier, went on to further speciate in their own isolated plot of earth.

In order to better understand the type of genomic difference between individuals from different species we examined the morphologies produced during these simulations. However, as can be seen in figure 3a, even the most complex one is a far cry from what we could expect from an L-System. Indeed, always the minimalist one, natural selection only produced that which is essential and plainly ignored the structural organs (stem s and root trunk t), instead focusing its efforts on extracting nutrients from the environment (root hairs h , in gray, and leaf l , in green) in order to grow the maximal amount of flowers (f in red) so as to maximize its reproductive potential.

Still, some degree of morphological divergence were observed from individuals in the same simulation with sample plants from figure 3b being representatives of the most populated species on the left and right side of the mountain for a run with a good speciation score (minimal $c_r = 29.6\%$). Obviously, given the depth of structural complexity, these differences are not as striking as one could wish for.

Thus, the non-uniform locusts are to be found in other parts in the genome (metabolic values, compatibility functions, ...) were direct observation is much less straightforward and is left to future work.

Parapatric speciation

$$\begin{aligned}
 c_1 &= 1 \\
 \dot{t} &= .4\sin(.5Y\pi).5(\tanh(8(.5 - x)) + 1) \\
 c_0 &= \dot{y} = \dot{w} = 0
 \end{aligned}$$

A slightly more complex scenario involves the gradual apportion of a niche with no geological separation from the

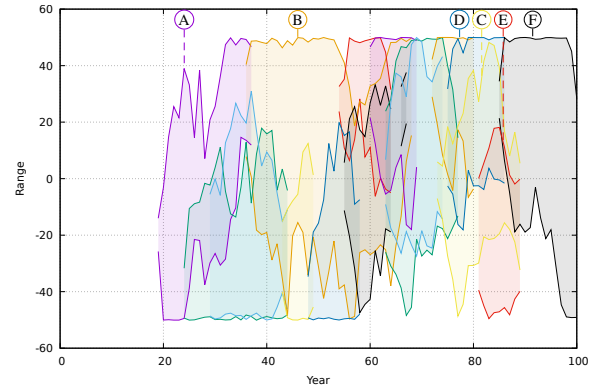


Figure 5: Colonization ranges for the 16 most populated species in a 'negative' run

rest of the environment. This implies that a contact zone exists between the two parts and thus that gene flow is not restricted by the abiotic component: speciation is left in the hands of the individuals themselves.

The left-hand side of the plot undergoes a gradual warming effect which, given the activation of constraint $C1$, also reduces the amount of available water.

Once more we refer to the relative compatibilities shown in figure 1 to assert that this experiment also produces divergences and clustering though more limited in range due to cross-breeding being expected but not enforced. One should also note that some simulations fail to colonize the harsher portion of the environment, thus degenerating into an evolution in uniform abiotic conditions.

The survivor-only version of our phylogenetic tree is displayed in figure 4b for the best scoring simulation ($c_r \in [35.7\%, 108\%]$) and it shows that the branching event that produced the two main strands occurred much later than in the previous experiment (slightly after the 50th year). Furthermore, the species density of these two branches is quite dissimilar with only the upper left portion accounting for those found off the desertic side. We could thus conclude that, to a weaker extent, the parapatric experiment successfully induced speciation.

However, the case of the worst scoring simulation ($c_r \in [89.1\%, 143\%]$) is much more interesting when looked at in more details. Indeed these c_r values show that not only reproductive isolation did not emerge in any significant proportion (even the term varieties might be too strong a word) but, on the contrary, there are cases of intense outbreeding: the 143% maximal relative compatibility indicates that for at least one species it is 1.5 more likely to reproduce with member of a foreign species than with more closely related mates.

We surmised that these results should come from a desertic species trying to gain ground into the temperature region by assimilating existing species and thus decided to look at

the dynamics of colonization. Summarized in figure 5 are the dynamics of the 16 more populated species generated by this ‘worse’ simulation. The height of a region depicts the range over which a given species has individuals alive at the end of the corresponding year which is why ranges can and *do* overlap.

Even broad analysis shows that, contrary to our hypothesis, the simulation has not degenerated into a champion-dominated situation. In fact, as time goes by and temperature diverges in the desert (lower part of the graph) and temperate regions (upper part) various dynamics emerge. During the first 18th year population count is too low to appear in the graph until species A emerges from a small region of the desert ($x \in [-26, -14]$). From there it quickly grows in range during the next years, colonizing the whole region and sending onward ‘scouts’ in the more temperate zone. This leads to migration, over the next decade, into the environment’s temperate portion where it is quickly overtaken by species B, an indirect descendant (see fig 4a).

Then starts a period of relative prosperity, where B has no real competition in its core range, so much that it regularly sends more ‘scouts’ back into the desert, though without much success. This era ends past the middle of the simulation (50th year) where it must, once again, share space with multiple, newly born challengers. This chaotic period lasts until about the 80th year with only three dominating species left: D in the temperate region, E in the desert and C their ancestral species. In time, D spawns a final species, F, which in about a year colonizes and dominates the whole right-side part of the environment. It takes little more than a decade for its influence to grow over the rest of the simulation into the desertic portion. Thus from the 98th year onward F is firmly anchored as a polyvalent species capable of thriving in a range of heat/water combinations, though one can see the start of a downward trend in its original biome.

These dynamics are not without similarities with those produced by natural selection in the real world which goes to show that, despite the simplicity of both the environment and the morphological adaptations displayed by its inhabitants much complexity still emerged. They also throw a measure of doubt on the metric used to broadly classify the results: despite being anchored in the pragmatic definition that a species is a “group of inter-breeding individuals reproductively isolated” we can see that it produced at least one (and probably many more) false negative.

Peripatric speciation

$$\begin{aligned} c0 &= 1 \\ \dot{y} &= .4\sin(.5Y\pi)(.5(\tanh(8(x - .5)) + 1) \\ &\quad + .5\text{gauss}(x, .5, .05)) \\ c1 &= \dot{t} = \dot{w} = 0 \end{aligned}$$

For the sake of completeness we briefly go into the de-

tails the last experiment performed: partial geological separation with niche subdivision which used the environmental parameters above.

The right side of the environment rises slowly from sea level up to a 20m high plateau which, due to the activation of constraint $c0$, is notably cooler than the adjacent lowlands. A small elevation in the center further separates both halves of the plot. This provides a more complex scenario which combines both of the previous approaches: on the one hand, the temperature differences stimulate generation of new shapes and exploration of genetic parameters while, on the other hand, the topological separation limits gene flow making it easier to keep true to the current evolutionary trend. In this case, however, the barrier is asymmetrical: as in the allopatric experiment, individuals at sea level have very limited chances to send seeds at such a remote altitude but plants on the plateau only have to cross the center elevation to disseminate their genetic material onto the lower half.

Given the intermediate nature of the setup, the fact that observed results, in terms of minimal/maximal relative compatibilities, are also intermediate does not come as a surprise. The topological asymmetry induces a slightly more dispersed distribution of relative compatibilities than in the parapatric case, as seen in figure 1. Conversely, these trends are inverted when compared with the purely continuous simulations.

There is, however, a point on which we can differentiate this experimental setting from the others as depicted in figure 6: the number of species.

Indeed the first produces an average of 2905 per run (1.1×10^6 plants, 849 generations), which is only marginally lower than the second one ($3228/1.09 \times 10^6/887$) and stays comparable with the third one ($4813/1.2 \times 10^6/898$). Even though these mean figures do not exhibit statistically significant differences, the distribution of values differ in a much more pronounced manner. While most runs for the allopatric speciation are clustered around the median and inter quantiles, runs in the peripatric experiment are more diffused, some reaching up almost to the next order of magnitude.

There is a similar trend with the number of generations but not the number of plants hinting that the lack of a strong geological separation promotes apparition of new species with roughly the same number of individuals by providing more noisy conditions.

Conclusion

In this work we set out to validate both our autonomous reproduction scheme and tool for automated phylogeny. To this end, we devised simple environmental settings that would mimic the natural conditions for known real-life cases of speciation.

Amidst the mass of data generated by our simulations, APOGeT managed to extract species trees which, when ren-

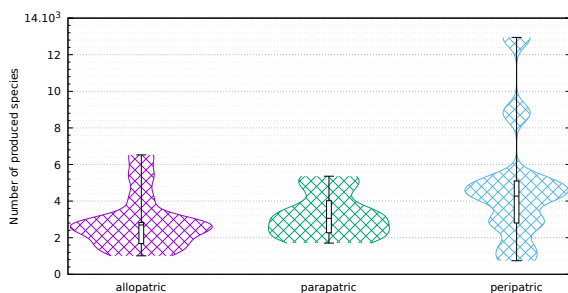


Figure 6: Number of generated species per experiment

dered into either their full or simplified forms, were instrumental in determining whether speciation emerged from the underlying plant-controlled reproductions. Though this process of natural selection did not feel the need to complexify the morphologies to any great extent, the dynamics exhibited on the species level were much more diverse and intricate, reminiscent of real-world ecosystem dynamics.

This paves the way for a very broad number of future works divided into two categories: investigation and complexification. Indeed despite the minimalist approach used to generate the test environments, the complete range of dynamics, competitions and inter-dependancies could not be fully investigated in this paper. Whether or not the situation described in the results of the parapatric experiment is a typical, favorable or below average case is left as an open question, pending further examination of the whole data set. Furthermore, the impact of individual genetic fields was only briefly examined, mostly regarding morphologies.

Additionally, using hand-crafted equations for generating environmental dynamics is not the most generic way to tackle the problem of environment-driven speciation. To this end, we plan to extend the presented model by using an evolvable substrate (CGP, GRN, ANN) as the basis for the environmental controller. This would allow for the automated generation of ecosystems displaying wider ranges of demeanors whether related to well-known examples of real-life equivalents or diverging into unfamiliar directions.

Source code

The C++ code for this project is available at <https://github.com/kgd-al> under the repositories Tools, APOGeT and ReusWorld

Acknowledgments

Computations for this work were performed using HPC resources from CALMIP (Grant P16043).

References

Adami, C. and Brown, C. T. (1994). Evolutionary Learning in the 2D Artificial Life System "Avida". *Artificial Life IV*, 1194:1–5.

- Bornhofen, S. (2008). *Emergence de dynamiques evolutionnaires dans une approche multi-agents de plantes virtuelles*. PhD thesis.
- Bornhofen, S., Barot, S., and Lattaud, C. (2011). The evolution of CSR life-history strategies in a plant model with explicit physiology and architecture. *Ecological Modelling*, 222(1):1–10.
- Butlin, R. K., Galindo, J., and Grahame, J. W. (2008). Sympatric, parapatric or allopatric: the most important way to classify speciation? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1506):2997–3007.
- Canino-Koning, R., Wiser, M. J., and Ofria, C. (2016). The Evolution of Evolvability : Changing Environments Promote Rapid Adaptation in Digital Organisms. *Proceedings of the European Conference on Artificial Life*, pages 268–275.
- Disset, J., Cussat-Blanc, S., and Duthen, Y. (2016). Evolved Development Strategies of Artificial Multicellular Organisms.
- Dubois, K., Cussat-Blanc, S., and Duthen, Y. (2017). Towards an Artificial Polytrophic Ecosystem. In *Morphogenetic Engineering Workshop, at the European Conference on Artificial Life (ECAL) 2017 September 4*.
- Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*.
- Godin-Dubois, K., Cussat-Blanc, S., and Duthen, Y. (2019). Self-sustainability Challenges of Plants Colonization Strategies in Virtual 3D Environments. In Kaufmann, P. and Castillo, P. A., editors, *Applications of Evolutionary Computation*, pages 377–392. Springer International Publishing, Cham.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo in silico—a Model of a Gene Network Regulating Multicellular Development in 3D Space with Artificial Physics. *Alife*, pages 297–304.
- Metivier, M., Lattaud, C., Heudin, J.-c., and Universitaire, L. (2002). A Stress-based Speciation Model in LifeDrop characters. *Artificial Life*, pages 121–126.
- Miconi, T. (2008). In silicon no one can hear you scream: Evolving fighting creatures. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4971 LNCS, pages 25–36.
- Prusinkiewicz, P., Hammel, M., and Mech, R. (1995). The Artificial Life of Plants. *Artificial life for graphics, animation, and virtual reality*, 7:39.
- Sims, K. (1994). Evolving 3D Morphology and Behavior by Competition. *Artificial Life*, 1(4):353–372.
- Singh, B. N. (2012). Concepts of species and modes of speciation. *Current Science*, 103(7).
- Ventrella, J. (2005). GenePool: Exploring the interaction between natural selection and sexual selection. *Artificial Life Models in Software*, pages 81–96.
- Woodward, F. I. and Williams, B. G. (1987). Climate and plant distribution at global and local scales. *Vegetatio*, 69(1-3):189–197.

To Evolve or Not to Evolve? That is the Question

Alex Ellery¹, A. E. Eiben²

¹Department of Mechanical & Aerospace Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON. K1S 5B6.
Canada: aellery@mac.carleton.ca

²Department of Computer Science, Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam. Netherlands:
a.e.eiben@vu.nl

Abstract

To evolve or not to evolve? That is the question: whether 'tis nobler in the mind to suffer the slings and arrows of grey goo, or to deny evolution to a sea of self-replicators and by prevention control them? We have been developing a physical self-replicating machine concept for deployment on the Moon built from local resources on the Moon. Here, we are concerned with architectural issues - we specifically address the problem of uncontrolled replication. We propose a multitiered approach to prevent this: (i) denial of self-replication through the implementation of centralised mass manufacturing of replicators; (ii) denial of scarce sodium and chlorine from Earth acts as an Earth-controlled kill switch in preventing further replication; (iii) denial of centralised supplies of asteroidal metals (tungsten-nickel-cobalt-selenium) at the lunar south pole acts as a Moon-controlled kill switch; (iv) denial of online learning capacity through fixed neural weights; (v) denial of extended computing resources through the elimination of transmit communications between self-replicators; (vi) denial of evolutionary capacity by implementing error detection and correction (EDAC) coding. Two kill switches and EDAC provide the backbone to our approach that maintain self-replication capability.

Introduction

We have been developing a physical instantiation of a self-replicating machine concept for service on the Moon to robotically construct a lunar infrastructure at low cost using local resources (Ellery, 2015a, 2016, 2017). To date, most effort has been devoted to 3D printing certain crucial components: (i) electric motors which has progressed to near completion; (ii) active computational components (vacuum tube) which has yet to be achieved but efforts are ongoing. We are also concerned with an important architectural issue – that of the prevention of uncontrolled replication. In approaching this problem, we are mindful of the Royal Navy's hard-learned lessons during the Falklands conflict regarding layered air defence for individual ships and flotillas, of which there are four – air combat patrol (Sea Harrier/F35 Lightning), area air defence (Sea Viper), point air defence (Sea Wolf) and close-in weapons (Phalanx/Goalkeeper). We explore a similar multi-tiered approach as our defence strategy against uncontrolled replication: (i) denial of self-replication through the implementation of centralised mass manufacturing of replicators; (ii) salt contingency – denial of

scarce sodium and chlorine from Earth acts as an Earth-controlled kill switch in preventing further replication; (iii) tunicose contingency – denial of centralised supplies of asteroidal metals (tungsten-nickel-cobalt-selenium) at the lunar south pole acts as a Moon-controlled kill switch preventing further replication; (iv) denial of online learning capacity through fixed neural weights controls the machine's intelligence; (v) denial of extended computing resources through the elimination of transmit communications between self-replicators (receive only); (vi) denial of evolutionary capacity by implementing error detection and correction (EDAC) coding controls the machine's adaptability. We pay special attention to (i), (ii), (iii), (iv) and (vi).

3D Printer-Based Turing Machine – Denial of Online Learning

To address the problem of 3D printing computing machines, we revert to the original model of a computer. The Turing machine is a finite-state machine comprising a read/write head mounted onto an infinitely long tape divided into discrete squares. The Church-Turing thesis asserts that the mechanistic computations of a Turing machine define an algorithmic process. The Turing machine sequentially reads an infinitely long digital tape of cells. Symbols from a finite alphabet are inscribed on the tape which are read in sequence by the read/write head. The initial tape encodes a set of input data. The read/write head incorporates a finite memory of internal state transitions constituting the computer program of the Turing machine. The motion of the read/write head – the behaviour of the Turing machine - is determined by the symbol inscribed on each cell of the tape and the internal state of the machine. The symbol is overwritten by a replacement symbol and/or the read/write head moves one cell left or right according to the Turing machine's state transition function. The resulting tape encodes a set of output data. Different Turing machines are specified by different state transition functions. This simple machine implements a mathematical function that converts its input into an output – the Turing machine's mechanical procedure encapsulates the algorithm concept as a finite sequence of simple operations. Any specific Turing machine may be encoded as an input tape so a universal Turing machine can emulate any specific Turing

machine, i.e. a universal Turing machine can compute any computable function given the appropriate algorithm.

Our implementation of a Turing machine comprises an input tape represented by magnetic core memory, an output tape represented by an analogue neural net circuit, and a read/write head represented by a 3D printer. The 3D printer thus becomes a central component of a universal computation capability – it prints out hardware circuitry according to the program stored in magnetic core memory. Magnetic core memory uses ferrite magnetic cores (toroids) through which wires are passed to convey read and write signals. Each core stores one bit of information non-volatily as zero or one depending on the direction of the core’s magnetisation. The invention of the coincident current system enabled a small number of wires to control a large number of cores in 3D stacks. A large number of small ferrite toroidal cores are held on layers of XY grids of wires through the toroidal centres. Only where the combined magnetic field from X and Y lines cross exceeds a threshold will the magnetic polarity reverse. Magnetic core memory offers high reliability and was used for the Apollo Guidance Computer and Space Shuttle Flight Computers.

We must now consider the 3D printed output circuitry. We have adopted the vacuum tube as the basis of our active electronics. Vacuum tube devices are based on the generation of relativistic electron beams and their interaction with electromagnetic waves. A vacuum tube is simple in construction - a tungsten cathode that emits electrons attracted to a nickel anode controlled by a third nickel grid electrode encased in an evacuated glass or ceramic tube and linked by silicone or ceramic-insulated kovar wiring. Only a small number of materials are required which are readily extracted from lunar resources. However, vacuum tubes are bulky and present challenges for building complex computational circuits. The von Neumann architecture computer is based on the central processing unit (CPU). The core of the CPU is one or more arithmetic logic units (ALU). The ALU is a combinatorial logic circuit for performing arithmetic operations (addition, subtraction, increment/decrement and sign) and bitwise logical operations (AND, OR, EX-OR and NOT) on 4-bit, 8-bit, 16-bit, 32-bit or 64-bit data widths. For example, the modest embedded 8051 CPU comprises 2,200 logic gates. Modern computers comprise ~500 million logic gates. Data is stored in a variety of different memory locations which must be fetched as input data to the CPU and the results of which must be pushed back into memory. The basic operation of the von Neumann architecture is the fetch-decode-execute cycle which is wasteful in hardware footprint. Using vacuum tube-based circuitry based on the von Neumann architecture would require very large computers.

To prevent runaway growth in the computer footprint imposed by the vacuum tube, the output of our Turing machine is an analogue neural network that encodes a specific algorithm in hardware form. The complexity of a neural network increases only with the logarithm of the task complexity unlike the exponential increase in circuit complexity of digital architectures (Parberry, 1994). Neural networks are under development for general purpose intelligence - the SpiNNaker (spiking neural network architecture) project is based on combining a large number of digital ARM processors within a grid of switches to emulate a

vast neural network representing a small brain of $\sim 10^6$ neurons. A simple electronic ring circuit of neurons has been proposed that emulates the neural processing function of the neocortex (Hahnloser et al, 2000). There is the prospect of implementing robust albeit simple behaviours neurally - one of the simplest biological neural networks in a non-aquatic free-living animal is that of the nematode worm *C elegans*: it comprises 959 cells in total as a hermaphrodite (of which 302 are neurons) or 1031 cells in total as a male (of which 381 are neurons) with approximately 5000 synapses. This potentially gives us a minimum neural network size though an engineered version might be subdivided into subnetworks of more modest dimension. On a much smaller scale, analogue neural circuits offer rapid computation with some biological fidelity in reducing specific energy consumption (energy/neuron) but at the cost of a fixed neural architecture.

We have adopted a modified version of the Yamashita-Nakamura neuron (Yamashita & Nakamura, 2007) which comprises an input summing amplifier, an inverting amplifier (for a step function) and a comparator. The weights of each neuron are pre-trained offline to implement its desired behaviour. We have demonstrated a pre-trained two-neuron hardware circuit implementing a Braitenburg control architecture of BV2/BV3 class (Braitenburg, 1984) performing automatic obstacle avoidance on a simple desktop mobile robot. We have begun exploring the potential for augmenting hardware neurons with online learning circuitry (Larson & Ellery, 2015). There are several intriguing possibilities for learning circuitry (Winter & Widrow, 1988; Martinelli & Perfetti, 1991) but we assume that we do not implement such capabilities to prevent uncontrolled learning – nevertheless, we have the quandary of requiring weight adjustment to permit fine-tuning of analogue neural circuits to variations in physical manufacture against the denial of online learning to ensure that behavior is both known and controlled.

Centralised Manufacturing – Denial of Self-Replication Capacity and Kill Switches

The most aggressive approach to prevent runaway replication is denial of self-replication of productive capacity. For this approach, we consider two options (centralized versus distributed production) across two dimensions (variability versus no variability), yielding four different cases (Table 1):

	Centralised Production	Distributed Production
Identical Copies	Conventional factory (e.g. six-sigma)	Self-replicators without evolutionary variation
Mutated Copies	EvoSphere	Self-replicators with evolutionary variation

Table 1. Replicator population options

The conventional factory employs mass production and has been employed since the Industrial Revolution for the

worldwide production of goods. In traditional factories, goods are produced in which variations are minimized. Before distribution, the quality of the goods are checked, and indeed, one of the hallmarks of quality is the so-called six-sigma quality control protocol.

Regarding terminology, self-reproduction may be regarded as an inaccurate form of self-replication that permits variation in offspring (Adams & Lipson, 2009) but we use the overarching term self-replication here with or without evolution. The EvoSphere concept envisions an entire ecosystem of physically and behaviourally evolving robots in the physical environment (Eiben, 2015a). It comprises a birthing clinic, a nursery and a living arena. In the birth clinic, robotic machines are constructed from raw materials; in the nursery, they undergo an online learning phase for fine-tuning their behaviours to their bodies; and in the living arena, only successfully graduated robots that perform their desired function are selected and are permitted to replicate exactly. The EvoSphere imposes two apparently contradictory objectives: (a) reproduction with variation and selection is permitted to implement robot evolution in the real world; (b) a kill switch is implemented to prevent procreation of undesired variants by human operators. Robot reproduction is divided into two phases that differentiate between the robot genotype and the robot phenotype: (a) recombination of robotic genotypes is permitted without constraint; (b) production of the genotype-encoded robot phenotype is constrained. This constraint is imposed by permitting only a single centralised production centre for the physical construction of robots. The kill switch is implemented at the centralised production centre. If invoked, it shuts down all reproductive processes and, as a consequence, halts evolution. All prior variations of robot generations however are permitted to continue operation.

Lunar Ilmenite

$\text{Fe}^0 + \text{H}_2\text{O}$ or silicone oil in colloidal suspension \rightarrow ferrofluidic sealing
1000°C

$\text{FeTiO}_3 + \text{H}_2 \rightarrow \text{TiO}_2 + \text{H}_2\text{O} + \text{Fe}$ (Fe separated by liquation)
ilmenite $2\text{H}_2\text{O} \rightarrow 2\text{H}_2 + \text{O}_2$ (H₂ recycling)

$2\text{Fe} + 1.5\text{O}_2 \rightarrow \text{Fe}_2\text{O}_3/\text{Fe}_2\text{O}_3 \cdot \text{CoO}$ - ferrite magnets

Nickel-iron meteorites

Mond process:

	Alloy	Ni	Co	Si	C	W
$\text{Fe}(\text{CO})_5 \leftrightarrow 5\text{CO} + \text{Fe}$ (175°C/100 bar) \rightarrow	Tool steel				<2%	9-18%
$\text{Ni}(\text{CO})_4 \leftrightarrow 4\text{CO} + \text{Ni}$ (55°C/1 bar) \rightarrow	Electrical steel			3%		
$\text{Co}_2(\text{CO})_8 \leftrightarrow 8\text{CO} + 2\text{Co}$ (150°C/35 bar) \rightarrow	Permalloy	80%				
	Kovar	29%	17%	0.2%	0.01%	
$\text{W}(\text{CO})_6 \leftrightarrow 6\text{CO} + \text{W}$ \rightarrow S catalyst	Thermionic cathodic material					

$4\text{FeS} + 7\text{O}_2 \rightarrow 2\text{Fe}_2\text{O}_3 + 4\text{SO}_2$

(Troilite) $\text{SO}_2 + \text{H}_2\text{S} \rightarrow 3\text{S} + \text{H}_2\text{O}$

$\text{FeSe} + \text{Na}_2\text{CO}_3 + 1.5\text{O}_2 \rightarrow \text{FeO} + \text{Na}_2\text{SeO}_3 + \text{CO}_2$

KNO₃ catalyst

$\text{Na}_2\text{SeO}_3 + \text{H}_2\text{SO}_4 \rightarrow \text{Na}_2\text{O} + \text{H}_2\text{SO}_4 + \text{Se} \rightarrow$ photosensitive Se

$\text{Na}_2\text{O} + \text{H}_2\text{O} \rightarrow 2\text{NaOH}$ (recycle)

Lunar Orthoclase

$3\text{KAlSi}_3\text{O}_8 + 2\text{HCl} + 12\text{H}_2\text{O} \rightarrow \text{KAl}_3\text{Si}_3\text{O}_{10}(\text{OH})_2 + 6\text{H}_4\text{SiO}_4 + 2\text{KCl}$

orthoclase

illite silicic acid (soluble silica)

$2\text{KAl}_3\text{Si}_3\text{O}_{10}(\text{OH})_2 + 2\text{HCl} + 3\text{H}_2\text{O} \rightarrow 3\text{Al}_2\text{Si}_2\text{O}_5(\text{OH})_4 + 2\text{KCl}$

kaolinite (clay)

\rightarrow porcelain

$\text{H}_4\text{SiO}_4 \rightarrow \text{SiO}_2 + 2\text{H}_2\text{O}$

Versions of the kill switch through centralized production facilities have been proposed in approaches (ii) and (iii) to deny specific resources for self-replicators. In Fig 1, we present a lunar industrial ecosystem with recycling loops representing the required chemical processing to yield material feedstock for 3D printing of the self-replicator from lunar raw material. Self-replication requires precisely green chemistry (Anastas & Warner, 1998) in order to achieve the material closure implicit in an industrial ecology. A corollary of this is that an evolving and diverging population would be wasteful in physical resources unless the littered carcasses of failed evolutionary experiments were scavenged efficiently. For our lunar ecosystem, the loss of iron-nickel-cobalt alloy from asteroidal resources – which must be mined from special ore locations on the Moon – and the loss of NaCl imported from Earth due to its scarcity of the Moon effectively decimates the entire ecosystem. The loss of tunicose materials prevent the manufacture of ferrite magnets (and so motors, etc), tool steel, permalloy, kovar, thermionic cathodes and photosensitive elements (tunicose contingency). Loss of NaCl prevents the manufacture of AlNiCo magnets (and so motors, etc), photosensitive elements, silica for transparent glass, piezoelectric sensors, regolith binder, drilling mud, silicone plastics and oils, Metalysis FFC process anode and electrolyte regeneration (salt contingency). These two sets of kill switches – one on the Moon and the other on Earth – provide the last lines of defence to uncontrolled replication. An important proviso is that as the number of replicating units grows, central supply hubs become traffic bottlenecks. Although these contingencies are specific to the self-replicating machine proposed for the Moon (for instance, the salt contingency would not be possible on Mars), they illustrate the effectiveness of multiple resource-denial kill switches.

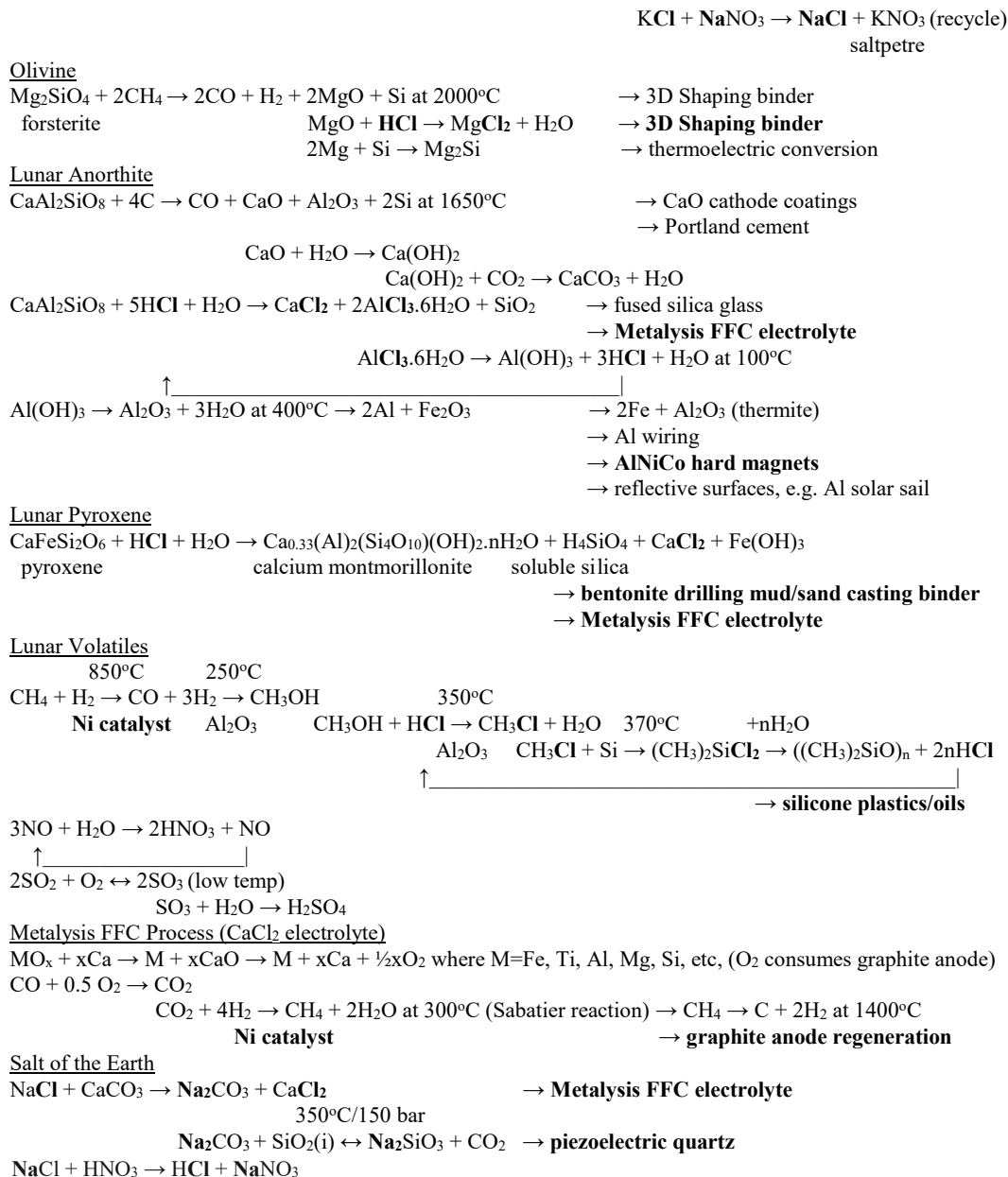


Fig 1. In-situ resource utilisation schematic with highlighted kill switches indicating loss of function

The full centralised replicator production facility approach denies the prospect of exponential growth of productive capacity (though the kill switches retain their applicability and effectiveness). A self-replicating machine offers unprecedented productive capacity by virtue of its exponential population P growth:

$$P = \sum_{i=1}^m (1+r)^i$$

where r=number of offspring per generation, i=generation number, m=number of generations. In a self-replicating machine, the number of offspring per generation, $r \geq 1$, e.g. our lunar application proposed $r=2$ and $m=13$ yielding $P \sim 2 \times 10^6$, i.e. the power of self-replication yields exponential growth in production. This is both its attraction and its prospect for jeopardy. Without restraint, robots can reproduce including their own reproductive capacity – this is equivalent to cell division in single cell populations, larvae in arthropods, egg laying in fish, reptiles and birds, and pregnancy in mammals. This introduces the potential for runaway replication with two specific dangers: (a) uncontrolled population growth (a macroscopic version of the grey goo scenario); (b)

uncontrolled and unwanted alterations in form and function of individuals. The dangers that this represents cannot be permitted (Eiben et al, 2012).

Error Detection & Correction Coding – Denial of Evolutionary Capacity

Despite the inadequacies of an operational definition of life to be a self-sustained chemical system capable of Darwinian evolution (Cleland & Chyba, 2002), we adopt it here with the corollary that any kind of self-replicator that copies genetic information is subject to Darwinian evolution due the genetic mutation (Ellery, 2018). Biological evolution has been broadly characterised by a growth in genetic complexity in that complex biological phenotypes require increased amounts of genetic information encoded from the environment in which they have evolved (Adami et al, 2000). Algorithmic complexity in a Kolmogorov sense may be regarded as the shortest bit sequence that can yield a given output. Although there are exceptions due to the C-value paradox (there is a fraction of non-coding genes that varies across species), genomic complexity reflects phenotypic complexity. At gene site i , there are four possible nucleotides with probabilities $(p_C(i), p_G(i), p_A(i), p_T(i))$ yielding an entropy per site:

$$H(i) = - \sum_{j=1}^4 p_j(i) \log p_j(i)$$

Hence, the maximum entropy per site is two bits due to base pair complementarity of A-T and C-G. This permits computation of the physical complexity of the organism as a whole by:

$$C = L - \sum_i H(i) \text{ where } L = \text{genome length (bp)}.$$

We expect this trend to continue in the self-replicating machine – evolutionary variation through generations implies a degree of uncontrollability. Our goal is to prevent evolutionary processes in the self-replicating machine to retain controllability. One proposal suggests that runaway self-replication of machines on Earth will yield at least 4°C temperature rise assuming thermal pollution within 2 years which would be readily detectable (Freitas, 2001). We aim to prevent uncontrolled replication in the first place so any self-replication scheme requires an error detection and correction (EDAC) coding strategy which we briefly review here (Griffith et al, 2005).

Biological self-replication employs template molecules to make copies of itself using building blocks in its environment. The invariant strands of alternating pentose and phosphate groups mount the four bases as rungs which form specific pairs A-T and G-C between purines (A and G) and pyrimidines (C and T). DNA bases A, T, G and C form a digital code with three-quad codons such as GCC which encodes the amino acid alanine. This provides 64 codons for only 20 amino acids providing redundancy in the genetic code, e.g. UC* codes for serine so the third position of the codon is a wildcard. There are others however that are uniquely coded, e.g. UGG is the unique code to tryptophan.

In its normal packaging state, DNA is coiled up in chromatin proteins which unrolls the sticky DNA strings rapidly for copying and re-rolls it after copying. DNA helicase separates DNA into two strands in preparation for replication. DNA polymerase then creates two double stranded DNA strings from the two separated single strands. The average copying error rate in human DNA is $\sim 10^{-8}$ but this varies with the sensitivity of the gene to allelic variation – histone genes which code for DNA packing proteins appear almost invariant to mutation across biological domains. The most significant treatment of information theory to genetics is the visionary monograph by Hubert Yockey that deserves wider recognition (Yockey, 1992).

The evolutionary pressure for genetic parsimony favours the short overlapping genes of low complexity eukaryotic viruses and DNA phages but favours non-overlapping modular genes in higher organisms (Ofria & Adami, 2002). Overlapping genes with multiple expression require slower evolutionary change. Neutral mutations occur in the third nucleotide of a codon afforded by coding redundancy but overlapping genes have offset reading frames making neutral mutations impossible. Larger genomes cannot exploit overlapping genes. The conundrum of replication copying fidelity requiring high genomic complexity (length) during early life before such high copying fidelity could evolve is referred to as the “error catastrophe” (Joyce, 2002). At higher mutation rates, genotypes of higher mutational robustness with lower replication rates are favoured irrespective of replication fidelity per genome $F = e^{-RL}$ where R=error rate per base pair, L=genome length (base pairs) (Wilke et al 2001). The vast majority of eukaryotic DNA is non-coding – in humans, only 3% of the genome is active, the other 97% being pseudogenes, etc. In eukaryotes, there is “junk” DNA (introns) that is excised from mRNA before being translated into proteins. Introns are marked for excision by a start sequence GT and a stop sequence AG after being looped to bring the active flanking genes into proximity and then cut out. Noncoding regions of the eukaryotic genome exhibit long range correlations (Buldyrev et al, 1995). There are epigenetic mechanisms for switching out genes - methylation of DNA either attaches methyl groups directly to DNA or modifies histones which dictate the activation status of genes.

In biology, the error correcting process during DNA replication involves proofreading (known as 3’-5’ end exonuclease) (Battail, 2004). In DNA, error detection and correction is accomplished with polymerases which check the complementary fit between base pairs. In bacteria, the three DNA polymerases I, II and III progress along the growing dual DNA strand from the 5’ end to the 3’ end, recognise incorrect bases, reverse direction from the 3’ end to the 5’ end, excise the incorrectly matched base and re-insert the correct base. There are other repair mechanisms for base excision repair, mismatch repair, strand break repair, cell cycle checkpoints and cell apoptosis. There are many environmental disruptions that can occur in cells: UV radiation can cause cytosine and thymine to fuse distorting the DNA shape which effectively marks the region to be excised; deamination converts the GC base pair to an AT base pair which can be corrected by DNA glycosylases; oxidised guanine emulates thymine and must be replaced. These are point mutations. Deletions and insertions of single base pairs

generate frameshift mutations which cause entire reading frames to become shifted. Similar frameshifts occur when entire sections are deleted, copied or inverted. Transposons are sections of DNA that cut-and-paste themselves out of and into different locations. In each case, it is the complementary base that provides the reference datum, detectable helix distortion, diploid genome copies and predictable corruptions that permit repair. The repair mechanisms however are not perfect but species such as *Deinococcus radiodurans* can survive extreme radiation environments by using an average of four to ten copies of its genome (evolved to cope with extremely dry conditions) (Battista, 1997). This is a form of repetitive coding. Such a form of coding may be employed through multiple gene copies – triple (or higher order n-tuple) modular redundancy with voting logic is common in safety critical systems in spacecraft. An example of such a safety critical function would be a Hayflick limit on the number of generational copies that a self-replicating population can produce, e.g. emulating telomere shortening as a counter.

For our self-replicating machine, it is essential that codec (coding/decoding) can be performed using simple circuitry and/or neutrally using analogue neural network circuits. We assume that genetic information is transmitted vertically through generations and hierarchically through the population as it increases (Battail, 2010). This constitutes a noisy and bursty communications channel for the transmission of genetic information. The maximum information transmission rate through a communication channel is given by Shannon's coding theorem:

$$R < B \log_2(1 + \text{SNR})$$

where B=bandwidth and SNR=signal-to-noise ratio. One way to transmit error-tolerant messages is to transmit the message f times (repetition coding) – if f=3, triple modular redundancy permits a simple majority voting logic. This is highly inefficient. A more efficient way is to add parity bits to the message data. Any kind of channel code adds structured redundant bits increase the fidelity of information transmission at a cost of higher bandwidth requirements (Berlekamp, 1980). Typically, error detection and correction (EDAC) codes are usually implemented in hardware using extra memory bus bits and encoding/decoding circuits. The data lines of the EDAC bus connect directly to RAM. Address lines to the memories are buffered by latches which synchronise the address to the system clock allowing synchronous burst of instruction and data caches. The SNR can be related to normalized signal-to-noise ratio per bit E_b/N_0 by:

$$E_b/N_0 = (S/N)/(R/B)$$

Above the theoretical Shannon coding limit, there is a code that can communicate with zero error (Costello & Forney, 2007):

$$E_b/N_0 > (2^{R/B} - 1)/(R/B) = \ln 2 = -1.6 \text{ dB}$$

A typical bit error rate (BER) used in spacecraft communications is 10^{-6} (less stringent than the biological BER of 10^{-8}) but the BER will depend on the genome size for the self-replicator. A BER of 10^{-9} is routinely achievable in space systems and that a BER of 10^{-15} is desirable – with a maximum population limit of 10^6 self-replicating units, this gives a standard BER of 10^{-9} per machine. The closest to the Shannon coding limit achievable are turbo codes which are formed by the parallel concatenation of two recursive codes separated by an interleaver code (Berrou et al, 1993). The interleaver is the crucial aspect as it implements a pseudo-random code. Turbo-codes are complex and are unlikely candidates for biological implementation. The extensive tandem repeats and introns in the eukaryotic genome and especially the human genome may be implementing error detection and correction codes as parity bits. One of the simplest error detection and correction codes is the Hamming code such as the Hamming (7,4) code which can correct a single-bit error but detect one-bit and two-bit errors. It has been suggested that exon-intron genes are Hamming codewords (Faria et al, 2012). There are two main types of EDAC other than turbo codes – block codes and convolutional codes. In an (n,k) linear block code, there are k information bits (input block) and n-k parity bits for n message bits in total (output block) with a code rate of $r=k/n$ (Bhargava, 1983). There are 2^k possible different messages of length k that are mapped onto 2^n codewords of length n. The summations require modulo-2 arithmetic without carries which can be implemented through the memoryless EX-OR circuit. The Hamming weight of a code word $w(c)$ is the number of nonzero components to the code word. The code can correct any pattern of e or fewer random errors provided $2e+1 \leq d$ where d =Hamming distance between two codewords (number of different elements in the codewords). The primary goal of block coding is to maximise the Hamming distance between codewords. The commonest used block codes are the Bose-Chaudhuri-Hocquenhem (BCH) codes and Reed-Solomon (RS) codes with the following parameters (Table 2):

	Block length	Number of coding bits	Hamming distance
Hamming code	$n=2^m-1$	$k=m$	$d=3$
BCH code	$n=2^m-1$ for $m=3,4,5,\dots$	$k \geq n-me$	$d \geq 2e+1$
RS code	$n=2^m-1$ symbols with m bits/symbol	$(n-k)=2e$ symbols with m bits/symbol	$d=2e+1$ symbols

Table 2: Properties of some common block codes

The Reed-Solomon code is a type of non-binary BCH code. The Reed-Solomon code is of particular interest because it offers the maximum Hamming distance for an (n,k) code with Hamming distance $d=n-k+1$ and can correct bursts of e symbol (m-bit) errors per codeword (Berlekamp, 1982). This is because an m-bit burst is concentrated into a single symbol error. The BCH code is decoded by an iterative Berlekamp-Massey decoding algorithm (Imamura & Yoshida, 1987). There is evidence that block codes such as BCH codes appear

to be implemented in evolutionarily ancient gene sequences of the *Arabidopsis* brassica flowering plant (Brandao et al, 2015). The Golay code is a “perfect” three-error block code that works only for (23,12) for d=7 and (24,12) for d=8 – it is based on a remarkable number theoretic relation:

$$\frac{2^{23}}{\binom{23}{1} \binom{23}{2} \binom{23}{3}} = 2^{23} / 2^{11} = 2^{12}$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ = binomial coefficients. It would be curious indeed if evolution had discovered the tri-error Golay code for the tri-base genetic code.

A convolutional code of code rate 1/r is a type of trellis code which can be generated by a sequential k-stage shift register with r modulo-2 adders. Convolutional codes do not segment the information stream into blocks but add redundant bits continuously and so requires a memory of order m. Each branch in the decoding trellis is labelled with an n-bit output block, so there are 2^n branch metrics. A common coding protocol is a constraint length of k=7 and code rate r=1/2 for decoding efficiency. Convolutional codes are decoded using the Viterbi algorithm which is a maximum likelihood decoder (Forney, 1973). A shift register represents every state in the decoding trellis. The complexity of Viterbi decoders is exponential with the constraint length of the code. An artificial neural network Viterbi decoder based on analogue neurons has been demonstrated (Wang & Wicker, 1996). It implemented discrete connections weights (+1,-1) to eliminate the need for network training. Neurons represented trellis elements and selected the maximum trellis path at each pass and updated the path metrics through feedback connections. It was a locally connected network to minimize its complexity. 2^n neurons are required for an n-bit input while 2^m neurons are required for m encoding feedback connections (m=k-1). For each state, $2(2^k-1)$ neurons are required to find the maximum metric of the 2^k paths. Thus, the total number of neurons required is $n=2^{m+k}+2^n-2$. For a r=1/2, k=7 convolutional code, n=514 neurons offering a much smaller footprint than a digital ASIC. The neural Viterbi algorithm performed significantly faster than a digital implementation due to its parallel architecture.

The CCSDS (consultative committee for space data systems) standard for spacecraft recommends concatenation of two EDAC by interleaving an inner (7,1/2) convolutional code (applied first) with an outer Reed-Solomon (255,223) block code (applied last) for high data rate telemetry downlinks or BCH (63,56) for low data rate command uplinks. This is a specific example of Battail’s nested code characterising aspects of the biological genetic code such as highly conserved HOX genes (Battail, 2008). HOX genes determine head-tail topological structure through morphological gradients and have diverged little since the emergence of multicellular organisms in the Cambrian explosion 540 My ago. The evolutionary rate can be controlled to significantly reduce evolutionary divergence. The CCSDS protocol illustrates that EDAC may be nested multiple times to give an arbitrary error rate, though of course at the cost of memory consumption. Resources devoted to EDAC during copying from n random components in the environment increases linearly as error rate decreases exponentially as $(1-e)^n$ (Griffith et al, 2005).

Conclusions

It appears that evolutionary divergence in a growing population of self-replicating machines are inevitable but prudence dictates that a multi-tiered system of safeguards should be adopted. The first layer of defence is the implementation of costly modular redundancy (say, five gene copies emulating the five modular redundancy of the integrated computers onboard the Space Shuttle) and multiple recursive layers of EDAC to reduce evolutionary divergence. The second layer of defence is the prevention of online learning through fixed weight neural networks – related to this is minimisation of social interaction through fixed communication protocols (which we have not addressed here). The third line of defence constitutes two layers of kill switches that are self-replicating machine specific – on the Moon, controlling access to centrally mined tungsten-nickel-cobalt-selenium from asteroidal resources, and on the Earth, denial of the reagents sourced from NaCl that must be transported from Earth. The final, most aggressive proposal is centralisation of all mass production facilities but this drastically reduces the attractive aspects of self-replication so we do not consider this to be practical.

A more philosophic issue is that by implementing EDAC we are effectively halting the evolutionary process by introducing high levels of copying fidelity. If life is defined as a self-sustained chemical system capable of undergoing Darwinian evolution (Luisi, 1998), there are three plausible interpretations of this definition. The first is that by denying evolutionary processes, our self-replicator is no longer alive as it is no longer subject to evolutionary processes. The second is that we have only suppressed the evolutionary process but not the capacity for evolutionary development, so it is alive. The third is that this evolutionary suppression is not absolute but based on bit error rate – later rather than sooner, copying errors will arise if the population grows beyond any imposed Hayflick limit, i.e. we have slowed evolution rather than halted it. Of course, the first interpretation states that this is not possible because of the integrity of the Hayflick limit imposed by EDAC. We have come full circle... A similar situation occurs in attempts to suppress evolution in synthetic biological organisms (Scharck, 2012).

Experiments in Avida indicate that self-replication does not guarantee evolvability (LaBar et al, 2015). Within Avida, a lack of evolvability can occur if all possible mutations to a specific genetic sequence prevent further self-replication. The closest biological organism in which this occurs is the mule which is a horse-donkey chimera that is nominally sterile; it might occur in engineered systems in which evolutionary brittleness is a consequence of the genetic encoding system – this has been explored in genetic algorithms to reduce brittleness through cellular encoding of embryonic development in hardware systems (Eiben & Smith, 2015b) such as genetic programming trees (Funes & Pollack, 1998) or L-systems. These represent growth processes which are tolerant of mutations but direct encodings are much more brittle rendering the possibility that engineered self-replicators may be designed so that they always mutate into dysfunction (most mutations are in fact dysfunctional – here, all mutations would be so).

Another interesting option concerns whether learning capability and evolutionary capacity might be permitted but moderated. This introduces the notion of shaping the learning or evolutionary process. In neural networks, this requires initialization of the network weights to incorporate pre-defined structures prior to learning (innate knowledge). Symbolic connectionism incorporates expert system-based structures into neural networks (Ellery, 2015b). Bayesian networks are particularly suitable as a priori neural network knowledge which imposes structure to any subsequent learning. In genetic algorithms, such shaping is imposed through the fitness function – usually a simple metric, there is no reason why it cannot become more prescriptive – indeed, the design of a planetary rover’s chassis (number of wheels, wheel radius and width, grouser size, vehicle weight, etc) was successfully evolved using a fitness function that implemented maximization of drawbar pull computed through a Bekker-Wong terramechanics model (Setterfield & Ellery, 2010 *unpublished data*). In effect, the fitness function substituted for the environment. The question is how to implement such fitness functions into a self-replicator to control the direction of its evolution.

These issues are, as yet, unexplored but warrant further investigation. The chief concern must be to what extent learning and evolution can be shaped and controlled. We are skeptical – until the advent of further evidence - that full control can be exerted because partial control is no control – indeed, partial control is more dangerous than no control because it offers an illusion of control where there is none.

References

- Adami C, Ofria C, Collier T (2000) “Evolution of biological complexity” *Proc National Academy Sciences* **97**, 4463-4468
- Adams B, Lipson H (2009) “Universal framework for analysis of self-replication phenomena” *Entropy* **11**, 295-325
- Anastas, P, Warner J (1998) *Green Chemistry: Theory and Practice*, Oxford University Press: New York
- Battail G (2004) “Engineer’s view on genetic information and biological evolution” *BioSystems* **76**, 279-290
- Battail G (2008) “Information theory and error-correcting codes in genetics and biological evolution” in *Introduction to Biosemiotics* (ed. Barbieri M), Springer Publishers, 299-345
- Battail G (2010) “Heredity as an encoded communication process” *IEEE Trans Information Theory* **56** (2), 678-687
- Battista J (1997) “Against all odds: the survival strategies of *Deinococcus radiodurans*” *Annual Reviews Microbiology* **51**, 203-224
- Berlekamp E (1980) “Technology of error-correcting codes” *Proc IEEE* **68** (5), 564-593
- Berlekamp E (1982) “Bit-serial Reed-Solomon encoders” *IEEE Trans Information Theory* **28** (6), 869-874
- Berrou C, Glavieux A, Thitmajshima P (1993) “Near Shannon limit error-correcting coding and decoding: turbo-codes” *Proc IEEE Int Conf Communications*, paper no 397441
- Bhargava V (1983) “Forward error correction schemes for digital communications” *IEEE Communications Magazine* **21** (1), 11-19
- Braitenburg V (1984) “Vehicles: Experiments in Synthetic Psychology” *MIT Press*
- Brandao M, Spoladore L, Faria L, Rocha A, Silva-Filho M, Palazzo R (2015) “Ancient DNA sequence revealed by error-correcting codes” *Scientific Reports* **5**, 12051
- Buldyrev S, Goldberger A, Havlin S, Mantegna R, Matsu M, Peng C-K, Simons M, Stanley H (1995) “Long-range correlation properties of coding and noncoding DNA sequences: GenBank analysis” *Physical Review E* **51** (3), 5084-5091
- Cleland C, Chyba C (2002) “Defining life” *Origins of Life & Evolution of the Biosphere* **32**, 387-393
- Costello D, Forney D (2007) “Channel coding: the road to channel capacity” *Proc IEEE* **95** (6), 1150-1177
- Eiben A (2015a) “EvoSphere: the world of robot evolution” *Proc Theory & Practice of Natural Computing LNCS 9477* (ed. Dediu A-H et al), Springer Publishers, 1-17
- Eiben A, Smith J (2015b) “From evolutionary computation to the evolution of things” *Nature* **521**, 476-482
- Eiben A, Kernback S, Haasdijk E (2012) “Embodied artificial evolution: artificial evolutionary systems in the 21st century” *Evolutionary Intelligence* **5** (4), 261-272
- Ellery A (2015a) “Engineering artificial extraterrestrial life?” *Proc European Conf on Artificial Life (Late Breaking)*, York, UK, 12-14
- Ellery A (2015b) “Artificial intelligence through symbolic connectionism – a biomimetic rapprochement” in *Biomimetic Technologies: Principles & Applications* (ed. Ngo D), Elsevier Publishing
- Ellery A (2016) “Are self-replicating machines feasible?” *AIAA J Spacecraft & Rockets* **53** (2), 317-327
- Ellery A (2017) “Building physical self-replicating machines” *Proc European Conf on Artificial Life*, Lyon, France, 146-153
- Ellery A (2018) “Engineering a lunar photolithoautotroph to thrive on the Moon – life or simulacrum?” *Int J Astrobiology* S1473550417000532
- Faria L, Rocha A, Kleinschmidt J, Silva-Filho M, Bim E, Herai R, Yamagishi M, Palazzo R (2012) “Is a genome a codeword of an error-correcting code?” *PLoS One* **7** (5), e36644
- Forney D (1973) “Viterbi algorithm” *Proc IEEE* **61** (3), 268-278
- Freitas R (2001) “Some limits to global ecophagy by biovirus nanoreplicators with public policy recommendations” *reprint*
- Funes P, Pollack J (1998) “Evolutionary body building: adaptive physical designs for robots” *Artificial Life* **4** (4), 337-357
- Griffith S, Goldwater D, Jacobson J (2005) “Self-replication from random parts” *Nature* **437**, 636
- Hahnloser R, Sarpeshkar R, Mahowald M, Douglas R, Seung S (2000) “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit” *Nature* **405**, 947
- Imamura K & Yoshida W (1987) “Simple derivation of the Berlekamp-Massey algorithm and some applications” *IEEE Trans Information Theory* **33** (1), 146-150
- Joyce G (2002) “Booting up life” *Nature* **420**, 278-279
- LaBar T, Adami C, Hintze A (2015) “Does self-replication imply evolvability?” *Proc 13th European Conf Artificial Life*, 596-602
- Larson S & Ellery A (2015) “Trainable analogue neural network with application to lunar in-situ resource utilisation” *Proc Int Astronautics Federation Congress*, Jerusalem, IAC-15-D3.3.6
- Luisi P (1998) “About various definitions of life” *Origins of Life & Evolution of Biospheres* **28**, 613-622
- Martinelli G & Perfetti R (1991) “Circuit theoretic approach to the backpropagation learning algorithm” *IEEE Int Symp Circuits & Systems* **3**, 1481-1484
- Ofria C, Adami C (2002) “Evolution of genetic organisation in digital organisms” in *Evolution as Computation* (ed. Landwehr L, Winfree E), Springer Publishing, 296-313
- Parberry I (1994) *Circuit Complexity and Neural Networks*, MIT Press Foundations of Computing, Cambridge, MA
- Schark M (2012) “Synthetic biology and the distinction between organisms and machines” *Environmental Values* **21**, 19-41
- Wang X & Wicker S (1996) “Artificial neural net Viterbi decoder” *IEEE Trans Communications* **44** (2), 165-171
- Wilke C, Wang L, Ofria C, Lenski R, Adami C (2001) “Evolution of digital organisms at high mutation rates lead to survival of the flattest” *Nature* **412**, 331-333
- Winter R & Widrow B (1988) “Madaline II: a training algorithm for neural networks” *IEEE Int Conf Neural Networks*, 401-408
- Yamashita Y, Nakamura Y (2007) “Neuron circuit model with smooth nonlinear output function” *Proc Int Symp Nonlinear Theory & its Applications*, Vancouver, 11-14
- Yockey H (1992) *Information Theory & Molecular Biology*, Cambridge University Press

Evolutionary rates of information gain and decay in fluctuating environments

Nicholas Guttenberg¹

¹Earth-Life Science Institute, Tokyo Institute of Technology, Tokyo, Japan
ngutten@gmail.com

Abstract

In this paper, we wish to investigate the dynamics of information transfer in evolutionary dynamics. We use information theoretic tools to track how much information an evolving population has obtained and managed to retain about different environments that it is exposed to. By understanding the dynamics of information gain and loss in a static environment, we predict how that same evolutionary system would behave when the environment is fluctuating. Specifically, we anticipate a cross-over between the regime in which fluctuations improve the ability of the evolutionary system to capture environmental information and the regime in which the fluctuations inhibit it, governed by a cross-over in the timescales of information gain and decay.

Intuitively, evolution is a process by which populations learn about the world. As such, information appears to be a natural concept for constructing an abstracted view of evolutionary process — how much information does the population have about the environment, how much information is retained between generations, how much more information is needed before the population can find a fitness optimum, and how much of that per generation does selection manage to bring into the population? Along the lines of (Adami, 2004), these ideas can be connected to specific information theoretic quantities and bounds. However, while such information theoretic quantities can be measured after the fact, we would like to see whether a formulation of evolutionary dynamics in terms of information is sufficiently resolved to be predictive of the behavior of that evolving system in different conditions. If we were to reduce the details of an evolving system's state to a set of information quantities, could we find equations of motion purely in terms of those variables which would still predict the future dynamics of that evolutionary process to some degree?

In this paper, we target the response of evolving systems to fluctuating environments in order to investigate this idea. Specifically, can we predict when fluctuations would help or hinder a population's ability to adapt, simply by looking at the dynamics of the information between population and environment? The literature on fluctuating environments has examples of both cases. In terms of positive contributions,

they can accelerate adaptation and even increase the asymptotically achievable fitness (Kashtan et al., 2007). It has also been proposed that the richness of open-ended evolution may depend on (or even originate from) the correspondingly richer problems posed by needed to be well-adapted to a multiplicity of environments (Neumann Jr, 1997; Bedau et al., 2000). Environmental variations have also been considered as a driving source behind multi-level organization and generalization properties of evolutionary systems (Medernach, 2017). At the same time, it is observed that environmental variations may induce evolving systems to make tradeoffs between optimality and robustness (Levins, 1967; Wilke et al., 2001), and adaptation between a succession of sufficiently unrelated environments can interfere with adaptation (Steiner, 2012).

In order to try to understand these tradeoffs, we consider a simplified case in which there are two independent environments given by random variables E_1 and E_2 , such that the population of organisms \vec{g} can have some mutual information with them $I(g; E_1)$ and $I(g; E_2)$. Here, we further specialize to the case in which these environments have no mutual information with each-other $I(E_1; E_2) = 0$, and that mutation operates independently from the environmental random variables. Based on the constraint that the only way for $I(g; E)$ to increase is through selection, we can construct a simple model in which while the population is being exposed to E_1 , $I(g; E_2)$ only has decreasing terms, and vice versa. If we then average over a cycle including both E_1 and E_2 , we can consider when the gain of mutual information during the selection phase would be balanced against the loss of information during the neutral phase. For a rapidly varying environment, this balance point should only depend on the instantaneous rates of change of the mutual information around the population's steady state, whereas for slowly varying environments, the overall shape of the trajectory may lead to systematic variation in things like the rates of increase and decrease.

We make a further assumption — that the underlying mechanisms responsible for determining the rate of increase and decrease of the mutual informations are intrinsic to the

dynamics of replication, mutation, and selection and should be the same both in slowly varying and quickly varying environments. If this is the case, then the balance point between information gain and loss in the rapidly fluctuating environment can be related to observations of the same evolutionary system made when the environments vary at a different rate. However, if for example the fluctuations led to significantly different population structures in steady state, then that would violate this assumption. We will ultimately return to this point, as it is likely that this does in fact occur, and may be connected to the relationship between fluctuating environments and the evolution of evolvability (Ofria et al., 2016).

Related Work

The sense in which evolved organisms contain information about the world has been explored in a variety of ways. One approach centers on looking at the Shannon information, concerned primarily with measures of the entropy of different parts of the genome (Schneider, 2000; Chang et al., 2005). This naturally extends into methods which use internal mutual informations between different bases, protein residues, and structures as a way of understanding things such as co-evolution and structured variation (Göbel et al., 1994; Martin et al., 2005; Gloor et al., 2005). These studies often relate to understanding the dynamics of neutral evolution. On the other hand, there is the question of what if anything a given portion of genetic entropy is 'about' — that is to say, not just whether it varies, but whether it has information about variations that exist within the environment or context of the organism (Adami, 2004). While identifying the random variables which characterize a given environment in nature is in general ambiguous, cases of co-evolution between competing sets of sequences can make this concrete by treating the sequences of one population (or corresponding phenotypic variations) as the environment of the other population and vice versa. For example, (Xia et al., 2009) looks at mutual information in the co-evolutionary dynamics between a virus and the immune response as a way to understand immune escape.

Evolutionary simulations

We consider a simple evolutionary simulation where each organism consists of a binary string \vec{g} of length L ($L = 50$ for all simulations presented in this paper), and each environment consists of a pair of binary strings: the target sequence \vec{E} , and the mask \vec{m} . The elements of \vec{E} are randomly 0 or 1 with equal probabilities, whereas the elements of \vec{m} are 1 with probability Γ and 0 otherwise, where Γ measures the fraction of the organism's capacity a given environment can be expected to take up. The relative fitness of each or-

ganism in the population is then given by:

$$F = 1 + \sum_i^L \vec{m}_i \left(1 - |\vec{g}_i - \vec{E}_i|\right) \quad (1)$$

This means that for a given environment there are a subset of specific sites (given by m_i) which have non-neutral fitness effects, and each site has an independent preferred value of the genome given by \vec{E}_i . The fitness is then linear in the number of sites which are matched. The independence between sites is chosen in order to enable a simplified sense of the mutual information to be used, so that we can factorize the overall joint distribution over sequences into a product of distributions at each site. Each generation, fitnesses are evaluated and a new population is composed by randomly sampling from the old population with replacement in proportion to fitness, so that the population size remains constant ($N = 200$ in all simulations reported in this work). Mutation occurs with a per-base probability μ/L .

In this system, we wish to consider the mutual information between the population of sequences and the environment, and how it changes over time. The mutual information between random variables x and y is defined:

$$I(x; y) \equiv \int p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dx dy \quad (2)$$

This quantity places a bound on the ability to better infer x given an observation of y compared to not having that observation of y — that is to say, if one knows $p(x)$ and then observes y , the mutual information measures the change in the entropy in the possibilities that x may take conditioned on knowing y . The quantity is symmetric, such that the same would be true for inferring y by predicting x . In the context of organisms and environments, the mutual information is not a quantity which would be defined in a single evolutionary trajectory but instead is a statistical property over entire ensembles of trajectories associated with a distribution over the different environments. We cannot ask 'what is the mutual information between this one organism and its environment?', but we can ask 'what is the mutual information between organisms and their environments in this evolutionary context?'

Measuring the mutual information directly in practice requires accumulating sufficient observations in order to construct estimates of $p(x, y)$, $p(x)$, and $p(y)$ via sampling. If the sum of the dimensions of x and y is large, this becomes prohibitively expensive to brute force as the number of samples required grows exponentially in the dimensions of the variables involved. However, due to the relationship between the mutual information and the bound on what can be inferred about the variables by observing each-other, any method of approximate inference can be used to generate a lower bound on the mutual information between variables. For example, there are techniques to use neural networks

to estimate mutual informations between high dimensional random variables (Belghazi et al., 2018). These methods could be used to extend this type of analysis to systems with arbitrary genotype-phenotype maps including the effects of epistasis, or even to systems where the information-carrying degrees of freedom are not a priori known such as chemical reaction networks.

Because our simulations consider only fitness landscapes without epistasis, we can consider the mutual information independently on per-base basis. That is to say, the relationship between each base in the genome and the corresponding base of the environmental random variable in the fitness function is statistically independent from all of the other bases, and each base only interacts with the corresponding base of the environmental random variable. Any interactions between bases occur only through aggregation into the fitness, and such interactions are independent of the specific environment sequences — that is to say, by observing the aggregate fitness, we would receive zero information as to the value of any particular base in the environment string. As such, the total mutual information between genome and environment is just the sum of per-base mutual informations, and we can reduce the multidimensional mutual information estimation problem into a collection of independent one-dimensional estimations for which direct sampling is sufficient.

Results

We measure the per-base mutual information by sampling over 5000 runs of the evolutionary simulation with independent random choices for the environments in each run. In order to sample the probabilities $p(x, y)$, $p(x)$, and $p(y)$ for a given base i , we first take the subset of runs in which that base is not masked out in the target environment. Then, from that set of runs, we measure the fraction of the population in each run which contains a 1 at that site $\bar{g}_i = \langle \vec{g}_i \rangle$. This results in approximately 5000Γ scalar values for each base. These values are quantized into a histogram with 100 bins between $[0, 1]$, and we then accumulate samples from the unmasked runs to estimate $p(\bar{g}_i, \vec{E}_i)$, $p(\bar{g}_i)$, and $p(\vec{E}_i)$. With these discrete distributions, we can directly evaluate the mutual information between population and environment as a function of time.

A run of the simulations involves first letting the population adjust to a ‘burn in’ environment for 200 generations where data are not taken, followed by 200 generations in environment E_1 and 200 generations in environment E_2 (the mask vectors also vary between these environments). We then measure the informations $I(\bar{g}(t); E_1)$ and $I(\bar{g}(t); E_2)$ per (coding) base. In the ‘varying environment’ experiments, we switch environments between E_1 and E_2 every generation, for the same total of 400 generations. Code for these simulations, results, and subsequent analysis are available at <https://github.com/ngutten/>

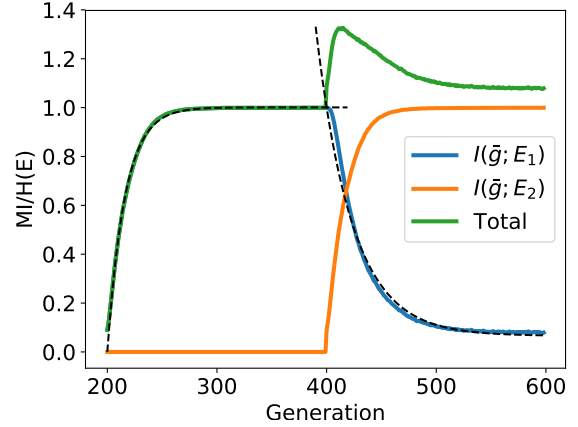


Figure 1: Dynamics of mutual information between population and environment for $\mu = 0.5$ and $\Gamma = 0.2$. Information measures are normalized by the maximum information per base associated with a single environment for that value of Γ . Dashed lines are fits to Eq. 3

evolution_infodynamics.

The first set of results involves the dynamics of the mutual information with respect to time, and are shown in Fig. 1. The observed dynamics of the mutual information under selection are quite close to a saturating function of the form $I(t) = A(1 - \exp(-t/\lambda_+))$. Meanwhile, the decay behavior when a different environment is being selected for appears to closely follow the form $I(t) = A \exp(-t/\lambda_-)$.

This type of saturating behavior is consistent with dynamics in which there is a constant source and a linear decay:

$$\partial_t I_i = \begin{cases} -I_i/\lambda_+ + A/\lambda_+, & \text{Selection on } E_i \\ -I_i/\lambda_-, & \text{Selection on } E_j \end{cases} \quad (3)$$

The fact that λ_- and λ_+ are not the same suggests that the decay is not just from mutation (which should be the same in both cases), but rather includes an effect where selection for one environment can influence the rate at which mutual information with a second environment is lost. It is a bit surprising that the form would be this simple, as why would selection provide information at a constant rate rather than one which depends on how the population is positioned relative to the fitness landscape? It may just be that for our particular fitness function, since each move towards the fitness optimum provides the same selective contrast regardless of how close or far an organism is to the optimum, that these effects are zero. In fact, if we switch to a fitness landscape in which the fitness is exponential in the Hamming distance from the optimum, we see kinetics of the form $A(1 - \exp(-t^2/\lambda_+^2)) + B$ instead (Fig. 2), so this is not universal.

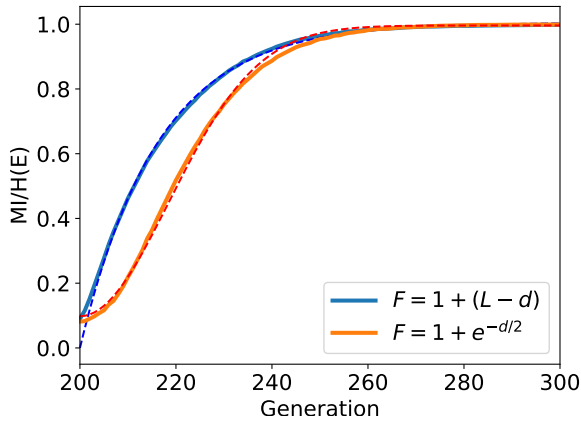


Figure 2: Comparison between the initial increase of mutual information for a linear fitness function versus an exponential fitness function. The blue solid line corresponds to a linear fitness function as given by Eq. 1, and the blue dashed line is a sigmoidal fit. The red solid line corresponds to a fitness $F = 1 + \exp(-d/2)$ where d is the Hamming distance between target and environment, and the red dashed line is a fit to $A(1 - \exp(-t^2/\lambda_+^2)) + B$. Parameters are $\mu = 0.5$ and $\Gamma = 0.2$.

Given that we do observe this form of kinetics over a large range of parameters for our case, we can attempt to use Eq. 3 to relate the kinetics of saturating adaptation in the case of one environment to what would happen in a system which oscillates between two environments with period T , by balancing the information gained during the selection phase against the information lost during the non-selected phase:

$$AT/\lambda_+ - \int_0^T I(t)/\lambda_+ dt - \int_T^{2T} I(t)/\lambda_- dt = 0 \quad (4)$$

For rapidly oscillating environments ($T \rightarrow 0$), this criterion is satisfied when:

$$I = A \frac{1}{1 + \lambda_+/\lambda_-} \quad (5)$$

This is in comparison to an asymptotically slowly-varying environment, in which the information about one environmental random variable $I_1 \rightarrow A$, while the other $I_2 \rightarrow 0$. In this case, since the different environments are independent ($I(E_1; E_2) = 0$), the total mutual information between the population and the set of environments is additive $I(g; E_1, E_2) = I(g; E_1) + I(g; E_2)$. We can now ask, at least in the context of this model, when does a fluctuating environment result in the population having more information about the set of environments in total than if it just adapted to

a single environment? In the context of the criterion given by Eq. 5, we should expect this to happen when the timescales of information gain and decay are equal $\lambda_+ = \lambda_-$ because at that point, each environment contributes half of its total entropy to the total information ($I_1 = I_2 = A/2$), and so the total information the system has about the oscillating environment pair is equal to the total information the system would have about one single static environment.

In the example of Fig. 3, we see that there is a point at which the sum total information between the system and both environments exceeds the entropy of a single environment — the information content is greater than what could be obtained if an environmental switch was not present. Similarly, we expect there to be cases in which switching between environments prevents the entirety of the available entropy of a single environment to be transferred into the system. According to our simplified model, we predict that the cross-over between these cases should occur when the timescales of information gain and decay are equal. As such, we will test the simplified model by empirically measuring those timescales, and then comparing the cross-over in timescales to the point at which the excess information peak disappears.

Measuring timescales

In order to test this relationship between the timescales of information gain and decay, we fit the observed dynamics of mutual informations in the slowly varying environments to Eq. 3 and look at how the timescales λ_+ and λ_- vary with μ (Fig. 3a), Γ (Fig. 3b), and population size (Fig. 3c). Unsurprisingly, as we increase the mutation rate, information is lost from the system more rapidly. However, increasing the mutation rate also increases the rate at which the information held by the system about the environment approaches its asymptotic value. Increased mutation rate may ultimately lead to less information being retained — and so in terms of total information gain by some fixed point in time τ we would expect there to be a local maximum with respect to mutation rate. However, when that total information gain is normalized out, the remaining effect of mutation on the timescale is monotonic.

When we increase the fraction of the non-neutral part genome Γ , we are in effect increasing the amount of information there is to learn about a given environment, and correspondingly this means that we are decreasing the relative information capacity with respect to the genome and the size of an optimal solution for a given environment. As such, the behavior of the onset and decay phases with respect to increasing Γ are markedly different. As increased Γ means there is more information to learn before convergence, the slope of the fraction of the total entropy captured is proportionally decreased and so the timescale of learning is correspondingly slowed (so λ_+ becomes larger with increasing Γ). On the other hand, as the genome is closer to

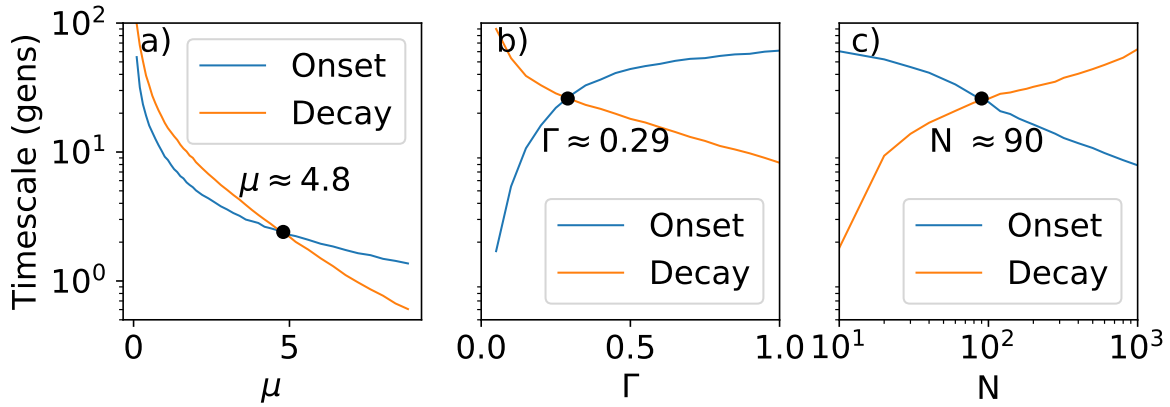


Figure 3: Variation of the time constant of information gain and loss with μ (a), Γ (b), and population size N (c). The baseline parameters are $\Gamma = 0.2$, $\mu = 0.5$, and $N = 200$, with the three panels each corresponding to varying one quantity around that baseline.

capacity, once the environment changes then a larger fraction of bits are in conflict and so experience decay at the rate imposed by selection effects, rather than the rate imposed by mutation effects. In response, λ_- monotonically decreases with increasing Γ .

Population size has a strong impact on timescales as well. Even if an individual organism experiences a mutation in a particular base, there are redundant copies of that information distributed among the population. As such, information about the previous environment decays not just at the rate of an individual mutating, but instead at the rate at which that mutation would proceed towards fixation. For neutral mutations, this timescale is linear in the population size (Eq. 14 of Kimura and Ohta (1969)), whereas for non-neutral mutations it is (to first order) logarithmic in the population size (Eq. 50 of Uecker and Hermisson (2011)). So it makes sense that as we increase the population size, we generally see an increase in the decay timescale λ_- . At the same time, a larger population means that selection has an increased bandwidth for transferring information about the environment into the system — and so the rate of information gain accelerates, and the onset timescale λ_+ becomes shorter.

Since the characteristic time-scale of information gain under a new selection pressure generally differs from the characteristic time-scale of information loss about previous environments, meaning that the population can end up with either an information excess (in that it stores more adaptive information than would be necessary to maximize fitness in the current environment alone) or an information deficit (in that adaptation to the new environment causes the old one to be forgotten more quickly than new information comes in). Holding Γ constant, we see that there is a particular value of the mutation rate at which this cross-over occurs, although higher mutation rates reduce both timescales strongly. In

comparison, holding μ constant, Γ also has a cross-over point but the effect is much stronger.

Varying environments

This now brings us to the case of varying environments. This cross-over point indicates that, for one switching event, the effect can either be to temporarily increase the information content of the system about both environments or temporarily decrease the information content of the system below where it would otherwise be without the switch. If we now have multiple switching events on a timescale comparable to the scale of information gain and decay, we might expect the parameters of the system relative to this cross-over point to determine whether the effect of environmental variations is force extra information into the system on the net, or to force information out of the system on the net. We show the peak total information observed in environments oscillating with period $T = 1$ in Fig. 4 in comparison with the same system at $T = 200$ both for varying μ and for varying Γ . As expected, we see a cross-over between the case in which fluctuations drive excess information into the system and the case in which fluctuations drive information out of the system. While the cross-over points are similar to the point at which the time-scale of information gain and information decay are balanced, they do disagree in detail by a factor of around 1.4. So while the broad idea that balance between information gain and loss informs us about how an evolutionary system would respond to a fluctuating environment, the actual relationship seems to differ in some details.

Conclusions

We have demonstrated a measurement of the dynamics of mutual information between population and environment in the case of a simplified binary sequence evolutionary simu-

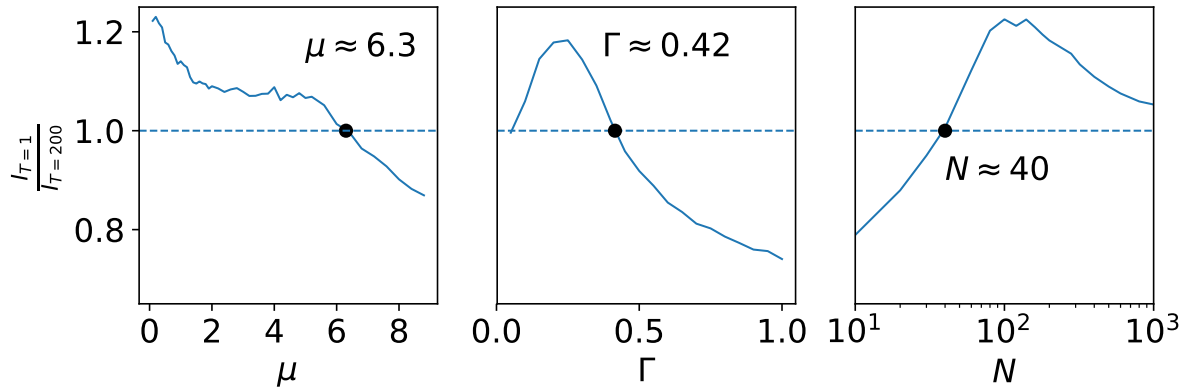


Figure 4: This figure shows the ratio of the peak sum information between a system with changes between a pair of environments every generation, and one in which a single change occurs at $T = 200$. The left panel shows the case in which the mutation rate μ is varied with a constant $\Gamma = 0.2$; the center panel shows the case where Γ is varied with a constant $\mu = 0.5$; and the right panel shows variation in population size, given constant $\Gamma = 0.2$ and $\mu = 0.5$. In the case of μ and Γ , the cross-over points occur at larger values than the corresponding cross-over between timescales in the static environment by about a factor of ≈ 1.4 : $\mu = 6.3$ versus $\mu = 4.8$ for static, and $\Gamma = 0.42$ versus $\Gamma = 0.29$ for static. For the population size effect, the cross-over occurs earlier at around $N = 40$, compared to $N = 90$ for equal timescales.

lation. The dynamics of information in this system exhibit sigmoidal growth and decay curves, consistent with the idea of a constant rate of information injection balanced against proportional information loss. Where that holds, changes in the environment result in dynamics with a characteristic time-scale associated the gain of information about the new environment and a different characteristic time-scale associated with loss of information about the old environment. Depending on mutation rate and the amount of information associated with environments relative to the genome capacity, these time-scales vary and may undergo a cross-over at particular values of the mutation rate and capacity. We expect from the sigmoidal model that this cross-over would correspond to the point at which a rapidly fluctuating environment would asymptotically induce either a net gain or loss of information in the population — in essence, determining whether or not adapting to multiple environments would enhance or inhibit the evolutionary process. While we found such a transition, the location of the transition differed by a significant factor from the point which would be predicted by looking at the time-scales alone, suggesting that there is still some additional consideration for how fluctuations interact with the evolutionary dynamics that the simplified information flow model is missing.

One potential factor is that ability of a population to gain information from the environment depends on the population's structure with respect to the structure of the environment — that is to say, selection does not simply introduce a constant transfer of information from environment to population, but rather causes changes in the population struc-

ture which have consequences for how much information will be able to flow from the environment to the population in subsequent generations. In both (Steiner, 2012) and (Ofria et al., 2016) a connection is made between environmental fluctuations and the evolution of evolvability. Similarly, (Virgo et al., 2017) suggests that even for simple fitness landscapes, lineage effects can lead to significant adaptation of the evolvability of the population. These factors are not captured by the simple rate model of information flow. However, at the same time, an investigation into how the rate of information flow within an evolving system changes over time could be a useful way to probe these evolvability effects in the future.

Acknowledgements

We would like to acknowledge Nathaniel Virgo and Harrison Smith for helpful feedback on the contents of this paper.

References

- Adami, C. (2004). Information theory in molecular biology. *Physics of Life Reviews*, 1(1):3–22.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open problems in artificial life. *Artificial life*, 6(4):363–376.
- Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. (2018). Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

- Chang, C.-H., Hsieh, L.-C., Chen, T.-Y., Chen, H.-D., Luo, L., and Lee, H.-C. (2005). Shannon information in complete genomes. *Journal of bioinformatics and computational biology*, 3(03):587–608.
- Gloor, G. B., Martin, L. C., Wahl, L. M., and Dunn, S. D. (2005). Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions. *Biochemistry*, 44(19):7156–7165.
- Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317.
- Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711–13716.
- Kimura, M. and Ohta, T. (1969). The average number of generations until fixation of a mutant gene in a finite population. *Genetics*, 61(3):763.
- Levins, R. (1967). Theory of fitness in a heterogeneous environment. vi. the adaptive significance of mutation. *Genetics*, 56(1):163.
- Martin, L., Gloor, G. B., Dunn, S., and Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, 21(22):4116–4124.
- Medernach, D. (2017). *Comparative study of effects of fitness landscape changes in open-ended evolutionary simulations and in genetic programming*. PhD thesis, University of Limerick. https://ulir.ul.ie/bitstream/handle/10344/6103/Medernach_2017_comparative.pdf.
- Neumann Jr, F. X. (1997). Organizational structures to match the new information-rich environments: Lessons from the study of chaos. *Public Productivity & Management Review*, pages 86–100.
- Ofria, C., Wisler, M. J., and Canino-Koning, R. (2016). The evolution of evolvability: Changing environments promote rapid adaptation in digital organisms. In *Artificial Life Conference Proceedings 13*, pages 268–275. MIT Press.
- Schneider, T. D. (2000). Evolution of biological information. *Nucleic acids research*, 28(14):2794–2799.
- Steiner, C. F. (2012). Environmental noise, genetic diversity and the evolution of evolvability and robustness in model gene networks. *PLoS one*, 7(12):e52204.
- Uecker, H. and Hermisson, J. (2011). On the fixation process of a beneficial mutation in a variable environment. *Genetics*, 188(4):915–930.
- Virgo, N., Agmon, E., and Fernando, C. (2017). Lineage selection leads to evolvability at large population sizes. In *Artificial Life Conference Proceedings 14*, pages 420–427. MIT Press.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331.
- Xia, Z., Jin, G., Zhu, J., and Zhou, R. (2009). Using a mutual information-based site transition network to map the genetic evolution of influenza a/h3n2 virus. *Bioinformatics*, 25(18):2309–2317.

Modulating Interaction Times in an Artificial Society of Robots

Yara Khaluf¹ and Heiko Hamann²

¹IDLab, Ghent University - imec, Belgium

²Institute of Computer Engineering, University of Lübeck, Germany
yara.khaluf@ugent.be

Abstract

In a collaborative society, sharing information is advantageous for each individual as well as for the whole community. Maximizing the number of agent-to-agent interactions per time becomes an appealing behavior due to fast information spreading that maximizes the overall amount of shared information. However, if malicious agents are part of society, then the risk of interacting with one of them increases with an increasing number of interactions. In this paper, we investigate the roles of interaction rates and times (aka edge life) in artificial societies of simulated robot swarms. We adapt their social networks to form proper trust sub-networks and to contain attackers. Instead of sophisticated algorithms to build and administrate trust networks, we focus on simple control algorithms that locally adapt interaction times by changing only the robots' motion patterns. We successfully validate these algorithms in collective decision-making showing improved time to convergence and energy-efficient motion patterns, besides impeding the spread of undesired opinions.

Introduction

Human societies are built upon social networks that form the infrastructure for sharing information, such as ideas (Leskovec et al., 2009; Orellana-Rodriguez and Keane, 2018), product innovations (Leskovec et al., 2007; Zhong et al., 2018), and political movements (Polletta and Jasper, 2001). The network metaphor suggests a discrete and static property (edge/no edge), however, agent-to-agent interactions have certainly different intensities and durations. Specific characteristics of these agent-to-agent interactions influence global phenomena, such as consensus formation in collective decision-making processes or the spread of an epidemic (Keeling and Rohani, 2011; Heesterbeek et al., 2015).

The interactions of an individual are key to understand information spreading processes in its neighborhood. They can significantly influence its opinions and define its information spreading capability. Local neighborhoods are relevant, as shown for example by Christakis and Fowler (2007) who report that your chances of being obese are probabilistically characterized by your social network. In turn, obesity as a global feature of society can also be analyzed based on the social interactions among individuals.

Furthermore, dynamic individual interactions drive the emergence of communities (Hess et al., 2016). In general, social systems have features that develop on different time scales ranging from micro-seconds to years. Spreading information travels magnitudes faster than the speed of change of the underlying network topology. Hence, in simplified models the topology can be approximated by static networks (Krings et al., 2012). Models representing interaction times may, however, prove to be more powerful to study the spread of information.

Here, we study the dynamics of information spreading in large societies as observed in artificial and natural collective systems. We focus on the timescales of topology dynamics (i.e., community formation) and their impact on global features. As key parameter of our study we choose the average and distribution of agent-to-agent interaction times. As interaction time we define the uninterrupted time spent by two agents in mutual communication range. The distribution of the interaction time is also known as edge life distribution (Yang et al., 2013). Our main inspiration comes from Meier (1962) who reports the importance of decreased interaction times in modern cities Khaluf (2017). We study two complementary agent behaviors that either (a) share a desirable piece of information or (b) trap an undesired piece of information. In natural systems, such as collective decision-making in ants (Pratt et al., 2002), trust is virtually taken for granted and spreading of information maximized. However, ants tweak their social network when they need to fight an epidemic (Stroeymeyt et al., 2018). In artificial systems, such as collective decision-making in swarm robots (Khaluf et al., 2018; Hamann et al., 2014; Valentini et al., 2016; Rausch et al., 2018), sharing of information is to be maximized and in many studies trust is also virtually taken for granted. Only few works on decentralized error detection focus on identifying and excluding damaged robots (Lau et al., 2011; Tarapore et al., 2015). In our study, we assume that not all information is useful but possibly harmful. We extend the current state of the art in collective decision-making of artificial societies by considering how to deal with undesired information. The focus is on how to modulate agent-to-agent interaction times

to either boost or impede spread of information.

We investigate our hypothesis using an artificial collective system, a simulated robot swarm (Hamann, 2018b), that has a dynamic topology. By using a collective robot system we restrict ourselves to local links of directly interacting neighbors. This spatial restriction allows us to draw relations to models of human networks for different spatial densities (e.g., urban vs. rural). Furthermore, simulated robot swarms serve here as systems that are easy to track and to engineer. We tune the interaction dynamics of the swarm (e.g., interaction times) to achieve desired collective behaviors, such as a decision consensus in collective decision-making. We examine the role of interaction times in a symmetry-breaking problem, that is, agents need to select between two options of equal quality. Two parameters are important to optimize: (1) degree of coherence, the percentage of robots that finally agree on the same option, in the best case consensus (100% agreement) and (2) time to convergence, the time to achieve a stable degree of coherence. The time to converge is particularly important in the symmetry-breaking type of decision-making, because both options are of the same quality, hence, further gathering of information is wasted time and cannot increase the accuracy of the decision. As commonly assumed in collective decision-making scenarios, we have an instantaneous exchange of information (opinions) between neighboring agents. We analyze the role of the interaction time under two modes (1) offline tuning of interaction times by adapting the population density externally (Khaluf et al., 2017), and (2) active tuning of interaction times by a set of proposed algorithms that exploit robot motion patterns to modulate the distribution of the agent-to-agent interaction times. The physical interaction time was highlighted as a key parameter in the emergence of collective motion behaviors by Stark et al. (2008).

Approach to modulate interaction times

We use a homogeneous swarm of N simulated robots.¹ Robots wander randomly in their arena and while moving they interact with their local neighborhood that is defined based on their communication range. All robots within communication range are neighbors and interactions are mutual. Therefore, the robot control algorithm influences the distribution of agent-to-agent interaction times but also the spatial distribution of robot density. (number of robots per area).

We consider a symmetry-breaking problem (Hamann et al., 2012). A binary collective decision-making problem with options A and B of the same quality. The robot swarm is asked to achieve a consensus on either one of them. This is a well-studied problem known from different fields, such as physics, biology, opinion dynamics, and others.

Study setups

Our study consists of the following experiment setups:

¹Footbot, http://www.swarmanoid.org/swarmanoid_hardware.php

A - interaction time as a function of population density:

In preliminary experiments, we measure and analyze the dependency of agent-to-agent interaction time changes on swarm density. We explore possibilities of tweaking interaction times offline by choosing proper swarm densities.

B - boosting spread of desired information in a decision-making system with majority rule:

We apply the majority rule in a symmetry-breaking scenario, the spreading information is the robots' opinions, that are in this case desired to spread widely. We define two sets of experiments here. First, we use offline tuning of interaction times (i.e., swarm density). Second, we validate our control algorithm to actively tune interaction times in order to achieve a system with an enhanced well-mixed property (i.e., closer to the ideal of a well-mixed system where the chance of any agent to interact with any other agent is equal). We modify the robot motion to improve the diffusion of robots. The result are decreased agent-to-agent interaction times and an increased chance of encountering more neighbors, that increase a robot's opinion sample. Well-mixed populations are advantageous in collective decision-making. However, once consensus is reached the robots may not need to continue their costly motion pattern. We propose an energy-aware algorithm that creates a global self-awareness of the degree of coherence in the swarm. Each robot samples opinions of its neighborhood and measure the degree of coherence. In the case of high coherence over a long-enough period, robots slow down until they stop and preserve their current neighborhood as it is not necessary for the decision-making process to move further.

C - impeding spread of undesired information in a decision-making system with majority rule:

We consider a heterogeneous swarm with two sub-populations. One sub-population uses the majority rule as above. The other sub-population is assumed to be malicious. They contradict the majority in their neighborhood and adopt the opposite opinion. In the literature they are called contrarians (Gambaro and Crokidakis, 2017; Khalil and Toral, 2019; Hamann, 2018a). First, we show that contrarians reduce coherence. Second, we show how our proposed control algorithm can tune interaction times with contrarians and contain them. Their influence is then limited and the degree of coherence is increased. We highlight that extending agent-to-agent interaction times can be of a significant benefit in this setup. This is in contrast to standard setups in collective decision-making where short interaction times are preferred because they improve the mixing and spread of opinions.

Motion control algorithms

In the following we describe the different robot control algorithms that we use to modify the motion behaviors of robots that, in turn, changes their interaction times.

Algorithm 1: Standard robot control algorithm for collective decision-making using the majority rule.

```

1 select initial opinion (uniform ( $A, B$ ));
2 while not(end of experiment) do
3   broadcast own opinion;
4   if  $Pr_d < Pr_\epsilon$  then
5     switch opinion spontaneously;
6   else
7     collect neighbors' opinions;
8     if # neighbors is even then
9       keep opinion;
10    else
11      adapt majority opinion;
12  diffuse with obstacle avoidance;

```

Robots move in a gas-like diffusion as they move away from areas of high robot density to spaces of low density. This motion mechanism serves as both, obstacle avoidance and exploration. In each simulation step, a robot collects vectors from its proximity sensors² consisting of relative distance and relative angle. A summed vector is used as indicator towards areas of low density (potential field method). If no other robots are detected, robots move in a straight line.

Following the standard control algorithm (Alg. 1), collective decision-making is implemented by the majority rule. We keep this decision algorithm unchanged for all experiments—since we focus exclusively on motion patterns and their interaction times. A robot collects the opinions of its neighborhood and switches to (or keeps) the opinion of the majority. We add a probabilistic element by spontaneous opinion switching with probability $Pr_\epsilon = 0.1$ (otherwise majority rule with probability $1 - Pr_\epsilon = 0.9$). Spontaneous switching helps to preserve the system's possibility to explore even when consensus is achieved. The decision-making process is continuous as it is active while the robots move around randomly, see Alg. 1.

We propose an algorithm to provoke a more mixed system by modifying the robot motion behavior. The idea is to increase the average number of neighbors and to reduce interaction times. Note that any opinion exchange requires a minimal time to transmit information. We assume this minimal time to be one discrete time step in our experiments (i.e., instantaneous opinion exchange). We call this variant the 'well-mixed algorithm,' see Alg. 2. A robot attempts to move away from its neighbors, once there can be no relevant information exchange anymore (i.e., homogeneous neighborhood). A robot computes a repulsive velocity vector using relative distances and angles to its neighbors. A robot is attracted by neighbors with opposing opinion. Hence, all robots try

²the simulated Footbot uses 24 proximity sensors

Algorithm 2: Well-mixed robot control algorithm for faster opinion sampling in collective decision-making.

```

1 select initial opinion (uniform ( $A, B$ ));
2 while not(end of experiment) do
3   broadcast own opinion;
4   if  $Pr_d < Pr_\epsilon$  then
5     switch opinion spontaneously;
6   else
7     collect neighbors' opinions;
8     if # neighbors is even then
9       keep opinion;
10    else
11      adapt majority opinion;
12  if  $N_{diff} > 1$  then
13    for  $i=1, i \leq N_{diff}$  do
14       $x = \text{sensor}(i).value * \cos(\text{sensor}(i).angle)$ ;
15       $y = \text{sensor}(i).value * \sin(\text{sensor}(i).angle)$ ;
16       $v.x = v.x + x$ ;
17       $v.y = v.y + y$ ;
18       $\alpha_{attract} = \text{atan2}(v.y, v.x)$ ;
19      compute linear speed from  $\alpha_{attract}$ ;
20  else if all neighbors with same opinion then
21    for  $i=1, i \leq N_{all}$  do
22       $x = \text{sensor}(i).value * \cos(\text{sensor}(i).angle)$ ;
23       $y = \text{sensor}(i).value * \sin(\text{sensor}(i).angle)$ ;
24       $v.x = v.x + x$ ;
25       $v.y = v.y + y$ ;
26       $\alpha_{repulse} = \text{atan2}(v.y, v.x)$ ;
27      compute linear speed from  $\alpha_{repulse}$ ;
28  else
29    diffuse with obstacle avoidance;

```

to maximize interaction between robots of different opinion. On the one hand, robots maximize their dissemination effect by approaching and convincing minorities of robots with opposing opinion. On the other hand, robots maximize chances to be convinced of the opposite by their exposure to robots of opposing opinion. The algorithm pushes robots from one neighborhood to another avoiding to waste time in homogeneous sub-populations. As mentioned above, once consensus is reached robots don't need to maintain a certain motion pattern for opinion mixture. In fact, that is a waste of energy. We propose Alg. 3 as the 'energy-aware' algorithm. Robots exchange opinions in their local neighborhood over time and use their collected sample to decide when they can stop their specific motion pattern.

Finally, we propose a robot control algorithm, that tries to contain contrarians in collective decision-making (see Alg. 4) called 'impeding algorithm'. We have two sub-populations: (1) the standard individuals executing the majority rule (as

Algorithm 3: Robot control algorithm for adaptive interaction time based on individual awareness of coherence degree in collective decision-making.

```

1 while not(end of experiment) do
2   Apply decision making using Alg. 1;
3   for  $i=1, i \leq N_{neighbors}$  do
4     if  $opinion(i)=A$  then
5       |  $\#Neighbors\_A=\#Neighbors\_A+1$ ;
6     else if  $opinion(i)=B$  then
7       |  $\#Neighbors\_B=\#Neighbors\_B+1$ ;
8    $Neighbors\_Composition=|\#Neighbors\_A-$ 
    $\#Neighbors\_B|/size\_local\_neighborhood$ ;
9   if  $simulation\_timer \% \delta\_t$  then
10    compute mean  $\mu_{neighborhood}$  of
    $Neighbors\_Composition$ ;
11    if  $\mu_{neighborhood} = 1$  then
12      | start slowing down by  $\delta\_s$  at each time step;
13    reset  $Neighbors\_Composition$ ;
14  diffuse with obstacle avoidance at the updated speed;

```

described in Alg. 1), (2) the contrarians that always adopt the minority opinion of the neighborhood. Our goal is to limit the effect of contrarians by containing them. Following the impeding algorithm, we modify the motion of standard robots who are neighbors of contrarians. Neighbors of a contrarian (who may already be affected by its opinion) try to increase their interaction time with it to prevent contact with other robots. When standard robots find a contrarian, they encircle it in rough analogy to immune systems (see footnote 3).

Results

We validate our algorithms by running physics-based simulations using the ARGoS simulator (Pinciroli et al., 2012). We set an arena of $6 \times 8 m^2$ and test swarm sizes $N \in \{50, 100, 200, 300, 400, 500\}$ in a symmetry-breaking collective decision task. Each experiment run is repeated 30 times independently with 2000 time steps each. Tab. 1 gives the parameter values of both the robot and control algorithms.

A - interaction times as function of swarm density

We study the relation between agent-to-agent interaction times and swarm densities (robots per area). Knowledge of this relation can be used to modulate interaction times offline by setting appropriate swarm densities. Due to limited space, increased swarm densities may cause cascades of collision avoidance actions, where robots take longer to leave their neighborhoods. In Fig. 1a, we show the average interaction times over swarm size. We keep the area constant,

Algorithm 4: Impeding robot control algorithm to contain contrarians in collective decision-making.

```

1 while not(end of experiment) do
2   broadcast own opinion;
3   if  $Pr_d < Pr_\epsilon$  then
4     | switch opinion spontaneously;
5   else
6     collect neighbors' opinions;
7     if  $\#neighbors$  is even then
8       | keep opinion;
9     else if  $(\#neighbors \text{ odd}) \& (\text{not contrarian})$ 
10      then
11       | adapt majority opinion;
12     else if  $(\#neighbors \text{ odd}) \& (\text{contrarian})$  then
13       | adapt minority opinion;
14   if  $\#contrarian\ neighbors > 0$  then
15     for  $i=1, i \leq N_{contrarian}$  do
16       |  $x = sensor(i).value * \cos(sensor(i).angle)$ ;
17       |  $y = sensor(i).value * \sin(sensor(i).angle)$ ;
18       |  $v.x = v.x + x$ ;
19       |  $v.y = v.y + y$ ;
20      $\alpha_{attract.to.contrarian} = \text{atan2}(v.y, v.x)$ ;
21     compute linear speed from  $\alpha_{attract.to.contrarian}$ ;
22   else
23     | diffuse with obstacle avoidance;

```

Table 1: Parameters used in the simulation.

Parameter	Value
<i>Robot parameters</i>	
Type	Footbot
Proximity sensor range r_{prox}	0.1 m
Range-and-bearing sensor range r_{rab}	1.0 m
Maximum moving speed	$5 \frac{m}{s}$
<i>Algorithms parameters</i>	
<i>Alg. 1</i>	
Pr_ϵ	0.1
<i>Alg. 3</i>	
δ_t	30 s
δ_s	$0.01 \frac{m}{s}$

hence, increased swarm size means increased swarm density. Average interaction times and their standard deviation increase super-linearly with increasing swarm size. In Fig. 1b, we show average interaction times as a function of average distance between robots. The values are averaged over all robots and 30 independent runs each. In full correspondence to Fig. 1a, we observe a decrease of interaction times with increased robot-to-robot distance.

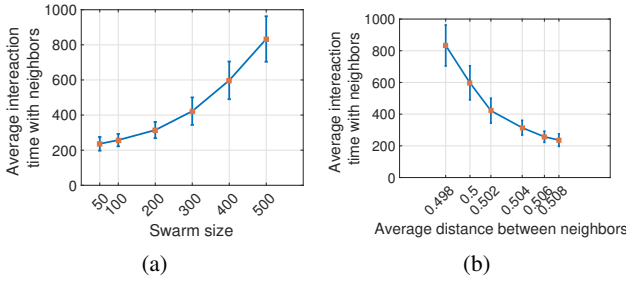


Figure 1: The average interaction times as a function of (a) the population size/density. (b) The average distance between robots for different densities. Error bars give the standard deviation over independent 30 runs.

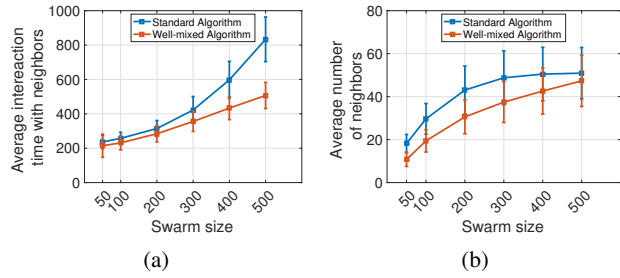


Figure 2: Comparison well-mixed to standard algorithm. The average of (a) interaction times and (b) neighborhood size is computed as a function of the swarm size/density. Error bars give the standard deviation over 30 runs.

B - boosting spread of desired information

In this set of experiments, robots use the majority rule as decision mechanism (Alg. 1) with a probability of $Pr_{\epsilon} = 0.1$ for spontaneous option switching. This implements an exploration behavior to keep the swarm adaptive. First, we exploit only the robots' (gas-like) diffusion mechanism that we used above to generate the interaction time (Fig. 1). Second, we validate our well-mixed algorithm.

In Fig. 2a we compare the interaction times for a swarm controlled by the well-mixed algorithm to a swarm controlled by the standard algorithm. Especially for large swarms (e.g., $N = 400$ and $N = 500$), we notice a significant reduction of interaction times using the well-mixed algorithm. In Fig. 2b we compare the resulting average neighborhood sizes. There is a trend to reduced neighborhood sizes using the well-mixed algorithm, specifically for medium swarm sizes (e.g., $N = 200$ and $N = 300$). For large neighborhood sizes (e.g., $N = 400$ and $N = 500$) both algorithms approach saturated neighborhood sizes of ≈ 48 robots. This is the limit due to densely packed clusters of robots.

In Fig. 3a, we compare the well-mixed to the standard algorithm in terms of time to convergence. We define time to convergence using a threshold as the first passage time to (absorbing) states of more than 90% majority. In our comparison, we notice that with increasing swarm size, robots spend

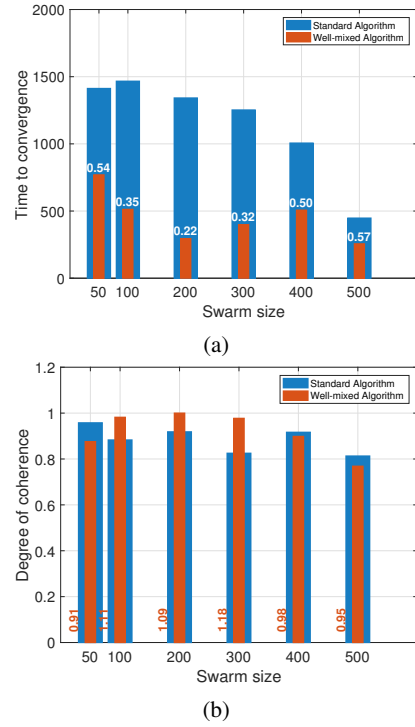


Figure 3: (a) The time the swarm takes to converge to an equilibrium, (b) the degree of coherence reached by the swarm. Both computed over the different population densities, and under both algorithms (i) standard, and (ii) well-mixed.

more time to avoid obstacles and they cover less distance. Due to the high density, swarm-wide connectivity is highly probable resulting in large connected components. The time to propagate opinions is reduced compared to sparse densities when robots need to move around in 'empty space' to connect to new neighborhoods. The result is a decreasing time to convergence with increasing swarm size. Our proposed well-mixed algorithm reduces the time to convergence significantly compared to the standard algorithm. The improvement (reduction of time to converge) reaches a maximum of 78% in the case of medium density ($N = 200$ robots), and a minimum of 43% in the case of dense swarms ($N = 500$ robots). Best improvements are achieved for medium swarm sizes ($100 \leq N \leq 300$). With increasing density (e.g., $N = 500$) robots cannot move anymore and hence no further improvement is possible.

We use the degree of coherence as a second method to measure performance. In Fig. 3b, we compare the degree of coherence for both algorithms (i.e., the standard and the well-mixed). For high densities ($N \geq 400$) coherence doesn't differ between the two algorithms because the swarms are fully connected. Similarly for sparse swarms ($N = 50$), robots mix well following the standard algorithm due to sparsely populated space. For low density ($N = 50$) the well-mixed algorithm is even detrimental to coherence but it converges

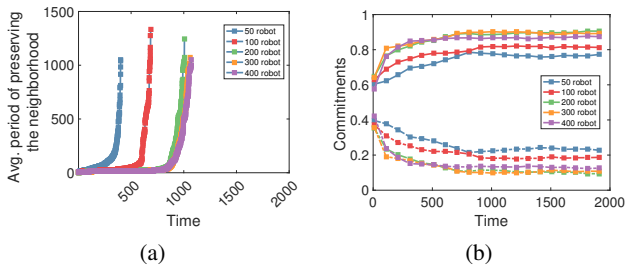


Figure 4: Results for the energy-efficient Alg. 3 for different swarm sizes, (a) average lifetime of a neighborhood (sub-population of the swarm), (b) commitment to the two opinions over time.

faster. This is because, in sparse swarms the well-mixed algorithm maximizes robot diffusion, that decreases robot interactions compared to the standard algorithm. The well-mixed algorithm enables the degree of coherence in medium densities ($100 \leq N \leq 300$) to reach one asymptotically—i.e., full consensus. The improvement is between 9% and 18% compared to the standard approach.

Next, we study situations towards the end of runs when high coherence is achieved and it gets irrelevant to maintain a good mixture of opinions. When robots get aware of a high coherence by local measurements, they reduce their motion and hence increase their interaction time with their local neighborhood. We use Alg. 3 that implements this energy-efficient mechanism for collective decision-making. In Fig. 4a, we show the average time of how long a certain neighborhood configuration (a sub-population of the swarm) is preserved during the simulation. At the beginning of each simulation the neighborhood changes often (i.e., short lifetimes of neighborhood configurations and short interaction times). These times increase gradually over time as the system achieves higher degrees of coherence. Robots slow down their nominal velocity by $\delta_s = 0.01$ at each time step when observing full consensus in their neighborhood for a time window of 300 time steps. As a result neighborhood configurations are preserved for longer times until a large majority of the swarm stops moving and builds stable neighborhoods, see Fig. 4a. The degree of coherence reached using Alg. 3 is high for all swarm sizes as seen in Fig. 4b. We also notice that neighborhoods freeze across almost all sizes of populations after the swarm has reached a high degree of coherence, except for swarm size $N = 50$. For this sparse swarm we would require larger time windows (i.e., > 300 time steps) to sample enough neighbors and have a more accurate decision about the degree of coherence reached at the system level.

C - impeding spread of undesired information

Next, we study collective decision-making with sub-populations of contrarian robots. As mentioned above, contrarians observe the current majority in their neighborhood and then switch to the opinion of the minority. In general,

in collective systems we assume that sharing information is necessary and useful or at least not harmful. This implies that maximizing the number of encountered peers is a key objective. Minimizing the agent-to-agent interaction time or limiting it to the duration needed to exchange the information becomes the desired design feature. Nevertheless, in this setup we consider contrarians, whose influence may harm the system by preventing it from achieving a consensus. For such setup, spreading the contrarians' opinions need to be limited. In a first set of experiments, we study symmetry-breaking when a sub-population of contrarians with percentages of 5%, 10%, and 30% is introduced.³ All robots use the gas-like diffusion motion to wander in the arena. The majority of robots (95%, 90%, and 70%) runs Algo. 1 based on the majority rule while contrarians follow their 'minority rule.' Fig. 5a-e shows, respectively, how the swarm loses its capability of making a decision caused by the contrarian sub-population for percentages of 10% and 30%.

In a second set of experiments, we tune the interaction times of the contrarians' neighbors in order to contain those and hence limit their influence to the system, see Alg. 4. We ignore aspects of how to detect and identify contrarians as this is out of scope of our study. Methods of fault detection could be applied (Lau et al., 2011; Tarapore et al., 2015). Here, we assume that robots detect contrarians immediately. Following our impeding algorithm, when a robot encounters a contrarian, it slows down to increase its interaction time with that contrarian. For a neighborhood with more than one contrarian, the robot computes a summed vector as an indicator towards containing all contrarians in the neighborhood. As soon as enough neighbours are available, they build up a cluster around that contrarian and isolate it from the rest of the swarm. This behavior triggered in the neighborhood of the contrarian restricts its influence by keeping it in the same neighborhood for long. In Fig. 5a, we show a swarm with a small sub-population of contrarians (5%) that have no considerable impact on the capability of the collective decision-making system to break the symmetry. This ability is lost when increasing the size of the contrarians sub-population to 10% and 30% as seen in Figs. 5b and d. Figs. 5c and e show the system's improved capability to make decisions when applying the containing algorithm. Clearly the swarm performs better for a contrarian population of 10%. The improvement is minimal, however, for a contrarian population of 30%. This is because a large portion of the swarm is busy forming clusters around contrarians and too few robots are left for the actual decision-making task.

Fig. 6 shows the distributions of interaction times for the standard and impeding algorithm. For high density ($N = 400$), the interaction times are similar but for all smaller tested swarm sizes ($N < 400$) the impeding algorithm increases interaction times significantly. We notice a wide interval of

³video online: <https://youtu.be/zjklEseERAK>

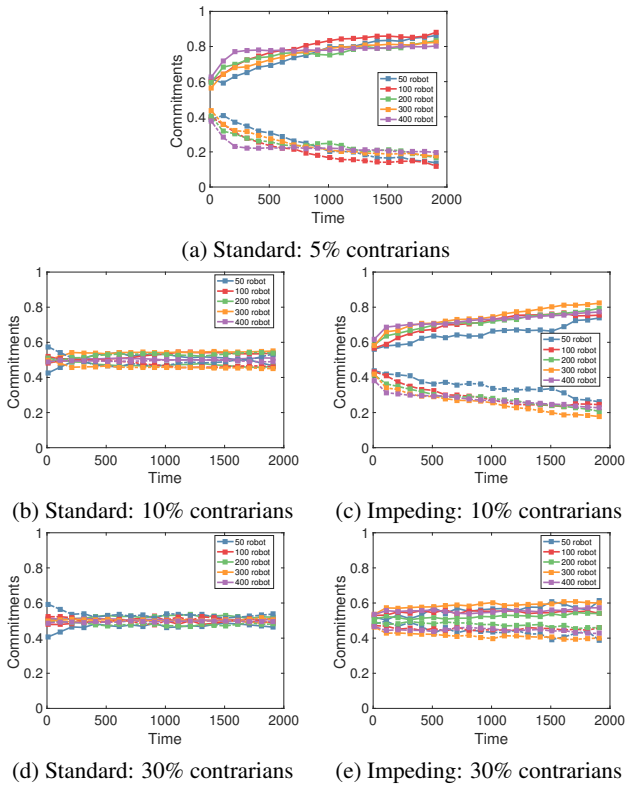


Figure 5: Commitment to the two options over time in the collective decision-making system using majority-rule (a, b, d) with the standard algorithm, (c, e) with the impeding algorithm.

interaction times using the impeding algorithm. This is a clear indicator of the emergence of sub-communities with long interaction times due to the impeding behavior.

Additionally, we compare the standard with the impeding algorithm using the histograms of interaction times in Fig. 7, for different swarm sizes. The histograms clearly indicate a bimodal distribution for the impeding algorithm (red) corresponding to the two sub-populations of contrarian containers and freely moving standard robots. For lower swarm densities ($N \leq 200$), the left peak of the bimodal distribution, that represents the median of the interaction times among the standard robots, overlaps with the peak of the standard algorithm (blue). The right peak is generated by containing and contrarian robots and indicates higher interaction times. Interestingly, for increasing swarm density, the left peak moves to the left relative to the blue peak indicating shorter interaction times than for the standard algorithm (while the median of the complete bimodal distribution of 595.75 is similar to the median of the standard algorithm, 581.97). This is the impeding algorithm’s effect of increasing free space for motion of standard robots. Containing robots densely cluster with contrarians consuming only a small area. This allows the standard robots (left peak) to move quickly and to experience shorter interaction times.

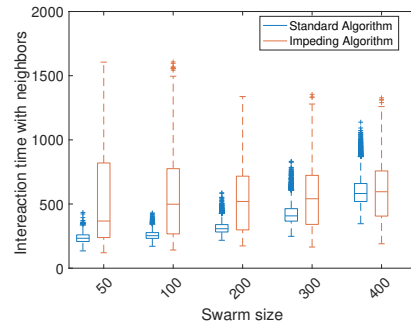


Figure 6: Robot interaction times over swarm size for the standard and the impeding algorithm. Notice the widely spread interaction times for the impeding algorithm.

Conclusions

With focus on interaction times, we have shown for the example of collective decision-making that time to convergence can be improved by carefully designing the motion patterns of robots. The mixture of robots with different opinions can be increased by minimizing interaction times. Following the city metaphor, we create a hyperactive metropolis. In the case of malicious robots (here contrarians), we do not require sophisticated methods of forming and administrating trust networks. Instead, we have shown that significantly increased interaction times with these robots—by exploiting a simple containing strategy designed based on robot motion patterns—can isolate them and reduce their undesired influence. Following the city metaphor, we force contrarians into unhurried spots (e.g., villages) reducing their ability to spread information. With the proposed energy-aware approach, we then regulate interaction times online depending on the level of coherence achieved by the society. Following the city metaphor, this may correspond to a weekly rest day when city life is reduced to a minimum, with the only difference that here the robots reach global awareness of when to schedule that. In future work we plan to test these algorithms on real robots.

References

Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379.

Gambaro, J. P. and Crokidakis, N. (2017). The influence of contrarians in the dynamics of opinion formation. *Physica A: Statistical Mechanics and its Applications*, 486:465–472.

Hamann, H. (2018a). Opinion dynamics with mobile agents: Contrarian effects by spatial correlations. *Frontiers in Robotics and AI*, 5:63.

Hamann, H. (2018b). *Swarm Robotics: A Formal Approach*. Springer.

Hamann, H., Schmickl, T., Wörn, H., and Crailsheim, K. (2012). Analysis of emergent symmetry breaking in collective decision making. *Neural Computing & Applications*, 21(2):207–218.

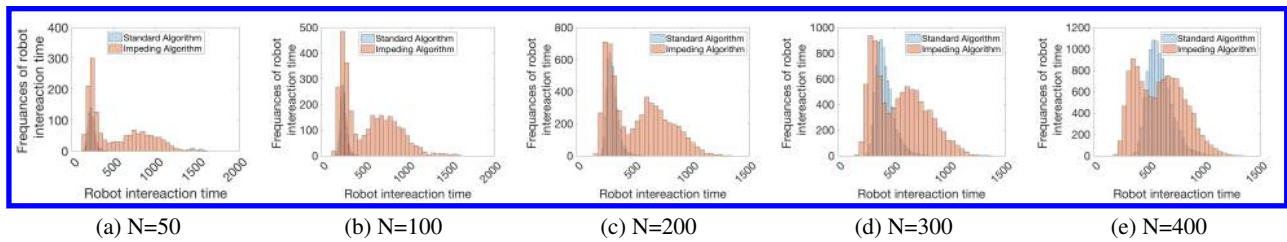


Figure 7: Distribution of interaction times over different swarm sizes, for the standard algorithm and the impeding algorithm, for the latter: bimodal distributions associated with the two sub-populations of contrarian containers and standard robots.

Hamann, H., Valentini, G., Khaluf, Y., and Dorigo, M. (2014). Derivation of a micro-macro link for collective decision-making systems. In *International Conference on Parallel Problem Solving from Nature*, pages 181–190. Springer.

Heesterbeek, H., Anderson, R. M., Andreasen, V., Bansal, S., De Angelis, D., Dye, C., Eames, K. T., Edmunds, W. J., Frost, S. D., Funk, S., et al. (2015). Modeling infectious disease dynamics in the complex landscape of global health. *Science*, 347(6227).

Hess, A., Hummel, K. A., Gansterer, W. N., and Haring, G. (2016). Data-driven human mobility modeling: a survey and engineering guidance for mobile networking. *ACM Computing Surveys (CSUR)*, 48(3):38.

Keeling, M. J. and Rohani, P. (2011). *Modeling infectious diseases in humans and animals*. Princeton University Press.

Khalil, N. and Toral, R. (2019). The noisy voter model under the influence of contrarians. *Physica A: Statistical Mechanics and its Applications*, 515:81–92.

Khaluf, Y. (2017). Can the mechanisms of self-organization help our cities in following the scaling laws of biology? In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 41–44. The Steering Committee of The World Congress in Computer Science, Computer .

Khaluf, Y., Pinciroli, C., Valentini, G., and Hamann, H. (2017). The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability. *Swarm Intelligence*, 11(2):155–179.

Khaluf, Y., Rausch, I., and Simoens, P. (2018). The impact of interaction models on the coherence of collective decision-making: A case study with simulated locusts. In *International Conference on Swarm Intelligence*, pages 252–263. Springer.

Krings, G., Karsai, M., Bernhardsson, S., Blondel, V. D., and Saramäki, J. (2012). Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Science*, 1(1):4.

Lau, H., Bate, I., Cairns, P., and Timmis, J. (2011). Adaptive data-driven error detection in swarm robotics with statistical classifiers. *Robotics and Autonomous Systems*, 59(12):1021–1035.

Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Tran. on the Web*, 1(1):5.

Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM.

Meier, R. L. (1962). *A Communications Theory of Urban Growth*. MIT Press.

Orellana-Rodriguez, C. and Keane, M. T. (2018). Attention to news and its dissemination on twitter: A survey. *Computer Science Review*, 29:74–94.

Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.

Polletta, F. and Jasper, J. M. (2001). Collective identity and social movements. *Annual review of Sociology*, 27(1):283–305.

Pratt, S. C., Mallon, E. B., Sumpter, D. J., and Franks, N. R. (2002). Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant *Leptothorax alpehensis*. *Behavioral Ecology and Sociobiology*, 52(2):117–127.

Rausch, I., Khaluf, Y., and Simoens, P. (2018). Applying scale-invariant dynamics to improve consensus achievement of agents in motion. In *Int. Symp. on Distributed Computing and Artificial Intelligence*, pages 344–348. Springer.

Stark, H.-U., Tessone, C. J., and Schweitzer, F. (2008). Decelerating microdynamics can accelerate macrodynamics in the voter model. *Physical review letters*, 101(1):018701.

Stroeymeyt, N., Grasse, A. V., Crespi, A., Mersch, D. P., Cremer, S., and Keller, L. (2018). Social network plasticity decreases disease transmission in a eusocial insect. *Science*, 362(6417):941–945.

Tarapore, D., Lima, P. U., Carneiro, J., and Christensen, A. L. (2015). To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspiration & biomimetics*, 10(1):016014.

Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016). Collective decision with 100 Kilobots: Speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, 30(3):553–580.

Yang, Y., Chawla, N. V., Basu, P., Prabhala, B., and La Porta, T. (2013). Link prediction in human mobility networks. In *2013 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, pages 380–387. IEEE.

Zhong, X., Zhao, J., Yang, L.-X., Yang, X., Wu, Y., and Tang, Y. Y. (2018). A dynamic discount pricing strategy for viral marketing. *PloS one*, 13(12):e0208738.

Acclivation of Virtual Fitness Landscapes

Ben Kovitz¹, David Bender¹, and Marcela Poffald

¹Fluid Analogies Research Group, Indiana University, Bloomington, IN 47408
bkovitz@indiana.edu

Abstract

Any part of a genome, considered separately from the rest of the genome, evolves against a “virtual fitness landscape” that results when the rest of the genome is held constant. We show how analyzing a genome in this way can explain one form of progressively increasing evolvability.

When one part of a genome is a vector of numbers (“knobs”) and the rest is a graph that determines the mapping from knobs to phenotype, the graph will respond to selective pressure to “acclivate” the virtual fitness function faced by the knobs—that is, to make it more hill-shaped. For as long as the knobs’ virtual fitness function provides opportunity for distorting it to make knob-turning mutations improve fitness, the graph experiences pressure to evolve those distortions as a side-effect of responding to its own virtual fitness function.

As the knobs’ virtual fitness function grows more hill-shaped, the knobs track upward paths more easily and hence so does the genotype as a whole. A synergy develops between incremental exploration of phenotypes by knob-mutations and discontinuous exploration by graph-mutations. A favorable condition for this is a global fitness function that frequently varies, changing constants but leaving structural invariants unchanged. The graph then accumulates a memory of the invariants as revealed across many previous epochs, held in the form of bias limiting and directing future evolution.

Introduction

In previous work (Kovitz, 2015), we found that cascading designs—organisms consisting of graphs that direct cascades of interactions among many parts—are well suited to evolve increasing evolvability, because a single mutation is likely to produce a coordinated change throughout the phenotype, preserving relationships among the parts of the phenotype that might be essential for survival while altering constants that incrementally improve fitness. The classic example of a cascading design is a metabolic network: a variety of enzymes, each catalyzing reactions that create, consume, speed, or slow other enzymes. Others include neural networks, genetic regulatory networks, and even software systems where cascades of activity are propagated by function-calls or message-passing.

In the present paper, we investigate a synergy between the “knobs” of a cascading design—elements subject to incremental, quantitative mutation—and the “graph” or “topology” of a cascading design—the structure of interactions that is subject to discontinuous, sometimes radical mutations. We find that under certain conditions, the graph faces selective pressure to map a rugged fitness landscape to a more hill-shaped virtual landscape for the knobs. The map often excludes the worst regions of the landscape from its range. The result is a mechanism by which evolvability can evolve (Colegrave and Collins, 2008).

Virtual Fitness Functions

Any part of a genome is selected against a *virtual fitness function* resulting from the interaction between the rest of the genome and the fitness function faced by the genome as a whole. If the whole-genome fitness function reflects the influence of the environment on the genome, then the virtual fitness function represents the same for a part of the genome, whose environment includes the rest of the genome.

Let a set of genotypes G have a mapping $m_G : G \rightarrow \Phi$ to a set of phenotypes Φ , and let $w_\Phi : \Phi \rightarrow \mathbb{R}$ be the fitness function for the phenotypes. Then $w_G : G \rightarrow \mathbb{R}$, the fitness function for the whole genome, is the composition of these functions, $w_G(g) = w_\Phi(m_G(g))$.

If we divide the genome into two parts G_1 and G_2 , then each genotype $g \in G$ consists of a $g_1 \in G_1$ and a $g_2 \in G_2$, in which each g_2 defines a partial-genotype–phenotype mapping $m_{g_2} : G_1 \rightarrow \Phi$. That is, if we hold part of the genome constant, say by fixing g_2 , this defines a mapping from all possible values of the rest of the genome, g_1 , to corresponding phenotypes. If we reverse g_1 and g_2 , then of course we get the opposite partial-genotype–phenotype mapping, $m_{g_1} : G_2 \rightarrow \Phi$.

These mappings, in turn, define virtual fitness functions $v_{g_2}(g_1) = w_\Phi(m_{g_2}(g_1))$ and $v_{g_1}(g_2) = w_\Phi(m_{g_1}(g_2))$. As mutations and crossovers can alter either or both of g_1 and g_2 , the partial genomes G_1 and G_2 coevolve cooperatively, each selected by the fitness functions v_{g_1} and v_{g_2} , which vary among all the individuals and vary each generation.

Let *evolvability* be defined in some reasonable way (there are many), so that greater evolvability implies some advantage in navigating a fitness landscape upward faster or further over succeeding generations. Let $g_a, g_b \in G$ be two individuals in the same population and the same generation, g_a having parts g_{a1} and g_{a2} , and g_b having parts g_{b1} and g_{b2} . Assuming no other advantages favoring either g_{a1} or g_{b1} , if g_{a1} presents its mate g_{a2} with a virtual fitness function $v_{g_{a1}}$ that g_{a2} finds more evolvable than $v_{g_{b1}}$ is for g_{b2} , then the descendants of g_a will evolve faster or further than the descendants of g_b (according to how evolvability is defined).

Therefore each partial genome responds to any available selective pressure to create a virtual fitness landscape for the other partial genome that gives the latter greater evolvability. To illustrate with an unrealistically simple example, if the eye and the arm are governed by separate sets of genes, and some arm shapes make it easier for the eye to evolve—say, by providing cues that the eye can track to for hand-eye coordination—then there is selective pressure favoring alleles for those arm shapes. All other factors being equal, evolution favors arms that make eyes easier to evolve. This selective pressure happens indirectly; in any single generation, greater fitness wins. But over successive generations, descendants of organisms with greater evolvability will tend to have greater fitness than organisms with lesser evolvability.

The above considerations make no difference for homogeneous genomes, where every part of each genotype undergoes mutation and crossover the same as every other part and exerts the same effect on the phenotype or on the total fitness as every other part. However, if G_1 and G_2 vary according to different operators and/or affect the phenotype or total fitness differently, there is potential for each part to seek values that make the other part more evolvable, resulting in a period of progressively increasing evolvability for the organism as a whole.

In the rest of this paper, we examine a simple and natural way for this synergy to occur: when g_1 consists of a vector of real numbers (“knobs”) and g_2 consists of a network that provides connections through which the numbers from g_1 interact.

Acclivation

As is well known, a genome consisting of a vector of numbers, where mutations alter the numbers by small amounts, evolves most easily against a hill-shaped fitness function. In a hill-shaped fitness function, local increases in fitness correlate with movement toward the peak of the whole fitness landscape (Kauffman and Levin, 1987). The more “rugged” the landscape, the weaker is this correlation, so that following the local gradient can lead organisms to become stuck at local maxima from which they cannot escape by local mutations (though they might escape by crossover).

Therefore, if a vector of numbers faces a rugged fitness

landscape, with difficult features such as low local peaks and impassable moats, we can improve its evolvability for a vector of numbers by making its fitness function more hill-shaped. Let us call the process of making a fitness landscape more hill-shaped *acclivation*.¹

So, in a genome where G_1 is a vector of numbers that mutate by small amounts, and G_2 is a directed graph that feeds the numbers in G_1 through nodes that perform some function on the numbers from their input edges, eventually leading to a phenotype whose fitness determines the fate of the whole organism, we should expect selective pressure for genotypes $g_2 \in G_2$ to produce mappings that induce acclivation on the virtual fitness functions v_{g_2} . Evolution should favor graphs that put knobs in a position where they can hill-climb successfully.

Genome for Experimentation

To test the preceding hypothesis in a form in which acclivation will be visually apparent on plots printed on paper, we limit ourselves to genomes where g_1 and the phenotype are 2-dimensional vectors and g_2 is a graph connecting them. The whole genome is a directed graph where:

1. Two nodes, called the *knobs*, k_1 and k_2 , are designated to each hold a number in $[-1.0, +1.0]$, called an *initial activation*.
2. Two other nodes, p_1 and p_2 , are designated to hold the phenotype.
3. Zero or more additional nodes n_1, n_2, \dots
4. Each edge has a weight of either $+1.0$ or -1.0 .

Genotype–Phenotype Mapping

The phenotype is determined by a process of spreading activation, run for 10 timesteps. At each timestep, each node can have either an *activation* in $[-1.0, +1.0]$ or no activation. At timestep 0, only the knobs have activations: the numbers stored in the genotype. Each successive timestep, activations spreads from nodes (the ones with activations) to their neighbors. If none of a node’s incoming neighbors has an activation, its own activation (or lack of one) is unchanged. Otherwise, the activation of a node a_j at timestep $t + 1$ is calculated according to the following function:

$$a_j(t + 1) = T(a_j(t) + \sum_i W_{ij} a_i(t))$$

where W_{ij} is the weight of the incoming edge, if any, from node i to node j , and T is the following transfer function:

$$T(x) = \frac{2}{1 + \exp(-Sx)} - 1$$

¹From Latin *clivus*, meaning a slope or a hill, combined with the prefix *ad-* indicating in this context an upward slope or becoming more sloped.

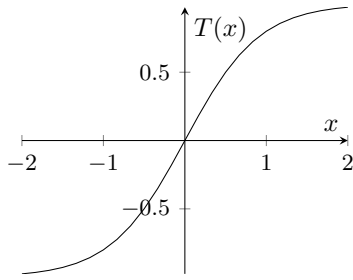


Figure 1: The transfer function T is a sigmoid function with attractors at ± 0.5 and a repeller at 0.

where $S = 2.1972274554893376$. This constant gives T , when iterated, attractors at ± 0.5 and a repeller at 0.

If a node does not have an activation at time t , then it does not figure into the above sum for calculating any other node's activation. At time $t = 0$, only the knobs have activations.

The phenotype is the vector $(a_{p_1}(10), a_{p_2}(10))$, i.e. the activations of the phenotype nodes after 10 timesteps. If p_1 or p_2 has not received an activation after 10 timesteps, then the genotype has no phenotype and is given a fitness of 0.0. This can happen if no edges provide a path from a knob to p_1 or p_2 .

Example The following table shows step-by-step how the spreading-activation algorithm calculates the phenotype for the simple genotype in Figure 2.

t	k_1	k_2	n_1	p_1	p_2
0	-0.659	1.000			
1	-0.659	0.358	-0.619	-0.619	0.800
2	-0.044	-0.319	-0.970	-0.886	0.854
3	0.769	-0.379	-0.975	-0.771	0.529
4	0.958	0.404	-0.861	-0.002	0.164
5	0.964	0.905	-0.686	0.782	0.554
6	0.948	0.968	-0.420	0.958	0.922
7	0.906	0.971	-0.118	0.970	0.970
8	0.699	0.968	0.850	0.968	0.972
9	-0.164	0.950	0.990	0.950	0.971
10	-0.853	0.698	0.964	0.700	0.971

At $t=0$, the genotype provides the initial activations of the knob nodes.

At $t=1$, n_1 and p_1 each receive an input of -0.659 from k_1 ; each gets an activation of $T(-0.659) = -0.619$. Similarly, p_2 receives an input of 1.000 from k_2 , giving p_2 an activation of $T(1.000) = 0.800$. Since the only input to k_1 comes from n_1 , and n_1 had no activation on timestep 0, k_1 's activation is unchanged.

At $t=2$, n_1 receives inputs along two edges: -0.659 from k_1 and -0.619 from itself. These add to n_1 's preceding activation, so n_1 's new activation becomes $T(-0.619 - 0.619 - 0.659) = -0.970$. k_1 now receives the -0.619 from n_1

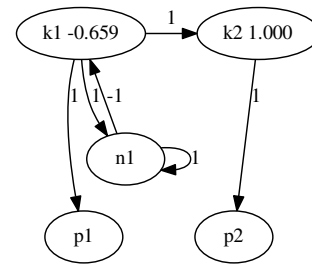


Figure 2: A simple genotype. The knob nodes are at the top, the phenotype nodes are at the bottom, and there is one additional node.

but the edge has weight -1 , so k_1 's activation becomes $T(-0.659 + 0.619) = 0.044$. k_2 receives an input of -0.659 from k_1 , so k_2 's activation becomes $T(0.358 - 0.659) = 0.319$. p_1 's activation becomes $T(-0.619 - 0.659) = -0.886$ and p_2 's becomes $T(0.800 + 0.358) = 0.854$.

Now that all the nodes have activations, the cycle continues: k_1 and n_1 interact and p_1 and p_2 essentially accumulate output from k_1 and k_2 , scaled back each timestep by the T function.

Finally, after 10 timesteps, p_1 's activation is 0.700 and p_2 's activation is 0.971, so this genotype's phenotype is $(0.700, 0.971)$.

Variation Operator

In generation 0 of the first epoch in each experiment, the population consists of genotypes containing only the two knob nodes and the two phenotype nodes, with up to four randomly placed edges with weights randomly chosen from $\{-1, +1\}$, and the knobs' initial activations chosen uniformly from $[-1.0, 1.0]$.

Each organism of each successive generation is generated by selecting one or two parents from the previous generation by tournament selection and making a child by a single mutation or by crossover. Crossover has a low probability, usually 0.02 or 0.05.

The possible mutations are: add a node, remove a node (but not a knob or phenotype node), add an edge, remove an edge, move an edge, or turn a knob. Knob-turning has a probability roughly equal to the sum of all the graph-edit mutations. Depending on the experiment, turning a knob chooses a knob delta from $\{-0.02, +0.02\}$ or from a normal distribution with mean 0 and $\sigma = 0.0$.

Population sizes range from 60 to 800 depending on the experiment. We omit some details of the variation operator here for lack of importance. The source code is publicly available at <https://github.com/bkovitz/acclivation>.

Experiments

In each experiment, we run the genome defined above against a different family of fitness functions and see what virtual fitness functions emerge. We only plot v_{g_2} , the virtual fitness function seen by the knobs, since we know of no way to plot fitness functions seen by a graph.

We found no reliably meaningful measure of acclivation. We tried running a hill-climbing algorithm on the virtual fitness functions but this yielded ambiguous results. For example, a higher fitness reached by the hill-climbing algorithm in many cases resulted not from acclivation but from extreme canalization: the graph forced the phenotype to a predetermined point regardless of the knobs. So, we plot no temporal dynamics of populations. Instead, we show only some representative individuals to illustrate the kinds of virtual fitness functions found.

Each experiment tries a “family” of fitness functions, because each experiment’s fitness function has a constant that changes randomly once per epoch: every 20 generations. This constant moves the peak of the fitness function to different places in phenotype space.

Experiment 1: Razorback

In this experiment, we ran the experimental genome against this fitness function, plotted in Figure 3(a):

$$w_{\Phi}(\phi) = 10.0 \cdot \hat{d}(\phi, P) \cdot \Lambda(|p_2 - p_1|; R) + \text{waves}(\phi; 30)$$

where:

P is a point (the peak) chosen randomly along the $y = x$ line each epoch;

$\hat{d}(\phi, P)$ is a measure of the proximity of ϕ to P equal to 0.0 for the maximum possible distance and 1.0 for zero distance:

$$\hat{d}(\phi, P) = \frac{\max - d(\phi, P)}{\max}$$

Λ is the “inverted-v” function: like an inverted-U function but peaking sharply at $x = 0$ and returning zero outside the radius R , set to 0.1 or 0.2 on different runs of the experiment:

$$\Lambda(x; R) = \begin{cases} 0 & \text{if } |x| > R \\ 1 - (\frac{x}{R})^2 & \text{if } |x| \leq R \end{cases}$$

and “waves” is a function that adds regular undulations, giving the overall fitness function an “egg carton” look, shown in Figure 3(a):

$$\text{waves}(\phi; \nu) = \cos(\nu p_1) \cdot \sin(\nu(p_2 + \frac{\nu}{2}))$$

So, this function rewards the phenotype up to 10 points for proximity to P , but only if the phenotype lies along a

narrow ridge running diagonally across phenotype space, complicated by the addition of a regular pattern of undulations. The undulations add local minima throughout the fitness landscape to trap searches that merely follow the local gradient.

Figure 3 shows an organism that evolved in this experiment. The virtual fitness function illustrates acclivation: there is a steep slope leading to a “butte” containing the global fitness peak, and the narrow ridge of the phenotype function is widened and distorted, making it climbable from different directions.

This organism also illustrates another fundamental way, aside from acclivation, of gaining evolvability: by restricting the range of m_{g_2} to exclude bad parts of the phenotype space. The genotype–phenotype mapping does not allow access to any points in phenotype space other than those along the center of the ridge.

Experiment 2: Circle

In this experiment, we ran the experimental genome against this fitness function:

$$w_{\Phi}(\phi) = 10.0 \cdot \hat{d}(\phi, P) \cdot \Lambda((p_1^2 + p_2^2) - r^2; R)$$

where r is the radius of a circle, R is the ridge radius as in the first experiment, and P is a point (the peak) chosen randomly along the circle at the start of each epoch. In words, the phenotype is rewarded up to 10.0 points for proximity to the peak, but only if the phenotype lies within R of the perimeter of the circle—a circular ridge. We set $r = 0.5$ and $R = 0.15$.

Figure 4 shows one organism that evolved in this experiment. It has evolved canalization for phenotypes near the circular ridge and decanalization for phenotypes in the center of the circle (knob-turnings quickly move the phenotype away from the center). All phenotypes outside the circle are inaccessible in this organism’s genotype–phenotype mapping.

Experiment 3: Moats

In this experiment, we run a modified version of the razorback fitness function: wherever $\text{waves}(\phi) \leq 0.5$, fitness is zero rather than slightly reduced; organisms with fitness zero are not allowed to reproduce; the “islands” where $\text{waves}(\phi) > 0.5$ have flat, neutral plateaus, so there is no smooth gradient to climb within any one island; and the islands are spaced further apart than in the razorback experiment. So, organisms can only cross from one island to a higher island by a single mutation. Ending the lineage of an organism that falls into the “moat” between islands simulates the tendency in nature for fitness landscapes be to “holey” (Gavrilets, 1997), requiring leaps over regions of non-viable genotypes in order to improve fitness.

Figure 5 shows an organism that successfully climbs the chain of islands. It has evolved a genotype–phenotype map-

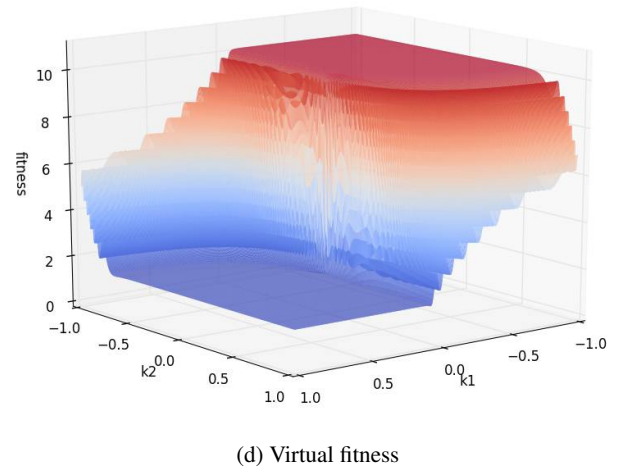
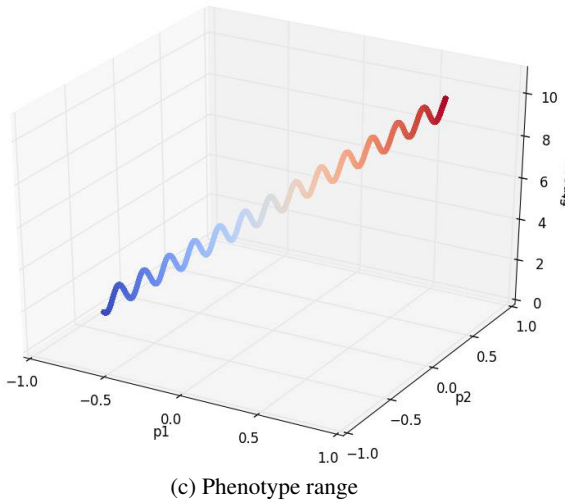
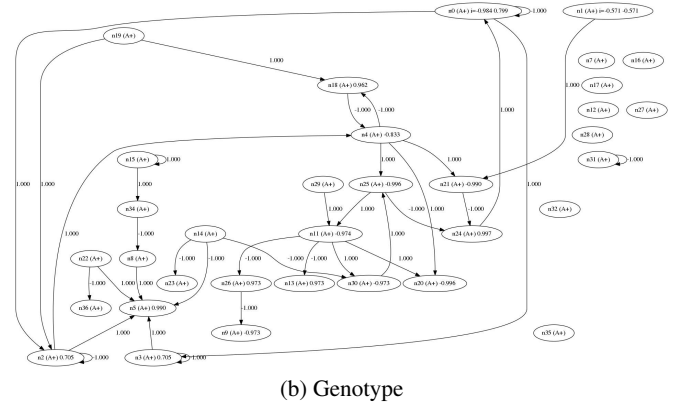
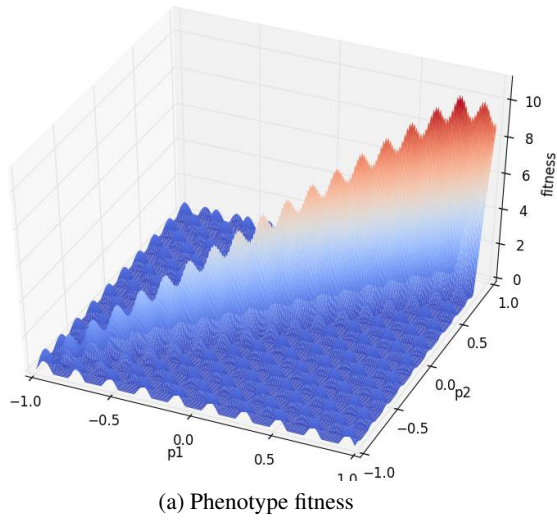


Figure 3: One organism from experiment 1, “Razorback”. The phenotype’s fitness function, a landscape filled with local maxima and one wavy narrow ridge (a), has become distorted into the roughly hill-climbable virtual fitness function (d) seen by the “knobs” of the genotype (b). In (c), the phenotype range, x, y values indicate points in phenotype space that have a preimage in knob space when the knobs are mapped through m_{g_2} . The z values are the fitnesses of those phenotypes (the same as are plotted in (a)). In (b), the knob nodes are at the top, the phenotype nodes are at the bottom, numbers preceded by “i=” are initial activation levels, and the other numbers are activation levels after 10 timesteps.

ping that squeezes the islands closer together in the virtual fitness function so that single knob-turns can leap the moats between them, as well as limiting the range of the virtual fitness function to the line along the centers of the islands.

Observations and Conclusions

The main result is that against these fitness functions, filled with traps that flummox direct evolution, a genome with continuously varying “knobs” mapped to its phenotype by a discontinuously varying topology or “graph” tends to evolve

increasing evolvability by (a) presenting the knobs with a more hill-shaped virtual fitness function and (b) restricting the range of the knob–phenotype mapping to exclude “bad” parts of the phenotype space. Close observation of genotypes and lineages revealed a number of subtleties regarding how and when this process happens, explained below.

Limitations on Generality

Modelable Fitness Functions and Genetic Memory
Over many generations, the graph accumulates a “memory”

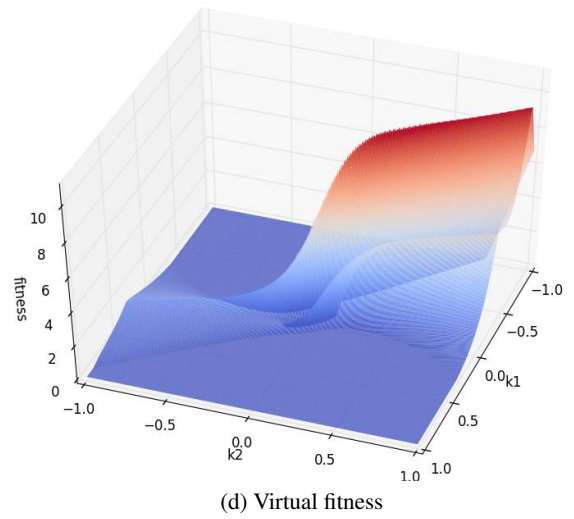
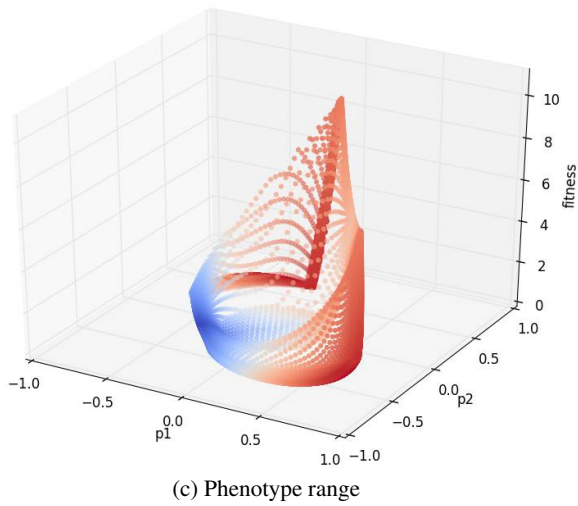
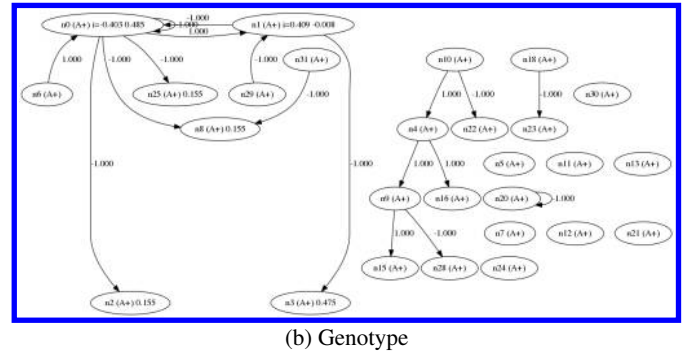
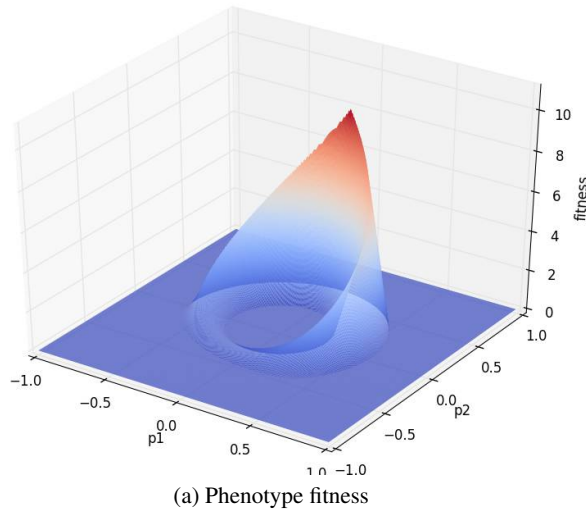


Figure 4: One organism from experiment 2, “Circle”.

of the family of whole-genome fitness functions, encoded in the form of *bias* in the way its lineage searches the phenotype space. This bias reflects invariants in the fitness family, such as where ridges occur and how they’re oriented, moat size, and where zero-fitness “deserts” consistently lie.

In effect, the graphs tended to evolve into models of the invariants in the family of fitness functions. This means that difficulties in modeling the invariants with a graph will shut down acclivation. For example, a graph can easily model the Razorback family because it lies along $y = x$, by disconnecting one knob and linking the other knob to both phenotype nodes. But a shifted Razorback family, say along $y = 2x - .4$, is much harder for the graph to model.

Knobs themselves cannot accumulate useful bias beyond being positioned where they will be mapped to high-quality

phenotypes. This bias can be effective, though: we often observed several lineages in a single population with knobs positioned far apart, each ready to capitalize if the whole-genome fitness function or the genotype–phenotype mapping changes to favor them again.

Non-Stationary Fitness Function We frequently witnessed the decay of a population’s genetic memory. Many times, a population that was responding quickly to shifts of the fitness peak, moving rapidly toward it one knob-turn per generation, lost its ability to do this when a few epochs went by with little or no movement in the fitness peak. When the peak stayed constant too long, selective pressure favored canalization: genotype–phenotype mappings that held the phenotype at the peak in the face of most mutations. In

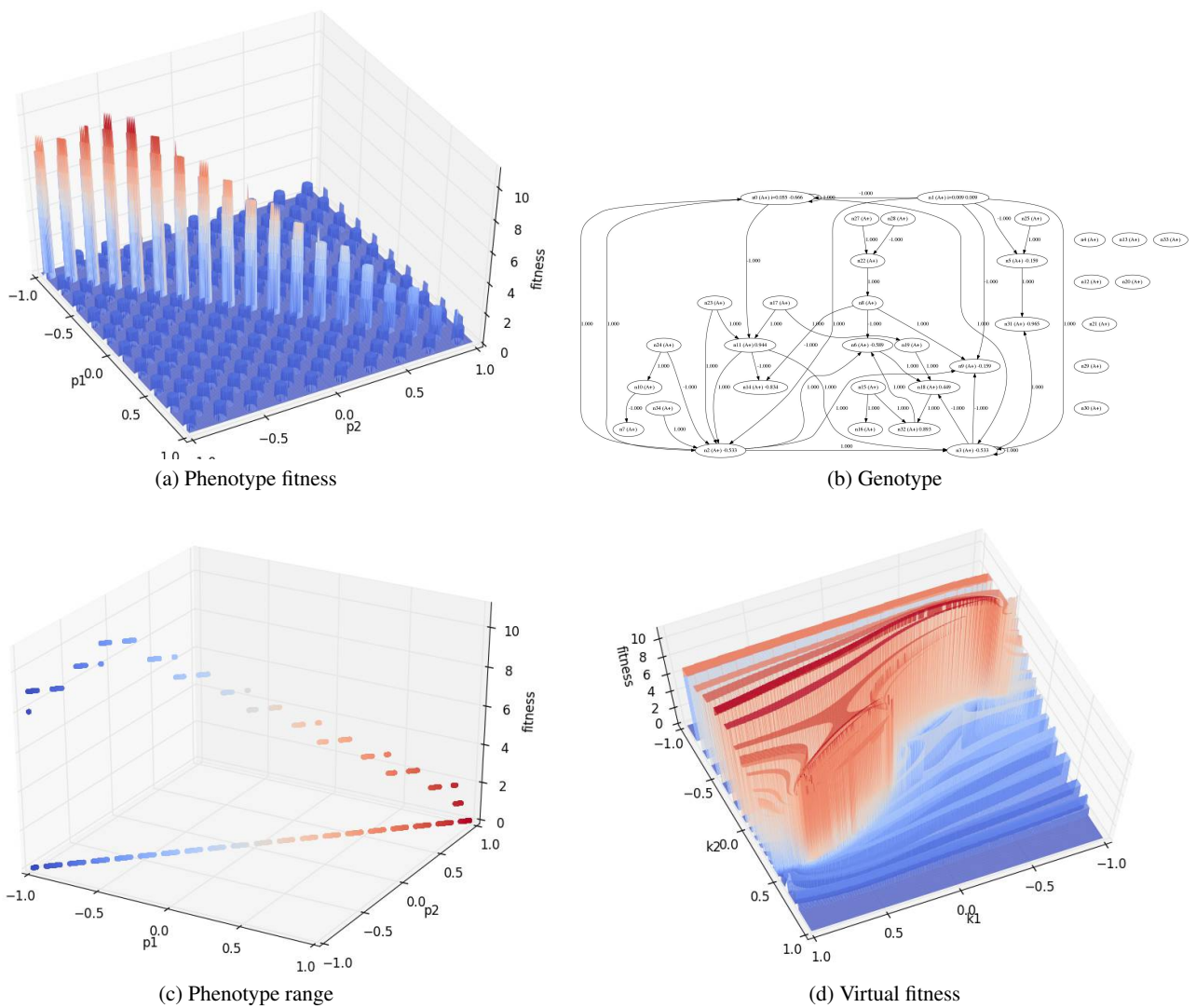


Figure 5: One organism from experiment 3, “Moats”.

that circumstance, if knob-turning alters the phenotype at all, it lowers fitness, so selective pressure favors rendering it inert—that is, making a genotype–phenotype mapping in which all points in knob-space map to the same point in phenotype space.

Genetic memory also frequently decayed when a knob had a value of -1.0 or $+1.0$. A knob-turning mutation that goes beyond the limits has no effect. At these times, knob-turning often lost its sensitivity to the virtual fitness gradient—and so improvements in fitness had to come by graph-edits alone, which can spoil previous acclivation.

Thus a non-stationary fitness function is most favorable for acclivation: one that changes frequently, shifting the peak, while retaining invariants that the graph can model in a stable way. See Reisinger et al. (2005) for discussion

of evolvability in connection with this kind of nonstationary fitness function, including a measure, *acquired evolvability*, of a genome’s ability to “represent” its invariants.

Mutation Rate When instead of limiting each offspring to a single mutation from its parent, we allowed a number of mutations proportional to the size of the parent (the number of nodes), genotypes tended to “bloat”, acquiring hundreds of disconnected nodes and edges. The problem is that when larger genotypes can make more mutations per generation, they have no incentive to optimize the way they respond to graph mutations. When each organism can only vary from its parent by a single mutation, those who do not optimize their mutation exposure are at a disadvantage in the race to the new peak at the start of each epoch. A lineage with an

unnecessarily large number of ways to make neutral mutations will tend to lose those races to lineages with the minimal amount of “padding”.

The Transfer Function We expected that nearly any transfer function typically used in simulated activation networks would induce acclivation of virtual functions (given appropriate fitness functions, etc.), but this was decidedly not the case. When we tried a simple $y = x$ function clamped within $[-1.0, 1.0]$, step functions, rectifier functions, and letting the constant in the T function stray far from S all produced much less acclivation as well as phenotypes of much lower fitness. The graphs could not “lock on” to the ridge, making knob-turning nearly useless for navigating up the fitness functions.

The T function has a peculiar characteristic that makes it suitable for these experiments, where the only constants in the genotype that are allowed to vary in small increments are those in the knobs. T is expansive in the range $-.28 < x < .28$ and contractive everywhere else. When a constant input, as from a knob, is fed into T repeatedly ten times, this yields a function $T(x + T(x + T(\dots)))$, which is expansive in $-.14 < x < .14$ and contractive everywhere else. This makes T well suited to forming a wide variety of functions that simultaneously dilate and compress different ranges of the phenotype space, by composition with itself alone—without constants. Activations from incoming edges a_i beyond the first edge make a node calculate the function $T(a + \sum_i a_i)$, giving compositions of T the ability to shift their output right or left.

Compositions of linear transfer functions can shift phenotype space but they can’t dilate or compress it. This makes it harder, perhaps impossible, to evolve an acclivated virtual fitness function. When the constant S is varied too far from that in T , the resulting function’s range of expansion quickly shrinks to a tiny region around $x = 0$ or grows to nearly the whole interval $[-1.0, 1.0]$.

Virtual Knobs

We ran variations on the above experiments where nodes other than the knobs were allowed to inherit constants. For example, we tried allowing non-knob nodes to inherit an initial activation. Under this condition, successful organisms tended to accumulate a collection of nodes with different constants, none of which were connected to the knobs and only one of which was connected to a phenotype node. They exploited the “move edge” mutation to make these collections of nodes function as a virtual knob. Both knob nodes were often disconnected from the rest of the graph.

The organisms seemed to prefer their virtual knobs. Virtual knobs are subject to evolutionary pressure determining how fast they turn, i.e. the probability distribution of knob-turning deltas. The hard-coded knobs are limited to deltas in the range of about ± 0.02 . The evolved virtual knobs tended

to turn much faster than our hard-coded knobs.

When we removed nodes with constants, we tried allowing more than one edge between nodes. The organisms evolved to exploit the “add edge” and “remove edge” mutations as knobs. The number of edges between two nodes effectively served as an adjustable multiplier.

To get the organisms to make use of our hard-coded knobs, necessary to examine virtual fitness functions whose domain is the knob settings, we had to purge the graph of all other constants capable of varying in small increments. This severely reduces evolvability.

Acknowledgements

We thank Jerry Swan (University of York), Zoltán Kocsis (University of Manchester), and Etienne Barnard (North-West University, South Africa) for helpful discussion and suggestions. We thank David Landy and Brad Rogers (Indiana University) and the anonymous reviewers for many valuable suggestions for improving the article. And thanks to Joonas Ilmavirta (University of Jyväskylä) for suggesting *acclivus* as the appropriate Latin root for “making sloped or hill-shaped.”

References

- Colegrave, N. and Collins, S. (2008). Experimental evolution: experimental evolution and evolvability. *Heredity*, 100(5):464.
- Gavrilets, S. (1997). Evolution and speciation on holey adaptive landscapes. *Trends in ecology & evolution*, 12(8):307–312.
- Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology*, 128(1):11–45.
- Kovitz, B. (2015). Experiments with cascading design. In *Proceedings of the 13th European Conference on Artificial Life (ECAL 2015), Workshop EvoEvo*.
- Reisinger, J., Stanley, K. O., and Miikkulainen, R. (2005). Towards an empirical measure of evolvability. In *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 257–264. ACM.

Continuous Long-Term Evolution of Genetic Programming

William B. Langdon¹ and Wolfgang Banzhaf²

¹ Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK

² Department of Computer Science and Engineering, Michigan State University, East Lansing, USA

Abstract

We evolve floating point Sextic polynomial populations of genetic programming binary trees for up to a million generations. Programs with almost 400 000 000 instructions are created by crossover. To support unbounded Long-Term Evolution Experiment LTEE GP we use both SIMD parallel AVX 512 bit instructions and 48 threads to yield performance of up to 149 billion GP operations per second, 149 giga GPops, on a single Intel Xeon Gold 6126 2.60 GHz server.

Introduction

Nature has had billions of years for evolution to work its way to the organisms we see today. Not only was a long time available to achieve results, but many generations passed before the present. In evolutionary biology, there is discussion about the long-term innovative capabilities of evolution. Some say, evolution happens on a short time-scale, and even a few hundred generations are enough to produce completely different species (Palumbo, 2001; Owen et al., 1990). Others maintain that natural evolution is an open-ended process that will continue to produce novelty, even if many millions of generations pass (Evans et al., 2012).

Thus different aspects are considered when studying long-term evolution. One aspect is continuity: If one wants to study evolution in the laboratory, one should strive to set up experiments similar to Nature's evolutionary "experiment" that go on for a long time, and are not disrupted. The other aspect is duration: To attempt to evolve for many generations, trusting in the turn-over of information during the evolutionary process. How does evolution proceed after 100, 1,000, 10,000 etc. generations of continued evolution? Does it stagnate? Does it continue to produce surprises?

Richard Lenski and his collaborators have used the evolution of *E.coli* bacterial strains in the laboratory to examine these questions. Since 1988, the evolution of these bacterial strains continues, with the experimental conditions being recorded and bacterial generations being frozen every so often to conserve a time-slice of evolution of these strains (Lenski, 1988). This natural system is studied with both aspects of long-term evolution in mind: The experiment has

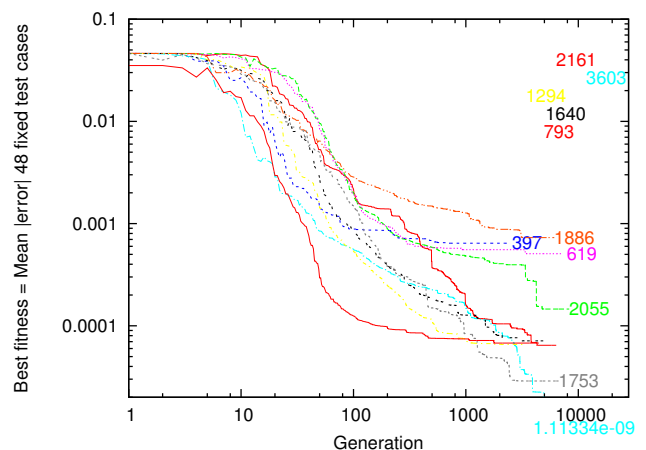


Figure 1: Evolution of mean absolute error in ten runs of Sextic polynomial (Koza, 1992) with population of 4000. (Runs aborted after first crossover to hit 15 million node limit.) End of run label gives number of generations when fitness got better (five shown at top right to avoid crowding).

run uninterrupted since 1988, and the fast reproductive cycle of bacteria allows to study evolution over many generations (Lenski et al., 2015). In 2019, 70,000 generations have been reached, with no end to evolution in sight.

We focus on one aspect of these long-term evolutionary experiments: The number of generations. The medium in which we consider this question, however, is computational. We started to investigate what happens if we allow artificial evolution, specifically genetic programming (GP) with only crossover (Koza, 1992; Banzhaf et al., 1998; Poli et al., 2008), to evolve for tens of thousands, even hundreds of thousands of generations.

With the continuous progress in technology, new hardware has become available, so we built a new GP engine based on Andy Singleton's GPQUICK (see next section). This allowed us to switch from the Boolean to the continuous domain and run experiments of up to a million generations. Excluding some special applications or Boolean benchmarks based on graphics hardware (GPUs), at up to

149 billion GP operations per second (149 giga-GPops, see Table 3), this appears to be the fastest single-computer GP system (Langdon, 2013, Tab. 3).

In the Boolean domain we found usually the population quickly found the best possible answer and then retained it exactly for thousands of generations (Langdon, 2017). Nonetheless under subtree crossover we reported interesting population change with trees continuing to evolve. Indeed we were able to report the first signs of an eventual end of bloat due to fitness convergence of the whole population. We can now report in the continuous domain we do see continual innovation and improvement in fitness like in the bacteria experiments. Figure 1 shows that although the rate of innovation falls (as in Lenski’s *E. Coli*¹ populations), typically better solutions are found even towards the end of the runs. In these runs, there are several hundred or even a few thousand generations where sub-tree crossover between evolved parents gave a better child.

We are going to run GP far longer than is normally done. Firstly in search of continual evolution but also noting that it is sometimes not safe to extrapolate from the first hundred or so generations. E.g., McPhee (McPhee and Poli, 2001, sect. 1.2) found that his earlier studies which had reported only the first 100 generations could not safely be extrapolated to 3,000 generations.

It must be admitted that without size control we expect bloat², and so we need a GP system not only able to run for a million generations³ but also able to process trees with well in excess of a 100 million nodes⁴. The new system we use is based on Singleton’s GPQuick (Singleton, 1994; Keith and Martin, 1994; Langdon, 1998), but enhanced to take advantage of both multi-core computing using pthreads and Intel’s SIMD AVX parallel floating point operations. Keith and Martin (1994) say GPQuick’s linearisation of the

¹The *E. Coli* genome contains 4.6 million DNA base pairs.

²GP’s tendency to evolve nonparsimonious solutions has been known since the beginning of genetic programming. E.g. it is mentioned in Jaws (Koza, 1992, page 7). Walter Tackett (Tackett, 1994, page 45) credits Andrew Singleton with the theory that GP bloat is due to the cumulative increase in non-functional code, known as introns. The theory says these protect other parts of the same tree by deflecting genetic operations from the functional code by simply offering more locations for genetic operations. The bigger the introns, the more chance they will be hit by crossover and so the less chance crossover will disrupt the useful part of the tree. Hence bigger trees tend to have children with higher fitness than smaller trees. See also Altenberg (1994); Angeline (1994). In Langdon (2017) we showed prolonged evolution can produce converged populations of functionally identical but genetically different trees comprised of the same central core of functional code next to the root node plus a large amount of variable ineffective sacrificial code.

³The median run shown in Figure 2 took 39 hours (mean 62 hours). Under ideal growing conditions, a million generations for *E. Coli* corresponds to about 38 years.

⁴Again referring to the extended runs in Figure 2, crossover creates highly evolved trees containing almost four hundred million nodes. These are by far the largest programs yet evolved.

GP tree will be hard to parallelise. Nevertheless, GPQUICK was rewritten to use 16 fold Intel AVX-512 instructions to do all operations on each node in the GP tree immediately. Leading to a single eval pass and better cache locality but at the expense of keeping a $T = 48$ wide stack of partial results per thread.

Although the populations never lose genetic diversity (Koza’s variety)⁵, with strong tournament selection (see Table 1) even the larger populations tend to converge to have identical fitness values. However 100% fitness convergence is only seen in long runs with smaller populations (500 or 48 trees). In contrast, in the Boolean domain (Langdon, 2017), even in the bigger populations (500) of that study, there are many generations where the whole population has identical fitness (but again variety is 100%).

The next section describes how GPQUICK was adapted to take advantage of Intel SIMD instructions able to process 16 floating point numbers in parallel and to use Posix threads to perform crossover and fitness evaluation on 48 cores simultaneously. The Experiments section describes the floating point benchmark (Table 1). Whilst the Results section describes the evolution of fitness and size and depth in populations of 4000, 500 and 48 trees. It finds the earlier predictions of sub-quadratic bloat (Langdon, 1999) and Flajolet limit ($\text{depth} \approx \sqrt{2\pi|\text{size}|}$ (Langdon, 2000b)) to essentially hold. More analysis can be found in the technical report (Langdon and Banzhaf, 2019). We finish with a short discussion about the continuous evolution permitted by floating point benchmarks and our conclusion that even something as simple as digital evolution in the Sextic polynomial genetic programming benchmark permits continuous innovation.

GPQUICK

First we describe how GPQUICK is used to do symbolic regression on a simple sixth order polynomial ($y = x^2(x-1)^2(x+1)^2$ known as the Sextic polynomial) and then how GPQUICK has been modified to run in parallel.

Sextic and GPQuick

Andy Singleton’s GPQUICK (Singleton, 1994) is a well established fast and memory efficient C++ GP framework. In steady state mode (Syswerda, 1990) it stores GP trees in just one byte per tree node. Using separate parent and child populations doubles this (although Koza (1992) shows doubling is not necessary). The 8 bit opcode per tree node allows GPQUICK to support a number of different functions and inputs. Typically (as in these experiments) the remaining opcodes are used to support about 250 fixed ephemeral random constants (Poli et al., 2008). In the Sextic polynomial we have the traditional four binary floating point operations

⁵Koza defines variety as the percentage of the population that has no genetically identical copy (Koza, 1992, p.93)

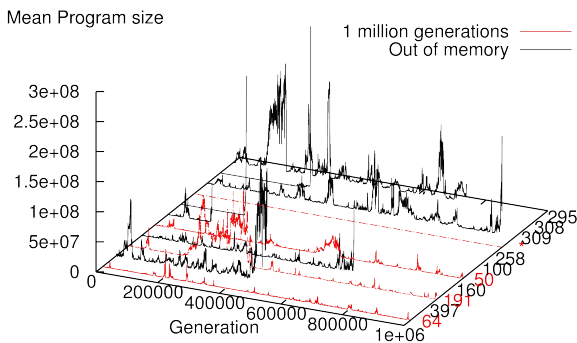


Figure 2: 11 extended runs pop=48. Numbers on right indicate size of largest tree before the run stopped in millions of nodes. One run (*) converged so that more than 90% of the trees contain just five nodes. Three of the other four runs that reached 1 million generations (red) took between half a day and five days. In all but one run (*) we see repeated substantial bloat (> 64 million nodes) and subsequent tree size collapse. Seven runs, in black, terminated due to running out of memory (on server with 46GB).

(+, -, \times and protected division), an input (x) and 250 constants. The constants are chosen uniformly at random from the 2001 floating point numbers from -1.000 to +1.000. By chance neither end point nor 0.000 were chosen (see Table 1).

The continuous test cases (x) are selected at random from the interval -1 to +1. At the same time the target value y is calculated (Table 1). Since both x and y are stored in a text file, there may be slight floating point rounding errors due to the standard float \leftrightarrow string conversions.

Whereas the Sextic polynomial is usually solved with 50 test cases (Langdon et al., 1999), since the AVX hardware naturally supports multiples of 16, in our experiments we change this to 48 (i.e. 3×16) (Table 1). The multi-core servers we use each support 48 threads and in the longest extended runs, we reduce the population to 48 (whereas in Langdon (2017) the smallest population considered contained 50 trees).

AVX GPQuick

GPQUICK stores the GP population by flattening each tree into a linear buffer, with the root node at the start. To avoid heap fragmentation the buffers are all of the same size. The buffer is interpreted once per test case by multiple recursive calls to EVAL and the tree's output is the return value of the outermost EVAL. Each nested EVAL moves the instruction pointer one position forward in the tree's buffer, decodes the opcode there and calls the corresponding function. In the case of inputs x and constants a value is returned via EVAL immediately, whereas ADD, SUB, MUL and DIV will each call EVAL twice to obtain their arguments before operating on them and returning the result. For speed GPQUICK's

FASTEVAL does an initial pass through the buffer and replaces all the opcodes by the address of the corresponding function that EVAL would have called. This expands the buffer 16-fold, but the expanded buffer is only used during evaluation and can be reused by every member of the population. Thus, originally, EVAL processed the tree $T + 1$ times (for $T=48$ test cases).

The Intel AVX instructions process up to 16 floating point data simultaneously. The AVX version of EVAL was rewritten to take advantage of this. Indeed as we expect trees that are far bigger than the CPU cache (≈ 16 million bytes, depending on model), EVAL was rewritten to process each tree's buffer only once. This is achieved by EVAL processing all of the test cases for each opcode, instead of processing the whole of the tree on one test case before moving on to the next test case. Whereas before each recursive call to EVAL returned a single floating point value, now it has to return 48 floating point values. This was side stepped by requiring EVAL to maintain an external stack where each stack level contains 48 floating point values. The AVX instructions operate directly on the top of this stack and EVAL keeps track of which instruction is being interpreted, where the top of the stack is, and (with PTHREADS) which thread is running it. Small additional arrays are used to allow fast translation from opcode to address of eval function, and constant values. AVX instructions are used to speed loading each constant into the top stack frame. Similarly all 48 test cases (x) are rapidly loaded on to the top of the stack. However, the true power of the implementation comes from being able to use AVX instructions to process the top of the stack and the adjacent stack frame (holding a total of 96 floats) in essentially three instructions to give 48 floating point results.

The depth of the evaluation stack is simply the depth of the GP tree. GPQUICK uses a fixed buffer length for every individual in the GP population. This is fixed by the user at the start of the GP run. Fixing the buffer size also sets the maximum tree size. Although in principle this only places a very weak limit on GP tree depth, it has been repeatedly observed (Langdon, 2000b) that evolved trees are roughly shaped like random trees. The mathematics of trees is well studied (Sedgewick and Flajolet, 1996) in particular the depth of large random binary trees tends to a limit $2\sqrt{\pi \lceil \text{treesize}/2 \rceil} + O(\text{tree size}^{1/4+\epsilon})$ (Sedgewick and Flajolet, 1996, page 256). (See Flajolet limit in Figure 4.) Thus the user-specified tree size limit can be readily converted into an expected maximum depth of evolved trees. The size of the AVX eval stack is set to this plus a suitable allowance for random fluctuations and $O(\text{tree size}^{1/4+\epsilon})$. Note, with very large trees, even allowing for the number of test cases and storing floats on the stack rather than byte-sized opcodes, the evaluation stack is considerably smaller than the genome of the tree whose fitness it is calculating. Additional details can be found in Langdon and Banzhaf (2019).

PTHREADS GPquick

The second major change to GPQUICK was to delay fitness evaluation so that the whole new population can have its fitness evaluated in parallel. As trees are of different sizes, each fitness evaluation will require a different time. Therefore which tree is evaluated by which thread is decided dynamically. Due to timing variations, even in an otherwise identical run, which tree is evaluated by which thread may be different. However great care is taken so that this cannot affect the course of evolution. E.g., pseudo random numbers are only generated in sequential code.

EVAL requires a few data arrays. These are all allocated at the start of the GP run. Those that are read only can be shared by the threads. Each thread requires its own instance of read-write data. To avoid “false sharing”, care is taken to align read-write data on cache line boundaries (64 bytes), e.g. with additional padding bytes and `((aligned))`. This ensures each thread writes to its own cache lines and therefore these cached data are not shared with other threads.

Surprisingly an almost doubling of speed was obtained by also moving crossover operations to these parallel threads. Since crossover involves random choices of parents and subtrees these were unchanged but instead of performing the crossover immediately a small amount of additional information was retained and to be read later by the threads. This allows the crossover to be delayed and performed in one of 48 C++ pthreads. The results are identical but give an additional \approx two-fold speed up.

Experiments

We use the well known Sextic polynomial benchmark (Koza, 1994, Tab. 5.1). Briefly, the task given to GP is to find an approximation to a sixth order polynomial, $x^6 - 2x^4 + x^2$, given only a fixed set of samples, i.e., a fixed number of test cases. For each test input x we know the anticipated output $f(x)$, see Table 1. Of course the real point is to investigate how GP works and how GP populations evolve over time. We ask ourselves whether it is possible for GP to continue to find improvements, even for such a simple continuous problem, as Lenski’s *E. Coli* experiments are showing, or, like the Boolean case (Langdon, 2017), whether the GP population will get stuck early on and from then on never make further progress. Note that we here make use of crossover exclusively, so no random mutations are allowed to introduce any new genetic material during the run. All the variation the algorithm can make use of must be present in the first generation.

We ran three sets of experiments. In the first the new GP systems was set up like the original Sextic polynomial runs which reported phenotypic convergence (Langdon et al., 1999, Fig. 8.5). The first set uses a population of 4000, the second 500 and the last 48.

Table 1: Long term evolution of Sextic polynomial symbolic regression binary trees

Terminal set:	X, 250 constants between -0.995 and 0.997
Function set:	MUL ADD DIV SUB
Fitness cases:	48 fixed input -0.97789 to 0.979541 (randomly selected from -1.0 to +1.0 input).
	Target $y = xx(x-1)(x-1)(x+1)(x+1)$
Selection:	Tournament size 7 with fitness = $\frac{1}{48} \sum_{i=1}^{48} GP(x_i) - y_i $
Population:	Panmictic, non-elitist, generational.
Parameters:	Initial population (4000) ramped half and half Koza (1992) depth between 2 and 6. 100% unbiased subtree crossover. 100 000 generations (stop run if any tree reaches limit $15 \cdot 10^6$).

DIV is protected division ($y \neq 0$)? x/y : 1.0f

Crossover

Each generation is created entirely using Koza’s two parent subtree crossover (Koza, 1992). (GPQuick creates one offspring per crossover.) For simplicity and in the hope that this would make GP populations easier to analyse, both subtrees, the one to be removed and the one to be inserted are chosen uniformly at random. That is, we do not use Koza’s bias in favour of internal nodes (functions) at the expense of external nodes (leafs or inputs). Instead, the root node of the subtree (to be deleted or to be copied) is chosen uniformly at random from the whole of the parent tree. This means there is more chance of subtree crossover simply moving leaf nodes and so many children will differ from the root node donating parent by just one leaf.

As mentioned above, once fitness evaluation has been sped up by parallel processing, for very long trees producing the child is a surprisingly large part of the remaining run time and so it, too, can be implemented in parallel. However, the choice of crossover points is done in sequential code and remains unaffected by multithreading. This ensures the variability introduced by multiple parallel threads does not change the course of evolution.

Fitness Function

The fitness of every member of every generation is calculated using the same fitness function as (Koza, 1994, Tab. 5.1). That is, barring rounding errors (previous page), fitness is given by the mean of the absolute difference between the value returned by the GP tree on each test case and the Sextic polynomial’s value for the same test input (see Table 1). We use tournament selection to choose both parents.

Like (Koza, 1994, Tab. 5.1), we also keep track of the number of test cases where each tree is close to the target (i.e. within 0.01, known as a “hit”). The number of hits is used for reporting the success of a GP run. It is not used

Table 2: 10 Sextic polynomial runs with population 4000

Gens	err10 ⁻⁹	impr ⁶	hits	size10 ⁶	x^i	conv	ops10 ⁹
6370	64487	2139	48	14.329	1.200	3981	58.2
8298	145796	2040	48	14.102	1.916	3982	57.4
2323	642006	389	47	13.441	1.387	3995	51.6
7119	507600	608	48	13.668	1.589	3997	55.0
11750	1	3583	48	13.854	1.364	3989	49.8
3412	65561	1277	48	14.348	1.625	3986	45.4
5106	71288	1615	48	14.233	1.146	3988	53.6
6112	728757	1871	48	14.500	1.254	3983	52.9
6679	28853	1741	48	14.022	1.396	3998	43.4
4454	67817	790	48	14.900	1.227	3997	54.9

⁶Figure 1 gives number of generations which improve on their parents, whereas here we give strictly better than anything previously evolved. Hence slight differences.

internally during a GP run. Also, our GP runs do not stop when a solution is found (48 hits) but continue until either the user-specified number of generations is reached or bloat means the GP runs out of memory.

Where needed, floating point calculations are done in a fixed order, to avoid parallelism creating minor changes in calculated fitness, which could quickly cause otherwise identical runs to diverge because of implementation differences in parallel calculations.

Results

Results Population 4000 trees

In the first set of experiments, we use the standard population of 4000 trees. Table 2 summarises the results of 10 runs. In all cases GP found a reasonable approximation to the target (the Sextic polynomial). Indeed in all but one run (47 hits) the best trees score 48 out of 48 possible hits. I.e. they are within 0.01 on all 48 test cases. Indeed in most cases the average error was less than 10⁻⁴. Figure 1 shows that GP tends to creep up on the best match to the training data. Typically after several thousand generations, GP has progressively improved by more than a thousand increasingly small steps. (See Table 2 column 3 and Figure 1).

In all ten runs with a population of 4000, we see enormous increases in size and all but one are stopped as they hit the size limit (15 000 000) before reaching 100 000 generations. Column 5 in Table 2 gives the size (in millions) of the largest evolved tree in each run. The log-log plot in Figure 3 shows a typical pattern of subquadratic (Langdon, 2000a) increase in tree size. The straight line shows a power law fit. In this run the best fit has an exponent of 1.2. Column 6 of Table 2 shows that the best fit between generations ten and a thousand for all 10 runs varies between 1.1 and 1.9.

As expected not only do programs evolve to be bigger but also they increase in depth. As described above highly evolved trees tend to be randomly shaped and so as expected

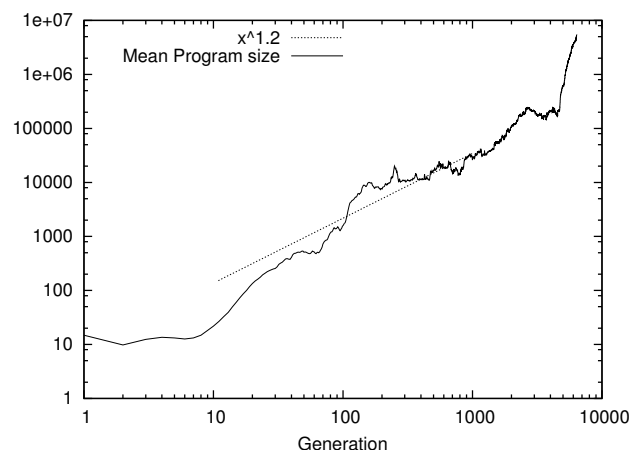


Figure 3: Evolution of tree size in first Sextic run (population 4000). (This run aborted after 6370 generations by first crossover to hit 15 million node limit.) Straight line shows best RMS error power law fit between generation 10 and 1000, $y = 8.65x^{1.2001}$

tend to lie near the Flajolet limit, depth $\approx \sqrt{2\pi|\text{size}|}$ (see Figure 4). (This is also true in the pop=500 and pop=48 runs, see following sections.)

In all ten runs we see some phenotypic convergence. The “conv” column in Table 2 shows the peak fitness convergence. That is, out of 4000, the number of trees having exactly the same fitness as the best in the population. Typically at the start of the run (see Figure 5), the population contains mostly trees with poorer fitness, but later in the run the population begins to converge and towards the end of the run we may see hundreds of generations where more than 90% of the population have identical fitness. Under these circumstances, even with a tournament size as high as 7, many tournaments include potential parents with identical fitness. These, and hence the parents of the next generation, are decided entirely randomly. However, even in the most converged population there are at least two individuals with worse fitness. (In Figure 5 it is at least 19.) As we saw with the Boolean populations (Langdon, 2017), even this small number can be enough to drive bloat (Langdon and Poli, 1997) (albeit at a lower rate).

Results Population 500 trees

We repeated the GP runs but allowed still larger trees to evolve by reducing the population from 4000 to 500 and splitting the available memory between fewer trees. Table 3 summarises 6 of these runs. Notice two runs do not really solve the problem and only hit less than half the test cases (see “hits” column in Table 3). Nonetheless, in all cases evolution continues to make progress and each GP run finds several hundred or more small improvements (third column in Table 3).

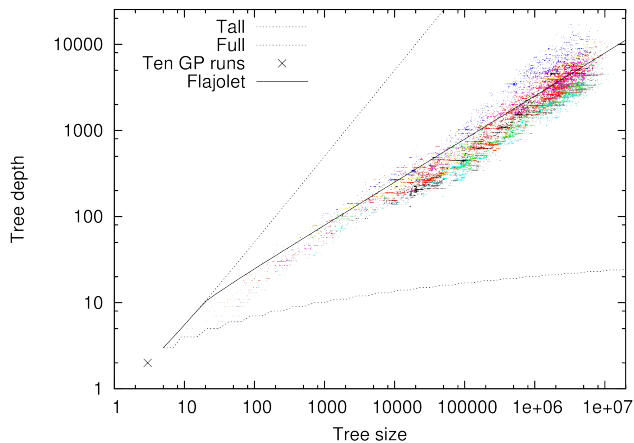


Figure 4: Plot of size and depth of the best individual in each generation for 10 Sextic polynomial runs with population of 4000. Binary trees must lie between short fat trees (lower curve “Full”) and “Tall” stringy trees. Most trees are randomly shaped and lie near the Flajolet limit (depth $\approx \sqrt{2\pi|\text{size}|}$, solid line, note log-log scales).

Table 3: 6 Sextic polynomial runs with population 500

Gens	err10 ⁻⁶	impr	hits	size10 ⁶	x^i	conv	ops10 ⁹
111582	538	3545	47	399.594	1.558	500	93.8
23937	34313	757	18	202.439	1.736	500	117.3
35783	307	3484	48	227.488	1.436	500	95.8
43356	18373	929	22	267.416	2.181	500	149.2
27713	137	5852	48	327.253	1.928	500	138.9
103953	1765	664	48	230.106	1.408	500	69.6

Since we have deliberately extended the space available to GP trees, it is no surprise that the trees grow even bigger than before (column 5 in Table 3). Again bloat is approximately following a power law. Although in one unsuccessful run we see a power law exponent greater than 2, mostly growth is at a (sub-quadratic) rate similar to the bigger population runs (1.4–2.2 v 1.1–1.9, column 6 in Table 2 (pop 4000)).

Unlike with the large populations, all the runs with populations of 500 trees showed some cases of complete fitness convergence (“conv” column in Table 3 is 500). For example, in the first Sextic polynomial pop=500 run, the whole population has identical fitness 33 143 times (30% of the run). If we concentrate upon the last fitness improvement in generation 108 763 (2819 before the end of the run). This new improved Sextic polynomial performance takes over the whole population in half a dozen generations. However it fails to totally dominate the population in 861 (31%) of the remaining generations. Even though the mean number of lower fitness children is less than one (0.38) it is not zero, and this (given nearly three thousand generations) is still enough to double the average size of the trees.

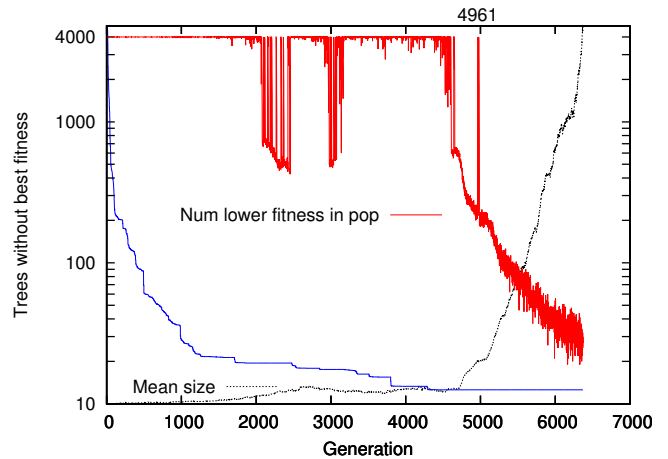


Figure 5: Fitness convergence in first Sextic polynomial pop=4000 run. Perhaps because of the continual discovery of better trees before generation 4975 and the larger population size, although the number of tree without the best fitness falls, unlike in the earlier Boolean problem (Langdon, 2017) it never reaches zero. Notice tiny fitness improvement in generation 4961 resets the population for ten generations. (Mean prog size (linear scale, dotted black) and best fitness (log, blue) plotted in the background.)

Results Population 48 trees

In the final experiments the population was reduced still further to allow even larger trees to be evolved (Figure 2). These smallest population runs were run with a population of 48, since this should readily map to the available Intel multi-core servers.

With the small population, none of the runs solve the problem. Indeed only three runs got close on 40 or more test cases (see Table 4). Of the remaining eight, only one finds a large number of fitness improvements. Seven runs have only between 3 and 30 generations with fitness improvements, column 3 in Table 4. In three of these, the population gets trapped at trees with just three nodes which evaluate to constants 0.0626506, 0.069169 and 0.0830508, although the population eventually escapes and large trees evolve by the end of the run. Except for these three runs, all the other runs contain populations where every member of the population has identical fitness. Therefore their maximum convergence is 48 (see “conv” column in Table 4). The final column is average speed, in giga GP operations/second.

For almost the whole of the first run with 48 trees the best fitness in the population is fixed but once trees get big enough further size changes are essentially random (Figure 2). The best fitness found in this run is given by robust trees which always return a midpoint value which only passes close to four test points. Trees which closely matched more test points were discovered in the first nineteen generation of this run. However, in terms of fitness, they scored worse than a constant and so went extinct.

Table 4: 11 Sextic polynomial runs with population of 48

Gens	err10 ⁻⁶	impr	hits	size10 ⁶	x^i	conv	ops10 ⁹
1000000	46215	11	16	63.920	1.633	48	36.5
491618	2748	745	46	396.576	2.060	48	34.9
1000000	46215	7	13	190.654	1.448	48	57.4
689414	4857	448	40	159.949	1.260	48	38.1
1000000	46215	8	14	50.365	1.701	48	26.2
143251	46215	11	14	99.541	1.672	48	54.1
212528	46650	30	14	257.766	na	42	26.7
1000000	46730	3	14	0.000	na	42	.004
958147	23259	1683	18	308.958	1.791	48	53.5
294098	47174	3	12	308.121	na	43	24.5
757830	2985	2921	44	294.821	1.320	48	50.2

Is there a Limit to Evolution?

In the Sextic Polynomial experiments with larger populations there is no hint of either evolution of fitness or bloat totally stopping. In the smaller populations, it is both possible to run evolution for longer and to allow trees to be even larger. Four of the eleven pop=48 runs reached a million generations but in the remaining seven, bloat ran into memory limits and halted the run. Only in one run did we see anti-bloat, in which the population converged in a few generations on a small high fitness tree which crossover was able to replicate across a million generations. Interestingly two other runs found similar solutions but after thousands of generations crossover found bloated version of them.

In the binary 6-Mux Boolean problem (Langdon, 2017) there are only 65 different fitness values. Therefore the number of fitness improvements is very limited. An end to bloat was found. By which we mean it was possible for trees to grow so large that crossover was unable to disrupt the important part of their calculation next to the root node and many generations were evolved where everyone had identical fitness. This led to random selection and random fluctuations in tree size, i.e. enormous trees but without a tendency for progressive endless growth.

This did not happen here. Even in some of the smallest Sextic polynomials runs, we are still seeing innovation in the second half of the run, with tiny fitness improvements being created by crossover between enormous parents. Also we are still slightly short of total fitness convergence.

However, there is a strong relationship between the size of the population and the success of the runs. All runs of size 4000 were successful, half of the runs of size 500 were successful, but none of the runs of size 48 were successful.

Even with populations containing Sextic polynomial trees of hundreds of millions of nodes, crossover can still be disruptive and frequently even tiny populations can contain a tree of lower fitness. This is sufficient to provide some pressure (over thousands of generations) for tree size to increase on average.

Can bloat continue forever? It is still difficult to be definitive in our answer. We have seen cases where it does not and of course there are plenty of techniques to prevent bloat (Poli and McPhee, 2013). But we see other cases where crossover over thousands of generations can create an innovative child which allows bloat into a converged population of small trees. Perhaps more interestingly, we see crossover finding fitness improvement in bloated trees after many thousand of generations.

Conclusions

Evolving binary Sextic polynomial trees for up to a million generations, during which some programs grow to four hundred million nodes, suggests even a simple GP floating point benchmark allows long-term fitness improvement over thousands of generations.

The availability of multi-core SIMD capable hardware has allowed us to push GP performance on single computers with floating point problems to that previously only approached with sub-machine code GP operating in discrete domains (Poli and Langdon, 1999; Poli and Page, 2000). This in turn has allowed GP runs far longer than anything previously attempted whilst evolving far bigger programs.

Without size or depth limits or biases crossover with brutal selection pressure tends to evolve very large non-parsimonious programs, known in the GP community as bloat (Koza, 1992, page 617). (See also footnote 2 on second page.) After a few initial generations, GP tree bloat typically follows a sub-quadratic power law (Langdon, 2000a). But eventually effective selection pressure (Nordin, 1997, sec. 14.2), (Banzhaf et al., 1998, page 187), (Stephens and Waelbroeck, 1999; Langdon and Poli, 2002) within highly evolved populations falls, leading to bloat at a reduced rate. However in this continuous domain we only see the chaotic lack of bloat found in long-running Boolean problems (Langdon, 2017) in a few unsuccessful runs with tiny populations (red plots in Figure 2). Nevertheless in all cases bloated binary trees evolve to be randomly shaped and lie close to Flajolet's square root limit.

Acknowledgements

This work was inspired by conversations at Dagstuhl Seminar 18052 on Genetic Improvement of Software.

The new parallel GPQuick code is available via <http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/gp-code/GPavx.tar.gz>

References

- Altenberg, L. (1994). The evolution of evolvability in genetic programming. In Kinnear, Jr., K. E., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press.
- Angeline, P. J. (1994). Genetic programming and emergent intelligence. In Kinnear, Jr., K. E., editor, *Advances in Genetic Programming*, chapter 4, pages 75–98. MIT Press.

- Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998). *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, San Francisco, CA, USA.
- Evans, A. R., Jones, D., Boyer, A. G., Brown, J. H., Costa, D. P., Ernest, S. M., Fitzgerald, E. M., Fortelius, M., Gittleman, J. L., Hamilton, M. J., et al. (2012). The maximum rate of mammal evolution. *Proceedings of the National Academy of Sciences*, 109(11):4187–4190.
- Keith, M. J. and Martin, M. C. (1994). Genetic programming in C++: Implementation issues. In Kinnear, Jr., K. E., editor, *Advances in Genetic Programming*, chapter 13, pages 285–310. MIT Press.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
- Koza, J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.
- Langdon, W. B. (1998). *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, volume 1 of *Genetic Programming*. Kluwer.
- Langdon, W. B. (1999). Linear increase in tree height leads to sub-quadratic bloat. In Haynes, T. et al., editors, *Foundations of Genetic Programming*, pages 55–56, Orlando, Florida, USA.
- Langdon, W. B. (2000a). Quadratic bloat in genetic programming. In Whitley, D. et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 451–458, Las Vegas, Nevada, USA. Morgan Kaufmann.
- Langdon, W. B. (2000b). Size fair and homologous tree genetic programming crossovers. *Genetic Programming and Evolvable Machines*, 1(1/2):95–119.
- Langdon, W. B. (2013). Large scale bioinformatics data mining with parallel genetic programming on graphics processing units. In Tsutsui, S. and Collet, P., editors, *Massively Parallel Evolutionary Computation on GPGPUs*, Natural Computing Series, chapter 15, pages 311–347. Springer.
- Langdon, W. B. (2017). Long-term evolution of genetic programming populations. In *GECCO 2017: The Genetic and Evolutionary Computation Conference*, pages 235–236, Berlin.
- Langdon, W. B. and Banzhaf, W. (2019). Faster genetic programming GPquick via multicore and advanced vector extensions. Technical Report RN/19/01, University College, London, London, UK.
- Langdon, W. B. and Poli, R. (1997). Fitness causes bloat. In Chawdhry, P. K. et al., editors, *Soft Computing in Engineering Design and Manufacturing*, pages 13–22. Springer-Verlag London.
- Langdon, W. B. and Poli, R. (2002). *Foundations of Genetic Programming*. Springer-Verlag.
- Langdon, W. B., Soule, T., Poli, R., and Foster, J. A. (1999). The evolution of size and shape. In Spector, L. et al., editors, *Advances in Genetic Programming 3*, chapter 8, pages 163–190. MIT Press.
- Lenski, R. E. (1988). Experimental studies of pleiotropy and epistasis in *Escherichia coli*. I. variation in competitive fitness among mutants resistant to virus T4. *Evolution*, 42:425–432.
- Lenski, R. E. et al. (2015). Sustained fitness gains and variability in fitness trajectories in the long-term evolution experiment with *Escherichia coli*. *Proceedings of the Royal Society B*, 282(1821).
- McPhee, N. F. and Poli, R. (2001). A schema theory analysis of the evolution of size in genetic programming with linear representations. In Miller, J. F. et al., editors, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 108–125, Lake Como, Italy. Springer-Verlag.
- Nordin, P. (1997). *Evolutionary Program Induction of Binary Machine Code and its Applications*. PhD thesis, der Universitat Dortmund am Fachereich Informatik.
- Owen, R. B., Crossley, R., Johnson, T. C., Tweddle, D., Kornfield, I., Davison, S., Eccles, D. H., and Engstrom, D. E. (1990). Major low levels of Lake Malawi and their implications for speciation rates in cichlid fishes. *Proceedings of the Royal Society (B)*, 240(1299):519–553.
- Palumbo, S. (2001). *The Evolution Explosion*. Norton.
- Poli, R. and Langdon, W. B. (1999). Sub-machine-code genetic programming. In Spector, L. et al., editors, *Advances in Genetic Programming 3*, chapter 13, pages 301–323. MIT Press.
- Poli, R., Langdon, W. B., and McPhee, N. F. (2008). *A field guide to genetic programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>. (With contributions by J. R. Koza).
- Poli, R. and McPhee, N. F. (2013). Parsimony pressure made easy: Solving the problem of bloat in GP. In Borenstein, Y. and Moraglio, A., editors, *Theory and Principled Methods for the Design of Metaheuristics*, Natural Computing Series, pages 181–204. Springer.
- Poli, R. and Page, J. (2000). Solving high-order Boolean parity problems with smooth uniform crossover, sub-machine code GP and demes. *Genetic Programming and Evolvable Machines*, 1(1/2):37–56.
- Sedgewick, R. and Flajolet, P. (1996). *An Introduction to the Analysis of Algorithms*. Addison-Wesley.
- Singleton, A. (1994). Genetic programming with C++. *BYTE*, pages 171–176.
- Stephens, C. and Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evolutionary Computation*, 7(2):109–124.
- Syswerda, G. (1990). A study of reproduction in generational and steady state genetic algorithms. In Rawlings, G. J. E., editor, *Foundations of genetic algorithms*, pages 94–101. Morgan Kaufmann, Indiana University. Published 1991.
- Tackett, W. A. (1994). *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA.

The impact of environmental history on evolved robot properties

Karine Miras¹ and A.E. Eiben¹

¹Vrije Universiteit Amsterdam, Amsterdam, Netherlands
k.s.m.a.dasilvamisdearaujo@vu.nl

Abstract

This paper studies the effects of changing environments on the evolution of bodies and brains of modular robots. Our results indicate that environmental history has a long lasting impact on the evolved robot properties. We show that if the environment gradually changes from type A to type B, then the evolved morphological and behavioral properties are very different from those evolving in a type B environment directly. That is, we observe some sort of “genetic memory”. Furthermore, we show that gradually introducing a difficult environment helps to reach fitness levels that are higher than those obtained under those difficult conditions directly. Finally, we also demonstrate that robots evolved in gradually changing environments are more robust, i.e., exhibit a more stable performance under different conditions.

Introduction

A widely acknowledged ground truth about natural evolution is that the environment largely determines the evolved lifeforms (Darwin, 2004; Sapolsky, 2017). However, within the field of artificial evolution, specifically robot evolution, things can be different and as of today there is not much evidence for this effect. As a potential explanation it can be noted that Evolutionary Robotics has a historical emphasis on evolving only the robot brains (controllers) and robot systems with evolvable forms (morphologies, bodies) have been paid relatively little attention, cf. (Floreano et al., 2008; Doncieux et al., 2015). Furthermore, existing studies on morphologically evolving robots focus on other aspects and only a few address the effect of the environment directly, e.g., (Auerbach and Bongard, 2014). Last, but not least, the effect might be hard to demonstrate with robots that have less complex morphologies and shorter evolution periods than the plants and animals observed in Nature.

Our own experience with a large number of different environments and various system setups seems to support the latter explanation.¹ Specifically, we repeatedly found that even very different environments can lead to the same evolved robot morphologies. After many experiments we identified

¹All our previous studies and the present research is done in simulations – just like the related work of others.

and investigated two environments that gave rise to measurably different morphological features in the evolved populations of modular robots (Miras and Eiben, 2019).

In this paper, we consider these two environments (described later on as “Plain” and “Tilted”) and define a couple of other ones that represent intermediary stages between them. This allows us to investigate the evolutionary dynamics when environment A is gradually transformed into environment B and compare this with evolving in constant environments A and B. In other words, we can research the effect of what can be called the *environmental history*. Our initial hypothesis is that starting evolution in a simple world (A) and smoothly transitioning into a difficult world (B) will make adaptation easier than starting directly in B. Hence, we expect a better adapted final population with more prominent B-type features and corresponding fitness values.

To be specific, we run experiments in constant worlds (A and B) and dynamically changing worlds (two systems that both gradually transform from A to B, but do this at a different pace) and investigate three questions regarding such a transformation from A to B:

- How does this affect the evolved morphologies?
- How does this affect the evolved behaviors?
- How does this affect the robustness of the evolved robots?

Related Work

Existing work related to morphological evolution of virtual creatures has been addressed in (Sims, 1994), put on a more solid footing later on (Pfeifer and Iida, 2005). The effects of different developmental mechanisms were studied in (Kriegman et al., 2018b), while a method for phenotypic plasticity of morphology and controller was proposed in (Kriegman et al., 2018a). Nevertheless, they did not experiment with different environmental conditions. In (Daudelin et al., 2018) reconfigurable robots were evolved to cope with actual changes in the environmental conditions as they moved about, but no quantification of this effect on the morphological level was provided. In (Auerbach and Bongard, 2014)

it has been demonstrated that increasing the complexity of the environmental conditions results in an increase in the morphological complexity of the creatures. Nevertheless, measuring complexity does not provide clear insights concerning intelligible morphological traits, as for instance the limbs of a robot. Moreover, in (Bongard, 2011a) it has been demonstrated that phylogenetic and ontogenetic morphological (and neurological) changes can not only accelerate the discovery of successful behavior but produce robots more robust to variations in environmental conditions. Finally, in (Bongard, 2011b) it was found that if environmental scaffolding is preceded by morphological scaffolding, significant performance improvement can be achieved.

Robot Framework

Morphology

We are using simulated robots based on RoboGen (Auerbach et al., 2014) whose morphologies (“bodies”) are composed of modules shown in Fig. 1. Any module can be attached to any other module through its attachable slots, except for the sensors, which can not be attached to joints. Our morphologies (Fig. 2) consist of a single layer, i.e., the modules do not allow attachment on the top or bottom slots, only on the lateral ones, but the joints can bend, so the robots can ‘stand’ in a 3D-shape. Each module type is represented by a distinct symbol in the genotype.

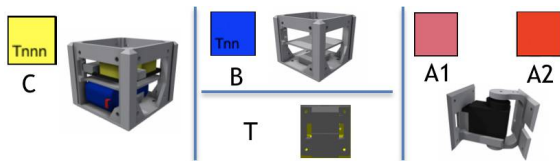


Figure 1: Robot modules: Core-component with controller board (C); Structural brick (B); Active hinges with servo motor joints in the vertical (A1) and horizontal (A2) axes; Touch sensor (T). C and B have attachment slots on their four lateral faces, and A1 and A2 have slots on their two opposite lateral faces; T has a single slot which can be attached to any slot of C or B. The sequence of letters (T or n) in C and B indicate if there is a sensor on the laterals left, front, right and back (for C only), in this order.

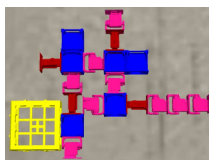


Figure 2: Example of robot in simulation.

Controller

The controller (“brain”) is a hybrid artificial neural network, which we call Recurrent CPG Perceptron (Fig. 3, right). For every joint in the morphology, there exists a corresponding oscillator neuron in the network, whose activation function is calculated through a Sine wave with three parameters: Phase offset, Amplitude, and Period. The oscillators are not interconnected, and every oscillator may or may not possess a direct recurrent connection. Additionally, every sensor is reflected as an input for the network, which might connect to one or more oscillators, having the weights of its connections ranging from -1 to 1 . The CPG (Ijspeert, 2008) generates a constant pattern of movement, even if the robot is not sensing anything, so that the sensors are used either to suppress or to reinforce movements.

Representation and operators

We use an evo-devo style generative encoding to represent the robots. Specifically, our genomes –that encode both morphology and controller– are based on a Lindenmayer-System (L-system) inspired by (Hornby and Pollack, 2001). The grammar of an L-System is defined as a tuple $G = (V, w, P)$, where

- V , the alphabet, is a set of symbols containing replaceable and non-replaceable symbols.
- w , the axiom, is a symbol from which the system starts.
- P is a set of production-rules for the replaceable symbols.

The following didactic example illustrates the process of iterative-rewriting of an L-System. For a given number of iterations, each replaceable symbol is simultaneously replaced by the symbols of its production-rule. Given $w = X$, $V = \{X, Y, Z\}$ and $P = \{X : \{X, Y\}, Y : \{Z\}, Z : \{X, Z\}\}$, the rewriting goes as follows.

Iteration 0: X
 Iteration 1: XY
 Iteration 2: XYZ
 Iteration 3: $XYZXZ$

In our system each genotype is a distinct grammar in the syntax specified by the types of modules we have. The alphabet is formed by symbols denoting the morphological modules and commands to attach them together, as well as commands for defining the structure of the controller. The construction of a phenotype (robot) from a genotype (grammar) is done in two stages. In the first stage (early development), the axiom of the grammar is rewritten into a more complex string of symbols (intermediate phenotype), according to the production-rules of the grammar. (Here we set the number of iterations to 3). In the second stage (late development), this string is decoded into a phenotype. The second stage of this process is illustrated in Figure 3.

The first stage was omitted because it is somewhat extensive, but it follows work flow shown in the example above. During the second stage of constructing a phenotype two positional references are always maintained in it, one for the morphology (pointing to the current module) and one for the controller (pointing to the current sensor and the current oscillator). The application of the commands happens in the current module in the case of the morphology, while for the controller it happens in (or between) the current sensor and the current oscillator. More details about the representation can be found in (Miras et al., 2018c,b).

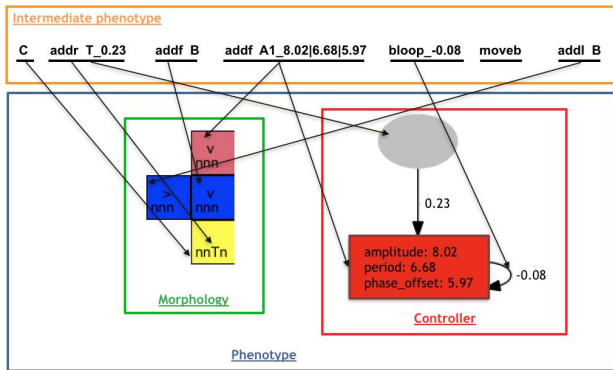


Figure 3: Process of late development: decoding an intermediate phenotype into a (final) phenotype with morphology and controller.

The initialization of a genotype adds, to each production rule, one random (uniformly) symbol of each the following categories, in this order: Controller-moving commands, and Controller-Changing commands, Morphology-mounting commands, Modules, Morphology-moving commands. This can be repeated for r times, being r sampled from a uniform random distribution ranging from 1 to e . This means that each rule can end up with 1 or maximally e sequential groups of five symbols (here e is set to 3). The symbol C is reserved to be used exclusively at the beginning of the production rule C .

The crossovers are performed by taking complete production-rules randomly (uniform) from the parents. Finally, individuals undergo mutation by adding/deleting/swapping one random (uniform) symbol from a random production-rule/position. All symbols have the same chance of being removed or swapped. As for the addition of symbols, all categories have equal chance of being chosen to provide a symbol, and every symbol of the category also has equal chance of being chosen. An exception is always made to C to ensure that a robot has one and only one core-component. This way, the symbol C is added as the first symbol of the C production rule, and can not be added to any other production rules, neither removed or moved from the production rule of C .

Once it is possible that only the rules of one single parent end up being expressed in the final phenotype, and also as it is not rare that one mutation happens for non-expressed genes, both crossover and mutation probabilities were set high, to 80%, aiming to minimize this effect.²

For practical reasons (simulator speed and physical constructability) we limit the the number of modules allowed in a robot to a maximum of 15.

Morphological Descriptors

For quantitatively assessing morphological properties of the robots, we utilized the following set of descriptors:

1. **Size:** Total number of modules in the body.
2. **Number of Joints:** Total number of active joints (motors) in the body.
3. **Number of Limbs:** The number of extremities of a body.
4. **Length of Limbs:** The number of modules with exactly two attachments.
5. **Relative Number of Joints:** The number of active joints relative to a practical limit³.
6. **Relative Number of Limbs:** The number of extremities of a body relative to a practical limit.
7. **Relative Length of Limbs:** The length of limbs relative to a practical limit.
8. **Proportion:** The length-width ratio of the rectangular envelope around the body.

The exact formulas for descriptors 5 to 8 can be found in (Miras et al., 2018c,b). Additionally, a complete search space analysis of the utilized robot framework and its descriptors is available in (Miras et al., 2018c,b,a), demonstrating the capacity of these descriptors to capture relevant robot properties, and proving that this search space allows high levels of diversity.

Behavioral Descriptors

Speed Describes the speed (cm/s) of the robot along the x axis as defined by Eq. 1.

$$s_x = \frac{e_x - b_x}{t} \quad (1)$$

where b_x is x coordinate of the robot's head in the beginning of the simulation, e_x is x coordinate of the robot's head at the end of the simulation, and t is the duration of the simulation.

²This means that around 80% of the offspring will be result of crossovers, and also that around 80% of the offspring will suffer the above explained mutation.

³The practical limits definitions can be found in (Miras et al., 2018c,b).

Balance We use the rotation of the head in the x - y plane to define the balance of the robot. In general, the rotation of an object can be described in the dimensions roll, pitch, and yaw. Thus, we consider the pitch and roll of the robot head, expressed in degrees between 0 and 180 (because we do not care if the rotation is clockwise or anti-clockwise). Perfect Balance belongs to $p = r = 0$, so that the higher the Balance, the less rotated the head is. Formally, Balance is defined by Eq. 2.

$$b = 1 - \frac{r + p}{t * 180 * 2} \quad (2)$$

where $r = \sum_{i=1}^t |r_i|$, representing the roll rotation accumulated over time, $p = \sum_{i=1}^t |p_i|$, representing the pitch rotation accumulated over time, and t is the duration of the simulation.

Experimental Setup

Environments

All environments we use are based on a flat terrain, distinguished by the angle of inclination α running from 0 to 15 with increments of 5 degrees. Thus we have four environments: Plain ($\alpha = 0$), Tilted 5 ($\alpha = 5$), Tilted 10 ($\alpha = 10$), and Tilted 15 ($\alpha = 15$). The tilted environments are more difficult than Plain because the robots must tackle gravity and we use a fitness function based on uphill locomotion. These choices are based on foregoing work that showed that Plain and Tilted 15 are really different in the sense that the robot populations evolved in them exhibit clear differences. Tilted 5 and Tilted 10 are introduced here as intermediary environments to allow a gradual transition between the two extremes, Plain and Tilted 15. Therefore, in the sequel we refer to Tilted 15 simply as Tilted. The environments are shown in Figure 4.

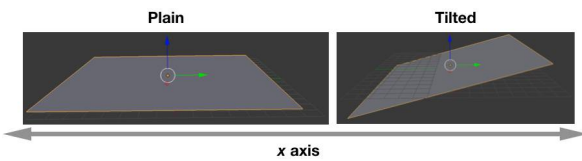


Figure 4: The Plain and Tilted 15 environments.

Environmental scenarios

We carried out four experiments using the same experimental setup, except for the environments in which the robots were evolved.⁴ The four environmental scenarios are depicted in Figure 5 and explained below.

⁴The datasets generated and analyzed in this study are stored at ssh.data.vu.nl, in the `karinemiras-alife2019` directory, available on request. The code can be found at <http://tinyurl.com/yxry52ge>

Plain	Plain			
Tilted	Tilted 15			
Equal Scaffolding	Plain	Tilted 5	Tilted 10	Tilted 15
Incremental Scaffolding	Plain	Tilted 5	Tilted 10	Tilted 15

Figure 5: Environmental scenarios.

- **Plain:** robots evolve in the Plain environment.
- **Tilted:** robots evolve in the Tilted 15 environment.
- **Equal Scaffolding:** robots evolve for an equal number of generations (25) in a sequence of environments Plain, Tilted 5, Tilted 10, and Tilted 15.
- **Incremental Scaffolding:** robots evolve for increasing numbers of generations in a sequence of environments Plain, Tilted 5, Tilted 10, and Tilted 15. The numbers are 10, 20, 30, and 40 respectively. The rationale behind this scenario is giving more time to the more difficult environments.

Evolution

We are using overlapping generations with population size $\mu = 100$. In each generation $\lambda = 50$ offspring are produced by selecting 50 pairs of parents through binary tournaments (with replacement) and creating one child per pair by crossover and mutation. From the resulting set of μ parents plus λ offspring, 100 individuals are selected for the next generation, also using binary tournaments. The evolutionary process is stopped after 100 generations, thus all together we perform 5050 fitness evaluations per run. For each environmental scenario the experiment was repeated 10 times independently.

Fitness function

In all environments the same fitness was utilized. It only concerns the speed of the robots along the x axis to push for uphill movement in the tilted environments. Locomotion along the y axis is ignored and not moving at all is penalized. The fitness function is defined by the following equation

$$f_x = \begin{cases} s_x & \text{if } s_x > 0 \\ \frac{s_x}{10} & \text{if } s_x < 0 \\ -0.1 & \text{if } s_x = 0 \end{cases}$$

where s_x is the speed of the robot as defined by Eq. 1. The duration t of the evaluation periods was set to 50 seconds in all environments.

Results

Behavior

In this section we review the evolved behaviors by considering speed, balance, and the actual gaits the robots exhibited.

Since fast uphill movement is our targeted quality, we start by analyzing the development of speed s_x across the generations for each environmental scenario (Fig. 6, left). The robots achieved a much higher speed in the Plain scenario than in Tilted, confirming the intuition that the Tilted environment constitutes a greater challenge to the robots than the Plain environment. The results for Equal Scaffolding are not significantly different from Tilted, having a lot of variance among the runs. (See Table 2 for the exact p -values.)

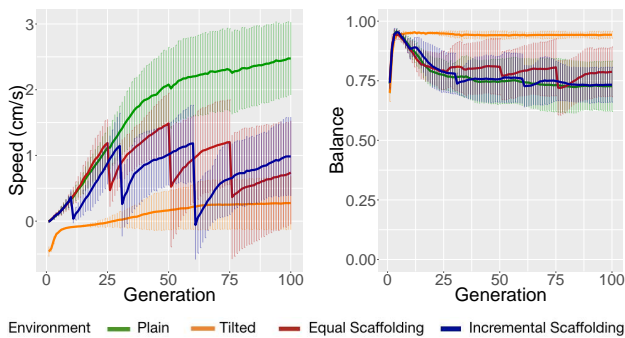


Figure 6: Progression of the mean of the population averaged (mean) over all runs for the behavioral descriptors.

Interestingly, Incremental Scaffolding achieved significantly higher speed than Tilted, proving helpful to succeed in the task, in accordance with our initial hypothesis. Let us note that four of the ten runs of Tilted stagnated in a local optimum, where the robots are small and hardly move. This seems like an avoiding strategy, that prefers reducing movement to risking a fall during the climbing attempt.⁵ Since this never occurred in a scaffolding scenario, we can observe that they help avoid such suboptimal strategies during the search.

Aiming to assess the emergent behavior beyond what is reflected in the fitness function, we measured the balance of the robots. Figure 6 (right) shows that while robots in the Tilted scenario converge to a high balance, in the two scaffolding scenarios they converge to a low balance, which is the same behavior emergent in the Plain scenario. The statistical tests in Figure 6 are shown in Table 2 confirm that these effects are significant. Thus, we found that in both scaffolding scenarios the robots evolved behavior that is similar to what is achieved in Plain and different from that in Tilted. This is in contrast with our initial hypothesis outlined in the Introduction.

As for the evolved gaits of the robots we could distinguish three prominent strategies, rolling, rowing, and standing still. Rolling is characterized by rotating over the roll or pitch dimension with the whole extension of the body. Rowing is characterized by simultaneously boosting with

⁵All runs of Tilted reached this local optimum; six passed it successfully and four got stuck there.

one or more parts of the body, keeping the head somewhat balanced. Standing still refers to robots that almost did not move.⁶ Table 1 shows the counts (out of the 10 independent runs) for each of these gaits in each of the four environmental scenarios. The distribution of these counts shows the same effect as the balance values: most runs of the scaffolding scenarios converged to the predominant gait for Plain (rolling), while all the Tilted runs that avoided the stand still strategy converged to another gait (rowing).

Scenario/Gait	Rolling	Rowing	Still
Plain	9	1	0
Tilted	0	6	4
Equal Scaffolding	7	3	0
Incremental Scaffolding	8	2	0

Table 1: Number of runs in which different gaits emerged within each environmental scenario.

Morphology

The morphological properties of the robots also present the same effect as observed for behaviors. Figure 7 contains the progression of the morphological descriptors for all the environmental scenarios.⁷ The scenario Tilted presents significant difference for several morphological descriptors when compared to all other scenarios. On the other hand, Plain, Equal Scaffolding, and Incremental Scaffolding present (almost) no cases of difference among themselves. These differences, i.e., induced properties on the evolved robots, between Tilted and Plain are evidence of the effects of the environmental conditions. In the case of Tilted, the selection pressure favored robots that are smaller, more proportional, less actuated and with more and shorter limbs in comparison with Plain. However, when introducing the context of a gradual change this pressure disappeared. What we see instead, is the gradualness of the changes causing the predominant robot traits of Plain to also emerge when scaffolding for Tilted. This phenomenon could perhaps be due to the scaffolding scenarios, once starting the evolution in the Plain environment, being driving the search to distinct areas which would be immediately favored when starting evolution directly in the Tilted environment. Statistical tests comparing the curves in Figure 7 are shown in Table 2.

To better illustrate the morphological differences and similarities in the last 10 generations we plotted the density maps for certain pairs of descriptors in Figure 8. These maps provide additional confirmation of the fact that the scaffolding scenarios end up with rather Plain-type populations. In

⁶A video showing some of the emergent robots in each environment can be found in <http://tinyurl.com/y5tdeuh5>

⁷We additionally analyzed other morphological descriptors, but as they presented no significant differences, we did not include them in this paper.

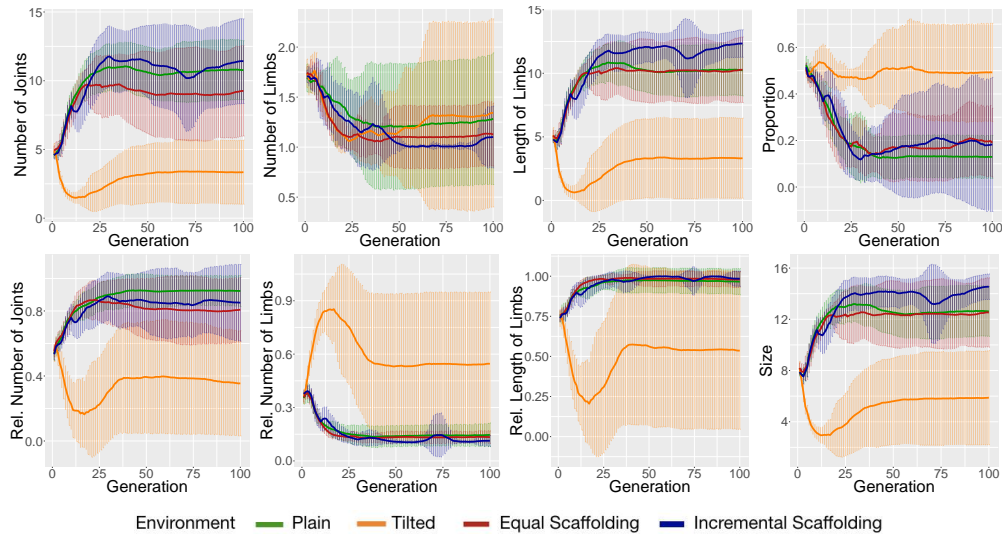


Figure 7: Progression of the mean of the population averaged (mean) over all runs for morphological descriptors.

	P vs T	P vs ES	P vs IS	T vs ES	T vs IS	ES vs IS
	$p <$	$p <$	$p <$	$p <$	$p <$	$p <$
Speed	0	1e-04	0	0.247	0.007	0.63
Balance	0.002	0.25	0.58	5e-04	0	0.32
Number of Joints	2e-04	0.171	0.6	0.001	4e-04	0.15
Rel. Number of Joints	2e-04	0.231	0.903	0.008	0.002	0.405
Number of Limbs	0.73	0.96	0.803	0.654	0.962	0.803
Rel. Number of Limbs	0.002	1	0.16	0.002	4e-04	0.182
Length of Limbs	2e-04	0.91	0.052	8e-04	2e-04	0.06
Rel. Length of Limbs	0.022	0.96	0.803	0.02	0.03	0.803
Proportion	3e-04	0.384	0.382	0.001	0.003	0.161
Size	4e-04	0.97	0.035	0.002	2e-04	0.056

Table 2: Wilcoxon tests (p -values) for differences in the means (averaged over all runs) in the final populations. Significant differences ($p < 0.05$) are highlighted. P = Plain, T = Tilted, ES = Equal Scaffolding ; IS = Incremental Scaffolding.

the meanwhile, the last generations in the Tilted environment are clustered in more distant areas.

Finally, we display the morphologies of the five best individuals of each run of each scenario for visual inspection in Figure 9. Looking at these forms, we can see that “snakes” are the winning morphologies in Plain and they are also very prominent in the scaffolding scenarios. The Tilted world is again the outlier, where only one run ends up with “snakes”, five runs lead to more complex shapes and four with small robots (that do not move).

Robustness

We performed a set of tests to check the robustness of the robots, i.e., their capacity to perform well in different environments. To this end, we looked at the speed of the final populations of each environmental scenario in both the Plain and in the Tilted environment. Figure 10 shows the

box-plots summarizing these results.

As expected, robots evolved in the Plain scenario were much slower when tested in the Tilted environment. Nevertheless, when the robots evolved in the Tilted scenario were tested in the Plain environment, their speed was not different. Additionally, for both scaffolding scenarios, the fitness of the evolved robots tested in the Plain environment was better than or equal to the fitness in the Tilted environment. In summary, for all scenarios robots tested in the Plain environment always achieved higher or at least no different speed from the ones tested in the Tilted environment. This is curious because although the Tilted environment proved more difficult for the robots than the Plain environment, it could be the case that the properties of one difficult environment might not be “compatible” with the properties of an easy one.

As for the comparisons for the scaffolding scenarios we

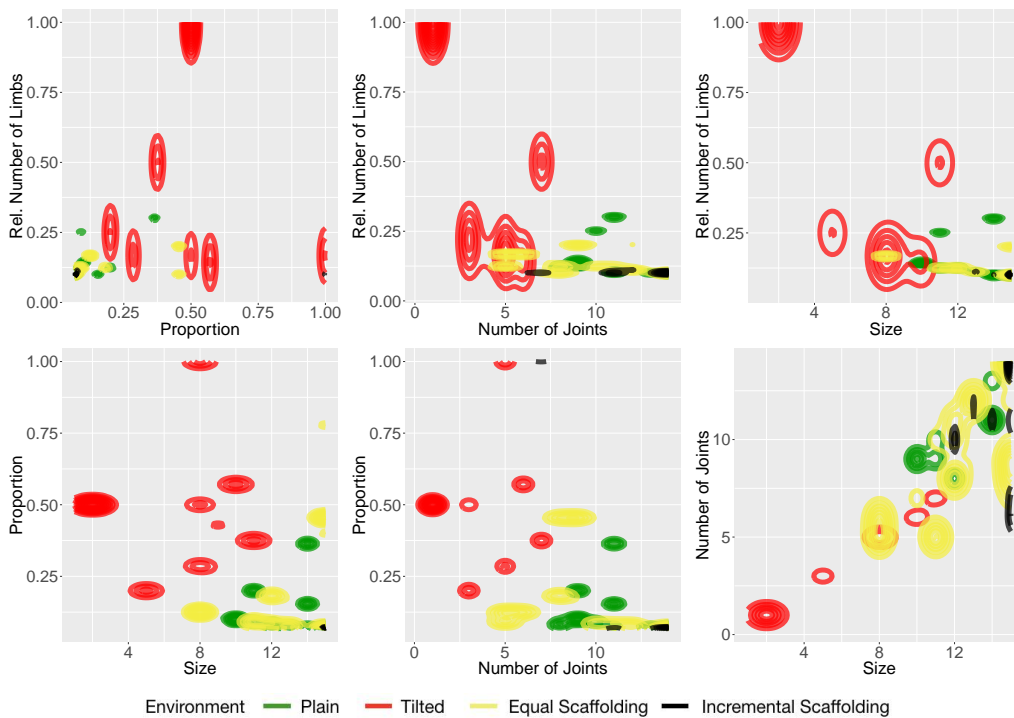


Figure 8: Density maps for pairs of morphological descriptors in the last 10 populations (all runs).

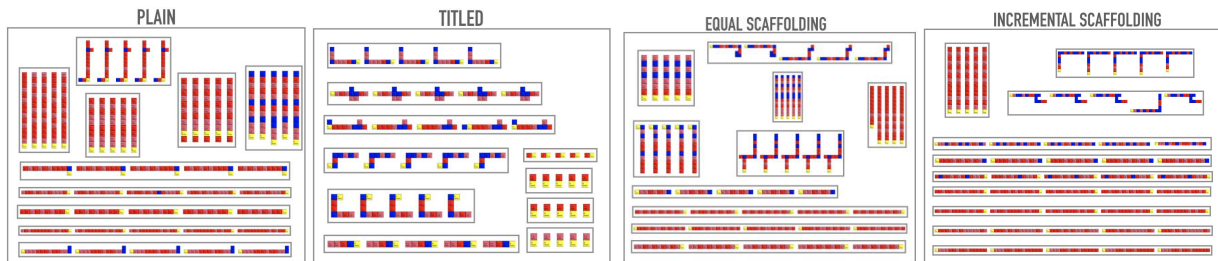


Figure 9: Best robots of the runs evolved in different environmental scenarios. Each sub-frame contains the 5 fastest robots of a run; 10 runs/sub-frames per scenario. Images of big robots were reduced to fit the sub-frames.

can observe the following. For Incremental Scaffolding the differences in speed are not significant between the Plain and the Tilted environments. However, for Equal Scaffolding the evolved robots are significantly slower in Tilted, even though only the first 25 generations have been evolved in the Plain environment and 75 generations were exposed to tilted floors with different angles of inclination. This could be related to the fact that Equal Scaffolding, differently from Incremental Scaffolding, allows robots to evolve for the same amount of generations in all stages, regardless of their difficulty. Thus, once the Tilted environment is more challenging, there is not time enough for the robots to adapt to it as well as to the Plain environment. Finally, for robots tested in the Tilted environment, there was no difference between

having been evolved with Incremental Scaffolding or Equal Scaffolding, and the same can be said about robots tested in the Plain environment.

Conclusions and Future work

We studied the effects of environmental histories by evolving modular robots using different environmental scenarios, a) *flat* terrain; b) *inclined* terrain; c) environmental *scaffolding* from flat to inclined, using an *equal* number of generations per stage; d) environmental *scaffolding* from flat to inclined, using *increasing* numbers of generations per stage. To assess the effects of these scenarios, we utilized a set of morphological and behavioral descriptors. Our results showed that 1) Both scaffolding scenarios helped in the dis-

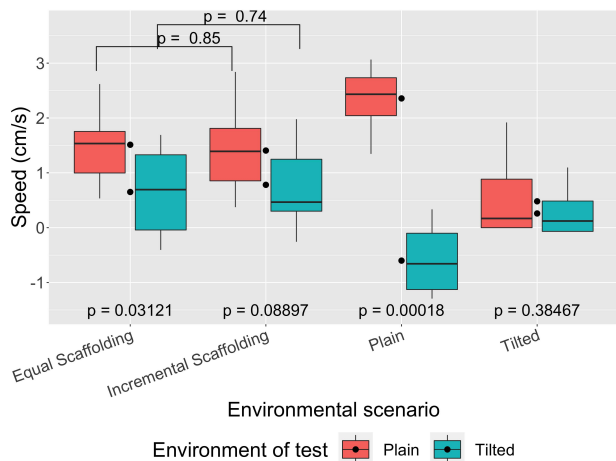


Figure 10: Final evolved populations of each environmental scenario tested in constant Plain and Tilted environments.

covery of better (faster) robots; 2) While the Tilted scenario induced morphological and behavioral properties different from the ones induced by the Plain scenario, both scaffolding scenarios induced the same properties as Plain. 3) From the perspective of robustness, the advantage of using Incremental Scaffolding instead of Equal Scaffolding concerns obtaining a population of robots that performs the task with equivalent quality in both the involved environments. In summary, we can draw two main lessons from this study. First, we observe that populations have a “genetic memory” of properties evolved in environmental conditions from early stages. Second, we note that to evolve robots for difficult conditions it is beneficial to gradually increasing the difficulty.

For future work we propose to verify the effects of inverting the scaffolding to scenarios, starting with the most challenging environment, and ending with the easiest one.

Acknowledgements

This work has received funding from the European Union’s PHOENIX project under grant agreement No 665347.

References

Auerbach, J., Aydin, D., Maesani, A., Kornatowski, P., Cieslewski, T., Heitz, G., Fernando, P., Loshchilov, I., Daler, L., and Floreano, D. (2014). Robogen: Robot generation through artificial evolution. In *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 136–137. The MIT Press.

Auerbach, J. E. and Bongard, J. C. (2014). Environmental influence on the evolution of morphological complexity in machines. *PLoS computational biology*, 10(1).

Bongard, J. (2011a). Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239.

Bongard, J. C. (2011b). Morphological and environmental scaffolding synergize when evolving robot controllers. *Morphological and Environmental Scaffolding Synergize When Evolving Robot Controller*.

Darwin, C. (2004). *On the origin of species, 1859*. Routledge.

Daudelin, J., Jing, G., Tosun, T., Yim, M., Kress-Gazit, H., and Campbell, M. (2018). An integrated system for perception-driven autonomy with modular robots. *Science Robotics*, 3(23):eaat4983.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. (2015). Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI*, 2(4).

Floreano, D., Husbands, P., and Nolfi, S. (2008). Evolutionary robotics. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, volume Part G.61, pages 1423–1451. Springer.

Hornby, G. S. and Pollack, J. B. (2001). Body-brain co-evolution using l-systems as a generative encoding. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 868–875. Morgan Kaufmann Publishers.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653.

Kriegman, S., Cheney, N., and Bongard, J. (2018a). How morphological development can guide evolution. *Scientific reports*, 8(1):13934.

Kriegman, S., Cheney, N., Corucci, F., and Bongard, J. C. (2018b). Interceptive robustness through environment-mediated morphological development. *arXiv preprint arXiv:1804.02257*.

Miras, K. and Eiben, A. E. (2019). Effects of environmental conditions on evolved robot morphologies and behavior. In *GECCO 2019 - Proceedings of the 2019 Genetic and Evolutionary Computation Conference*.

Miras, K., Gansekoele, A., Glette, K., and Eiben, A. E. (2018a). Insights in evolutionary exploration of robot morphology spaces. In *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence*, pages 867 – 874.

Miras, K., Haasdijk, E., Glette, K., and Eiben, A. E. (2018b). Effects of Selection Preferences on Evolved Robot Morphologies and Behaviors. In Ikegami, T., Virgo, N., Witkowski, O., Suzuki, R., Oka, M., and Iizuka, H., editors, *Proceedings of the Artificial Life Conference 2018 (ALIFE 2018)*, Tokyo. MIT Press.

Miras, K., Haasdijk, E., Glette, K., and Eiben, A. E. (2018c). Search space analysis of evolvable robot morphologies. In *Applications of Evolutionary Computation - 21st International Conference, EvoApplications 2018*, volume 10784 of *Lecture Notes in Computer Science*, pages 703–718. Springer.

Pfeifer, R. and Iida, F. (2005). Morphological computation: Connecting body, brain, and environment (:). , 58(2):130–136.

Sapolsky, R. M. (2017). *Behave: The biology of humans at our best and worst*. Penguin.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372.

The Cost of Big Brains in Groups

Danielle Nagar, Alexander Furman, Geoff Nitschke

Computer Science Department
University of Cape Town, South Africa
nrgdan001@myuct.ac.za, frmale003@myuct.ac.za, gnitschke@cs.uct.ac.za

Abstract

The *social brain hypothesis* posits that the evolution of big brains (neural complexity) in groups of social organisms is the evolutionary result of cognitive challenges associated with various complex interactions and the need to process and solve complex social tasks. This study aims to investigate the environmental and evolutionary conditions under which neural complexity evolves without sacrificing collective behavioral efficacy. Using an evolutionary collective robotics system this research evaluates the impact of imposing a fitness cost on evolving increased neural complexity in robot groups that must operate (accomplish cooperative tasks) in environments of varying complexity. Results indicate that for all environments tested, imposing a cost on neural complexity induces the evolution of smaller neural controllers that are comparably effective to more complex controllers.

Introduction

The fitness costs of evolving neural complexity (neural tissue) needed for socially adaptive behaviour are critically important to brain evolution (Azevedo and Houzel, 2012). Various neuroscience studies have demonstrated that increased neural complexity is metabolically expensive (Laughlin et al., 1998), though there are conflicting hypotheses about how the increased fitness costs of larger brain sizes (increased neural complexity) is compensated for in evolution (Armstrong, 1983), (Isler and van Schaik, 2009). Also, the environmental and evolutionary conditions driving the evolution of such complexity in the first place remains an open question (Fisher, 1930).

In some species of organisms it is hypothesized that evolved brain size (neural complexity) is correlated to the size, structure and complexity of social groups formed by such species (Dunbar and Shultz, 2007). The *social brain hypothesis* posits that the evolution of such neural complexity is a result of the cognitive challenges associated with varied and complex interactions and the need to process complex social information (Dunbar, 2009).

Studies of various social organisms including ant colonies (Kamhi et al., 2016), have supported the social brain hy-

pothesis via demonstrating that socially complex behavior such as division of labor and cooperation (collective intelligence) are likely driving forces of brain complexity evolution. However, there is also contradictory evidence in such studies as related work on other ant species (Feinerman and Traniello, 2016) elucidated the evolution of smaller brains (lower neural complexity) in groups of workers that were still able to collectively perform complex collective behaviors that supported and benefited the colony. In both cases, the exact impact of the environment and complexity of cooperative tasks (supporting group survival), on the evolution individual neural complexity and thus the group's social complexity, remains little investigated and unclear in the context of natural and artificial life (Yaeger, 2009).

This research takes inspiration from such evolutionary biology studies, and uses collective *evolutionary robotics* (Doncieux et al., 2015) as an experimental platform to test the impact of varying environment complexity (collective behavior task difficulty) on the evolution of neural complexity. This study tests the *social brain hypothesis* using robot groups, where behaviors are specified by evolved neural controllers and robots interact to cooperatively solve *collective gathering tasks*. Thus, we evaluate the impact of imposing fitness costs on evolving neural complexity in robot groups that must solve increasingly difficult collective gathering tasks, where task difficulty is the degree of cooperation needed for task accomplishment.

In this study, the cost of evolving increased robot neural controller complexity is tantamount to metabolic energy costs associated with increased brain sizes in nature (Armstrong, 1983; Laughlin et al., 1998; Isler and van Schaik, 2009). This study's core motivation is the general lack of understanding (across various fields including evolutionary biology and robotics) of how environment driven necessity for social complexity (for example, emergent collective behaviors and social structure in groups) impacts brain size and structure (Farris, 2016).

Various approaches for evolving neural complexity have been studied in related research topics such as *computational ecologies*, where for example, Williams and Yaeger (2017) demonstrated complexity evolution without explicit fitness costs. Fitness cost impact on complexity evolution has also been demonstrated in *simulated sensor systems* (Seth and Edelman, 2004), and with *neural modularity* (Clune et al., 2013; Lowell and Pollack, 2014). However, there are few evolutionary collective robotics studies that investigate the impact of environment complexity on evolving controllers given a complexity cost (Doncieux et al., 2015). This is significant as evolutionary collective robotics (Bredeche et al., 2018) allow experimenters to investigate competing hypotheses pertinent to, but not readily testable in natural social systems. That is, the social brain hypothesis is more suitably evaluated in embodied cognition systems such as evolutionary robotics given that such systems can readily implement controlled and testable *distributed* and *embodied* theories of cognition (Barrett et al., 2007).

This study thus aims to elucidate the advantages and disadvantages of imposing a fitness cost on evolving controller complexity (neural connectivity) in an evolutionary collective robotics system (Doncieux et al., 2015). Neural complexity is defined by evolved controller topology (connectivity between sensory, hidden and output nodes) and evaluation was evolved collective gathering behavior task performance. Given this and the social brain hypothesis, we formulated the following research objective.

To evaluate the impact of fitness costs on evolving neural controller complexity given increasing collective behavior task difficulty in collective robotics. We thus aim to ascertain if the social brain hypothesis holds for an evolutionary collective robotics system that must operate and accomplish cooperative tasks in environments of varying complexity.

Methods

This study evaluated *NEAT-M* (Hewland and Nitschke, 2015) controller-morphology neuro-evolution versus the *NEAT-M-MODS* multi-objective extension. Both methods co-adapted robot *Artificial Neural Network* (ANN) controllers and sensory-morphologies for given tasks. Groups were homogenous as the same behavior-morphology adaptations were applied to all robots in a group. Behavior-morphology evolution with *NEAT-M-MODS* included a *neural complexity cost* (section: *Neural Complexity*), imposed during neuro-evolution. This was represented as the minimization of neural complexity, in company with the maximization of collective gathering task performance, as part of multi-objective optimization. These objectives thus encouraged the evolution of minimally complex and behaviorally effective neural controllers. *NEAT-M* was comparatively evaluated, where it only maximized collective gather-

ing task performance, in order to ascertain the impact of a neural complexity cost during evolution.

NEAT-M-MODS: Overview

Neuro-Evolution for Augmenting Topologies (*NEAT-M-MODS*) is a multi-objective extension of *NEAT-M* (Hewland and Nitschke, 2015) and *NEAT-MODS* (Abramovich and Moshaiiov, 2016). *NEAT-M* evolves a direct encodings of both robot ANN controllers and morphologies (ANN connections to sensory input nodes constituting a robot's sensory configuration). *NEAT-M-MODS* supersedes the core functionality of *NEAT-M* (Hewland and Nitschke, 2015) via including an NSGA-II based *Multi-Objective Evolutionary Algorithm* (Abramovich and Moshaiiov, 2016), that uses multiple objectives to direct the evolutionary process of *NEAT* (Stanley and Miikkulainen, 2002).

NEAT-M-MODS initializes a genotype (controller-sensor) population, computes each genotype's score vector (multi-objective fitness), speciates the population and computes a *rank* for each genotype based on *non-dominated sorting* and *crowding distance comparisons* (Doncieux and Mouret, 2014). This process, evolutionary operators and parameters are detailed in Furman et al. (2019) and *NEAT-MODS* is described in Abramovich and Moshaiiov (2016).

Neural Controller-Sensor Evolution

For both neuro-evolution methods (*NEAT-M*, *NEAT-M-MODS*), robots began with a minimal sensory configuration of five sensors (one of each type), each sensor corresponding to a controller input node. Input nodes were fully connected to two motor output nodes (figure 1, left). Connections were randomly initialized and without any hidden layers and controllers subject to *complexification* during neuro-evolution. Controllers used sigmoidal (Hertz et al., 1991) hidden and output nodes and all sensory inputs were normalized to the range: [0.0, 1.0].

Figure 1 (center-left) presents the initial robot controller-sensory configuration used as an evolutionary starting point for both *NEAT-M* and *NEAT-M-MODS*. This initial controller-sensor configuration (motor outputs were fixed during evolution) was selected so as robots performed some useful behaviors at the start of the evolutionary process. The possible sensor types were: (1) *Ultrasonic*, (2) *Infrared Proximity*, (3) *Color*, (4) *Low Resolution Camera*, and (5) *Gathering Zone Detector* (table 1). These sensors were selected as they are typically available for the Khepera III mobile robot (Lambercy and Tharin, 2013). Parameters for all sensor types were perturbable by mutation operators that add and remove sensors (of a given type), as well as modify, add and remove ANN connection weight values, add and remove weight connections to sensors, and change sensor positions and orientations (on the robot's periphery).

Table 1: Experiment and Simulation Parameters

Block size	Small	0.01 x 0.01
	Medium	0.015 x 0.015
	Large	0.02 x 0.02
Sensor types : Range / FOV	Ultrasonic	(0.0, 1.0] / (0.0, π)
	Infrared Proximity	(0.0, 0.4] / ($\pi/6$, $5\pi/6$)
	Color	(0.0, 0.4] / ($\pi/6$, $5\pi/6$)
	Low Res Camera	(0.0, 0.8] / ($\pi/9$, $8\pi/9$)
	Gathering Zone Detection	Bottom facing
Sensor bearing range	$[-\pi, \pi]$ Radians	
Sensor orientation range	$[-\pi/2, \pi/2]$ Radians	
Robot <i>lifetime</i> (Time-steps per simulation task trial)	10000	
Robot group size	20	
Robot size (Diameter) / Gripping distance / Speed (per time step)	0.004 / 0.002 / 0.013	
Initial robot / block positions	Random (Outside gathering zone)	
Environment width x height / Gathering zone size	1.0 x 1.0 / 0.5 x 0.2	
Minimum / Maximum number of sensors	1 / 10	
Task environments (Blocks: small, medium, large)	Simple	10, 5, 0
	Medium	5, 5, 5
	Difficult	0, 5, 10
Cooperation needed for block pushing	Small	1 Robot
	Medium	2 Robots
	Large	3 Robots

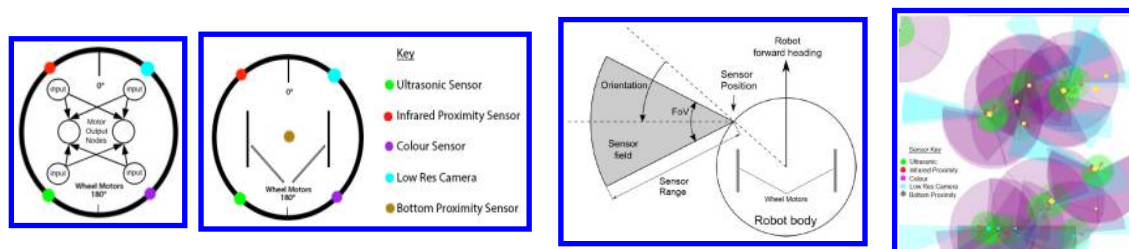


Figure 1: **Left:** Initial robot controller connecting 5 sensors to 2 actuators. **Center-Left:** Robots were initialized with one ultrasonic, infrared proximity, color, gathering zone detector (bottom proximity) sensor and one low-resolution camera. Wheel motors were fixed throughout controller evolution. **Center-Right:** Example robot with one sensor. Position determines sensor location on the robot’s chassis with respect to the robot’s heading. Orientation is then sensor direction with respect to this position. By default, a robot’s heading is forward facing (parallel to its wheels). **Right:** Environment with 20 robots and distributed blocks. The *gathering zone* containing gathered blocks (blue squares) is highlighted at the bottom. Sensory parameters (type, position, orientation, field of view and range) are highlighted as shaded semi-circles.

Mutation operators are presented in Furman et al. (2019). The parameter-set for each sensory input node is: *Sensor Type*, *Field of View* (FOV), *Range*, *Position*, and *Orientation*. Figure 1 (center), presents an example robot with one sensor and an illustration of sensor parameters.

Neuro-evolution was driven by genetic, that is, *crossover* and *mutation* operators. These operators adapted ANN connection weights and added or removed hidden or input nodes. This adapted the number of sensors or otherwise perturbed sensor parameters. At each generation (of NEAT-M and NEAT-M-MODS), either crossover or mutation operators were applied with a given degree of probability. The crossover and mutation operators are described in previous work (Hewland and Nitschke, 2015; Furman et al., 2019). If a new sensor was added (*add sensor* operator) it was placed at a given minimum position distance between two randomly selected sensors already on the robot’s chassis. If there was only one sensor currently on the robot’s body the new sensor was placed randomly to the left or right of this one sensor. The same procedure was followed for the

remove sensor operator, where at least one sensor had to be positioned on a robot’s chassis.

Two wheel motors controlled a robot’s heading at constant speed. Movement was calculated in terms of real valued vectors (dx and dy). Wheel motors (figure 1, center-left, center-right) were explicitly activated by the ANN and a robot’s heading was determined by normalizing and scaling output values by the maximum movement distance for one simulation time-step (Hewland and Nitschke, 2015).

Neural Complexity Definition

Neural complexity¹ is defined as the number of connections n ($n \in [0, 120]$) in an evolved neural controller (at generation g) and thus includes all connection weights linking sensory input nodes to hidden and output nodes. This neural complexity definition was adapted from related work (Nitschke and Didi, 2017) and selected for its simplicity and accounting of sensory (morphology) complexity with

¹Neural simplicity is synonymous in this article’s discussion.

respect to the neural controller. Thus, the more complex a robot’s sensory morphology, the higher its controller’s neural complexity will be. That is, this definition assumes sensory input nodes are connected to hidden or output nodes.

Collective Gathering Task

Collective gathering requires robots to locate distributed resources (blocks) in a bounded environment and transport them, via cooperative pushing, to a *gathering zone* (Nitschke et al., 2012). This task was selected given that relatively sophisticated collective behaviors (enabled by suitably complex controller-morphology couplings of interacting individuals) are required to solve such cooperative tasks (Nitschke, 2005). Also, collective gathering is an established collective evolutionary robotics benchmark task and is thus a suitable experimental platform for evaluating neural controller evolution (Doncieux et al., 2015).

Cooperation was defined as the number of robots required to push given block types and *task difficulty* was a function of the *number of blocks*, *block types* and *cooperation* needed. Block types were *small*, *medium*, or *large*, to be pushed by at least one, two and three robots, respectively (table 1). Task difficulty was calibrated via initializing environments (simple, medium, difficult) with varying combinations of block types (table 1). For example, the *simple* environment contains 10 small and 5 medium sized blocks, so robots could work concurrently with minimal cooperation to move all blocks into the gathering zone. Collective gathering task performance was the total number of blocks pushed into the gathering zone during the robots’ *lifetime* (table 1).

Experiments

Experiments measured the impact of a neural complexity (fitness) cost versus no such complexity cost imposed during controller-sensor (morphology) evolution in robot groups tasked with solving collective gathering tasks of increasing difficulty. NEAT-M-MODS used multi-objective controller evolution (task performance maximization and complexity minimization), and NEAT-M used single objective (task performance) optimization. The experimental platform was a collective robotics simulator (Hewland and Nitschke, 2015) implementing the collective gathering task (figure 1, right). Robots emulated the Khepera III (Lambercy and Tharin, 2013), with co-adapting controllers and sensor configurations². Experiments ran simulations of 20 robots in bounded two dimensional continuous environments with distributions of *small*, *medium*, and *large* blocks (table 1).

Blocks were randomly distributed throughout an environment, excluding the *gathering zone*. Block type

²Simulator & NEAT-M & NEAT-M-MODS source-code is online: <https://github.com/costcomplex/anonymous>

distributions given in table 1 correspond to increasing environment complexity (*simple*, *medium*, *difficult*) designed to test the impact of environment complexity on controller evolution with and without a neural complexity cost. Robots were initially randomly placed in the gathering zone.

To test this study’s research objective we designed two experiment sets. Experiment set 1 evaluated the impact of a neural complexity cost on controller evolution via evaluating NEAT-M-MODS evolved groups in all environments (table 1). Comparatively, experiment set 2 evaluated controller evolution without a neural complexity cost. That is, NEAT-M evolved groups were evaluated in the same environments. Only homogenous teams were tested, meaning that at each NEAT-M and NEAT-M-MODS generation, selected controller-sensor adaptations were copied 20 times (representing group sizes of 20 robots).

Fitness Function

In experiment set 1, task performance was maximized and neural complexity minimized. This second objective encouraged lower neural complexity, thereby imposing a *fitness cost* on controller complexity. In experiment set 2, only task performance was maximized. Task performance was the average *value of blocks* pushed into the gathering zone over five robot *lifetimes* comprising each generation.

We defined v_c as total value of resources (blocks) in the gathering zone, v_t as total value of all resources in the environment, s_e as the number of simulation time-steps in the robots’ lifetime and s_t as number of trial evaluations per genotype (controller-sensor configuration). As such, task performance T was maximised using equation 1:

$$T = 100 \times \frac{v_c}{v_t} + 20 \times \left(1.0 - \frac{s_e}{s_t}\right) \quad (1)$$

In equation 1, 100 was the maximum number of blocks that could be gathered during an experiment run, and 20 was an experimentally determined weighting (boosting fitness for efficient individual and cooperative gatherers).

Each experiment applied NEAT-M or NEAT-M-MODS to evolve collective gathering behavior for 200 generations. A generation comprised five robot lifetimes, where each lifetime was 10000 simulation iterations. Each lifetime was a simulated collective gathering task scenario that tested different robot starting positions, orientations, and block locations in either a *simple*, *medium* or *difficult* environment (table 1). Average collective gathering task performance was calculated at each run’s end and averaged over 20 runs. All simulation and neuro-evolution parameters (table 1) were experimentally determined, where those not reported here used the same settings as in previous work (Hewland and Nitschke, 2015; Abramovich and Moshaiov, 2016).

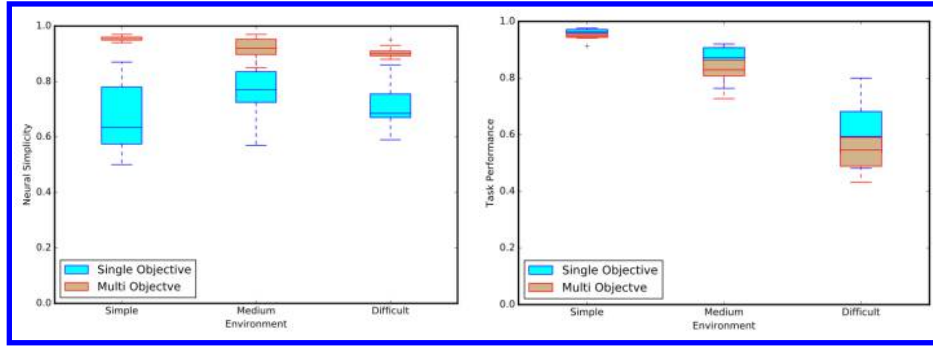


Figure 2: **Left:** Average maximum *neural simplicity* for *Single Objective* (SO): NEAT-M, *Multi Objective* (MO): NEAT-M-MODS (knee-points) for *simple*, *medium* and *difficult* environments, respectively. *Neural simplicity* ~ 1.0 indicates low evolved neural complexity. **Right:** Average maximum task performance of SO: NEAT-M versus MO: NEAT-M-MODS evolved groups.

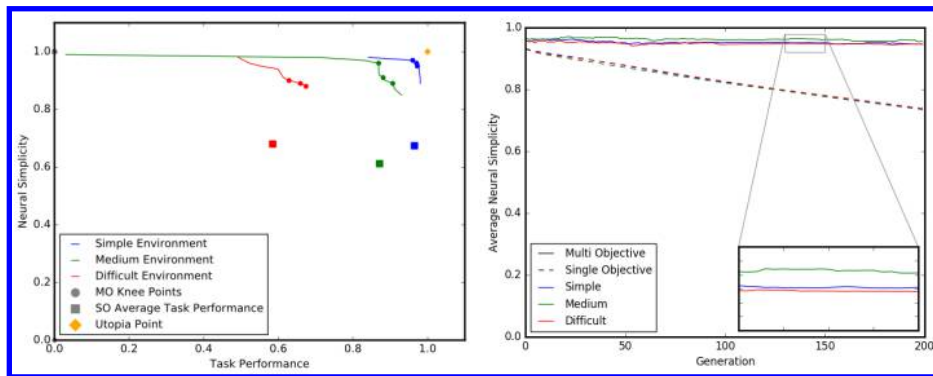


Figure 3: **Left:** Average *Multi-Objective* (MO: NEAT-M-MODS) Pareto front and *Single Objective* (SO: NEAT-M) scores. SO points are maximum task performance and corresponding neural simplicity scores. MO points are the 3 best knee-points. Averages were computed over all runs. **Right:** Progression of average *neural simplicity* for MO and SO over neuro-evolution.

Results & Discussion

Experiments evaluated the impact of a *fitness cost* on neural complexity versus *no cost* during neural network controller evolution in robot groups. Evolved robots were evaluated in terms of collective gathering task performance for increasing difficult environments: *simple*, *medium* and *difficult*. To enable analysis of the neural versus environment complexity trade-off, average neural complexity (and co-adapted sensory-morphology complexity) of the fittest controllers evolved in each environment was also computed. Figure 2 presents average *neural complexity* and *task performance* results. Figure 3 presents *Pareto-front* and evolutionary progression of neural complexity. Results compare these metrics for group behavior evolved *with* (NEAT-M-MODS) and *without* (NEAT-M) a neural complexity cost, across increasingly difficult tasks. Averages were calculated for each environment and over 20 runs for each method.

Figure 2 (left) presents the average *neural complexity*³ of

³Note that the term *simplicity* is used in figure 2 for clarity and consistency with related work (Auerbach and Bongard, 2014).

the fittest controllers evolved by NEAT-M and NEAT-M-MODS in each environment. Neural controller evolution in NEAT-M had a *Single Objective* (SO) of maximizing task performance (fitness), whereas, NEAT-M-MODS had the *Multiple Objectives* (MO) of maximizing task performance and minimizing neural complexity. A neural complexity value of 1.0 indicates the simplest possible controller (one sensor connected to motor outputs) and 0.0 indicates the most complex controller (10 sensors connected to 10 hidden nodes⁴ and outputs, table 1). Figure 2 (right) presents average maximum task performance of NEAT-M (SO) versus NEAT-M-MODS (MO) groups evolved in each environment. Calculations for the latter used the three knee-points with the *highest-task performance* and *lowest neural-simplicity* on the *Pareto front* (figure 3, left).

Figure 3 (left) presents the best three *knee-points* for each *Pareto front*. A knee-point was defined as that yielding the highest value for both objectives, where such values

⁴Fixed topology parameter tuning experiments indicated negligible fitness increases in all environments for > 10 hidden nodes.

were closest to the *utopia* point for the simplest and most effective controllers. For comparison, the best SO points (average maximum task performance and matching neural complexity) of NEAT-M evolved groups are also presented. Figure 3 (right) presents the evolutionary progressions of *neural simplicity* for NEAT-M-MODS and NEAT-M groups evolved in each environment. Values close to 1.0 and 0.0 indicate *low* and *high* neural complexity, respectively.

Figure 3 (right) illustrates that groups evolved *with* a neural complexity cost (NEAT-M-MODS) in all environments were consistently comparable, in terms of neural simplicity, throughout neuro-evolution. For clarity, exemplar generations 125 to 150 in figure 3 (right) are enlarged to highlight that the *medium environment* encouraged the evolution of simpler neural controllers (though this difference was not statistically significant). Whereas, groups evolved *without* a neural complexity cost became increasingly complex over the course of neuro-evolution, for all environments. This is supported by statistically significant neural simplicity differences between groups evolved (in all environments) *with* and *without* neural complexity costs (table 2).

Statistical comparisons used *Shapiro-Wilk* (to confirm data normality) and independent *two-tailed t-tests* (Flannery et al., 1986) to test for significant differences ($p < 0.05$) in average task performance and neural complexity (simplicity). Statistical tests were run between respective results of NEAT-M and NEAT-M-MODS evolved groups. The latter used averages calculated from the three *best knee-points* on each Pareto front (figure 3, left).

Statistical comparisons (table 2) of *average neural simplicity*⁵ indicated that for all environments, groups evolved *with a neural complexity cost* comprised significantly simpler controllers compared to those evolved *without a complexity cost*. Thus, neuro-evolution with a complexity cost given increasing environment complexity consistently resulted in low selection pressure for complex controller evolution. *Environment complexity* (task difficulty) was a function of the number of blocks and block types, where some degree of cooperation was needed for optimal gathering in all environments (section: *Collective Gathering Task*). Figure 3 (right: evolving average neural simplicity) and figure 2 (left: average maximum evolved neural complexity) further support this result, indicating that neural complexity costs consistently enable the evolution of significantly simpler controllers (versus evolution with no complexity cost). This result is also supported by related work (Lowell and Pollack, 2014) similarly finding that fitness costs on connection weights in NEAT evolved networks produced smaller networks that were just as effective

⁵*Simplicity* instead of *complexity* is used in this discussion.

as the best evolved networks without such a complexity cost.

Also, figures 2 and 3 indicate groups evolved without a neural complexity cost resulted in higher neural complexity which in turn enabled higher group task performance (for all environments). However, these task performance differences were negligible for the *simple* environment (SO: 0.96 versus MO: 0.95 in table 2) and minimal for the *medium* (SO: 0.86 versus MO: 0.78 in table 2) and *difficult* (SO: 0.61 versus MO: 0.49 in table 2) environments. Thus, in the *medium* and *difficult* environments, there was an average task performance difference of only 10% between groups evolved with and without a neural complexity cost.

The complexity-fitness trade-off for these minimal task performance differences were significantly less complex neural controllers. Groups evolved *with* a complexity cost in the simple, medium and difficult environments, were 27%, 15% and 22% simpler (table 2, right), respectively. The complexity cost also enabled the evolution of significantly simpler (independent two-tailed t-test, $p < 0.05$) sensor-morphologies coupled with evolved controllers. Sensor-morphologies of groups evolved with the complexity cost were on average, 32%, 35% and 27% simpler, in the *simple*, *medium* and *difficult* environments (respectively). This was compared to the sensory-morphologies of groups evolved without a complexity cost in the same environments.

Sensor-morphology complexity (*simplicity*) was a function of sensors (number and type) and sensor parameters coupled to evolved neural controllers (Furman et al., 2019). For consistency, values were calculated according to *sensory-system simplicity*, where a value of 1.0 indicated one sensor with minimal range and *Field of View* (FOV) and 0.0 indicated a controller with 10 sensors (maximum number) of all types with maximal range and FOV (table 1).

Also, observing the *best knee-point* (figure 3, left) controllers evolved with multi-objective neuro-evolution (incorporating the complexity cost), one notes such controllers had comparable task performances (no statistical difference) in all environments. Thus, even though having no neural complexity cost enabled the evolution of increasingly complex neural controllers⁶ (and coupled sensor-morphologies), the added neural and sensor-morphology complexity was largely redundant and unnecessary. This was especially the case when considering the *best knee-point* controllers that evolved to be simple yet effective.

The collective behavior effectiveness of such simple controllers is further theorized to be a result of the nature of task (environment) complexity. Consider that optimal

⁶Controller topologies evolved with and without complexity costs are online: <https://github.com/costofcomplexity/ALIFE2019>

Table 2: **Left:** Statistical comparisons for task performance (fitness) and neural complexity of fittest controllers evolved by NEAT-M (SO: Highest task performance) versus NEAT-M-MODS (MO: 3 Knee-points on Pareto-front, figure 3). $>$: Greater than with statistical significance. **Right:** Average task performance and neural simplicity values for groups evolved *without* (SO) and *with* (MO) a neural complexity cost. Neural complexity (simplicity) is defined in section: *Neural Complexity Definition*.

Environment	Task Performance	Neural Simplicity
Simple	SO $>$ MO	MO $>$ SO
Medium	SO $>$ MO	MO $>$ SO
Difficult	SO $>$ MO	MO $>$ SO

Environment	Task Performance	Neural Simplicity
Simple	0.96 (SO) 0.95 (MO)	0.67 (SO) 0.94 (MO)
Medium	0.86 (SO) 0.78 (MO)	0.77 (SO) 0.92 (MO)
Difficult	0.61 (SO) 0.49(MO)	0.71 (SO) 0.93 (MO)

task accomplishment required the group to gather all blocks distributed throughout the environment into a *gathering zone*, within the group’s *lifetime*. However, all environments required some cooperation to achieve an optimal or near optimal task performance. For example, the simple environment contained five medium-sized blocks requiring two robots to cooperatively push. The medium environment also contained five large-sized blocks requiring three robots to cooperatively push and the difficult environment included only medium-sized and large-sized blocks. In all environments the efficacy of an emergent cooperative problem-solving behavior was determined by the neural and morphological (sensor) complexity of evolved controllers.

As demonstrated in related work (Waibel et al., 2009; Nitschke et al., 2012; Duarte et al., 2016), we posit that the emergence of effective collective behaviors was enabled by evolved interactions of relatively simple individual controllers. Furthermore, other evolutionary robotics studies have demonstrated evolving increased neural complexity often yields negligible benefits as collective behavior task complexity increases (Nitschke and Didi, 2017).

The overall implication of this study’s results is two-fold. First, it indicates that neuro-evolution *with* a neural complexity cost facilitates efficient neural controllers comprising minimal connectivity and simple coupled sensor-morphologies. These controllers were consistently effective for increasing environment complexity (addressing the research objective). Second, results indicated neuro-evolution *without* a neural complexity cost supports the social brain hypothesis in the context of groups surviving in environments of varying complexity. That is, for all environments, groups evolved without a neural complexity cost evolved both significantly larger neural controllers and coupled sensor-morphologies, compared to those evolved with a neural complexity cost. However, the pertinence of the social brain hypothesis was limited in that these more complex neural structures (and sensor-morphology couplings) did not yield clear benefits across all environments.

Specifically, there was negligible group task performance differences in the simple environment and minimal differ-

ences ($\sim 10\%$) in the medium and difficult environments. As presented in figure 3 (left), the best groups (knee-point controllers on the Pareto front) evolved with a complexity cost yielded comparable task performances to groups evolved without a complexity cost, though did so with less neural complexity and simpler sensor-morphologies. Hence, the neural complexity cost was able to suitably complement the *complexification* and *speciation* mechanisms of NEAT-M (Hewland and Nitschke, 2015) via enabling the evolution of simple yet effective controllers. However, without such a complexity cost, NEAT-M complexification ensured the evolution of increasingly complex controllers and coupled sensor-morphologies, where the speciation mechanism was unable to mitigate this evolution of neural complexity.

This study’s contribution was two-fold. First, using evolutionary collective robotics to test the impact of a complexity cost, the study supported related work demonstrating that greater evolved social complexity correlates with smaller brains coupled to less complex sensory-systems (Gronenberg and Riveros, 2009). In this study, social complexity was the efficacy of evolved collective behavior across increasingly difficult task environments. Second, results lent empirical support to the use of distributed and embodied cognition systems (Barrett et al., 2007), in this case for the purpose of elucidating the impact of a complexity cost on evolving neural and morphological complexity in social systems.

Conclusion

This study investigated how imposing fitness costs on evolving neural controller complexity impacted evolving collective behaviors in an evolutionary collective robotics system. Experiments evaluated the impact of evolving neural controllers, *with* and *without* a neural complexity cost, on robot group task performance in increasingly complex environments. In this study environment complexity was equated with collective behavior task difficulty. Results indicated that a neural complexity cost enabled the evolution of simpler controllers, where the best of such simple controllers produced collective behaviors comparable to that of more complex controllers evolved without any complexity cost. This result held for collective behavior evolution in task environments of increasingly complexity.

References

- Abramovich, O. and Moshaiov, A. (2016). Multi-objective topology and weight evolution of neuro-controllers. In *Proceedings of the Congress on Evolutionary Computation*, pages 670–677. IEEE Press.
- Armstrong, E. (1983). Relative brain size and metabolism in mammals. *Science*, 220(1):1302–1304.
- Auerbach, J. and Bongard, J. (2014). Environmental influence on the evolution of morphological complexity in machines. *PLoS Computational Biology*, 10(1):e1003399. doi:10.1371/journal.pcbi.1003399.
- Azevedoand, K. and Houzel, S. (2012). Metabolic constraining imposes tradeoff between body size and number of brain neurons in human evolution. *Proceedings of the National Academy of Sciences*, 109(18):571–576.
- Barrett, L., Henzi, P., and Rendall, D. (2007). Social brains, simple minds: Does social complexity really require cognitive complexity? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):561–575.
- Bredeche, N., Haasdijk, E., and Prieto, A. (2018). Embodied evolution in collective robotics: A review. *Frontiers in Robotics and AI*, 5(12):10.3389/frobt.2018.00012.
- Clune, J., Mouret, J.-B., and Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society B*, 280:20122863.
- Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. (2015). Evolutionary robotics: what, why, and where to. *Frontiers of Robotics and AI*, doi: 10.3389/frobt.2015.00004.
- Doncieux, S. and Mouret, J.-B. (2014). Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93.
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., and Oliveira, S. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE*, 11(3):e0151834. doi:10.1371/journal.pone.0151834.
- Dunbar, R. (2009). The social brain hypothesis and its implications for social evolution. *Annals of Human Biology*, 36(5):562–572.
- Dunbar, R. and Shultz, S. (2007). Evolution in the social brain. *Science*, 317(1):1344–1347.
- Farris, S. (2016). Insect societies and the social brain. *Current Opinion in Insect Science*, 15(1):1–8.
- Feinerman, O. and Traniello, J. (2016). Social complexity, diet, and brain evolution: modeling the effects of colony size, worker size, brain size, and foraging behavior on colony fitness in ants. *Behavioral Ecology & Sociobiology*, 70(7):1063–1074.
- Fisher, R. (1930). *The Genetical Theory of Natural Selection*. Oxford University Press, Oxford, UK.
- Flannery, B., Teukolsky, S., and Vetterling, W. (1986). *Numerical Recipes*. Cambridge University Press, Cambridge, UK.
- Furman, A., Nagar, D., and Nitschke, G. (2019). The cost of complexity in robot bodies. In *Proceedings of the Congress on Evolutionary Computation*, Wellington, New Zealand. IEEE Press.
- Gronenberg, W. and Riveros, A. (2009). Social brains and behavior past and present. In Gadau, J. and Fewell, J., editors, *Organization of insect societies: from genome to socio-complexity*, pages 377–401. Harvard University Press, Cambridge, USA.
- Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley.
- Hewland, J. and Nitschke, G. (2015). The benefits of adaptive behavior and morphology for cooperation. In *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pages 1047–1054, Cape Town, South Africa. IEEE Press.
- Isler, K. and van Schaik, C. (2009). The expensive brain: a framework for explaining evolutionary changes in brain size. *Journal of Human Evolution*, 57(1):392–400.
- Kamhi, J., Gronenberg, W., Robson, S., and Traniello, J. (2016). Social complexity influences brain investment and neural operation costs in ants. *Proceedings of the Royal Society B*, 283(1):20161949.
- Lambercy, F. and Tharin, J. (2013). *Khepera III User Manual: Version 3.5*. K-Team Corporation, Lausanne, Switzerland.
- Laughlin, S., vanSteveninck, R., and Anderson, J. (1998). The metabolic cost of neural information. *Natural Neuroscience*, 1(1):36–41.
- Lowell, J. and Pollack, J. (2014). The effect of connection cost on modularity in evolved neural networks. In *Proceedings of the 2014 Conference on Artificial Life: International Conference on the Synthesis and Simulation of Living Systems*, pages 821–826, New York, USA. MIT Press.
- Nitschke, G. (2005). Emergence of cooperation: State of the art. *Artificial Life*, 11(3):367–396.
- Nitschke, G. and Didi, S. (2017). Evolutionary policy transfer and search methods for boosting behavior quality: Robocup keep-away case study. *Frontiers in Robotics and AI*, 4(1).
- Nitschke, G., Schut, M., and Eiben, A. (2012). Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm and Evolutionary Computation*, 2(1):25–38.
- Seth, A. and Edelman, G. (2004). Environment and behavior influence the complexity of evolved neural networks. *Adaptive Behavior*, 12(1):5–20.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–659.
- Williams, S. and Yaeger, L. (2017). Evolution of neural dynamics in an ecological model. *Geosciences*, 7(3):doi:10.3390/geosciences7030049.
- Yaeger, L. (2009). Evolution of neural dynamics in an ecological model. *HFSP Journal*, 5(3):328–339.

Graph Product Representation of Organism-Environment Couplings in Evolution

Hiroki Sayama^{1,2}

¹Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, USA

²Waseda Innovation Laboratory, Waseda University, Japan
sayama@binghamton.edu

Abstract

We present a theoretical framework that mathematically formulates the evolutionary dynamics of organism-environment couplings using graph product multilayer networks, i.e., networks obtained by “multiplying” factor networks using some graph product operator. In this framework, one factor network represents different options of environments and their mutual physical reachability, and another factor network represents possible types of organisms and their mutual evolutionary reachability. The organism-environment coupling space is given by a Cartesian product of these two factor networks, and the nodes of the product network represent specific organism-environment combinations. We studied a simple evolutionary model using a reaction-diffusion equation on this organism-environment coupling space. We numerically calculated correlations between the inherent fitness of organisms and the actual average fitness obtained from the graph product-based evolutionary model, varying the spatial diffusion rate while keeping the type diffusion rate small. Results demonstrated that, when the spatial diffusion is sufficiently slow, the correlation between inherent and actual fitnesses drops significantly, where it is no longer valid to assume that fitness can be attributed only to organisms.

A classical view of biological evolution typically assumes that the fitness is an attribute of individual organisms or genes. However, this simplistic view is known to be invalid in many realistic scenarios, e.g., when the habitats are spatially non-homogeneous and organisms’ fitnesses depend on their surrounding environments (Tilman and Kareiva, 2018). Here we present a theoretical framework that mathematically formulates the evolutionary dynamics of such organism-environment couplings using *graph product multilayer networks* (Sayama, 2018), i.e., networks obtained by “multiplying” factor networks using some graph product operator. While graph products have already been used in theoretical evolutionary biology (e.g., to model the quasi-independence of characters in phenotype space (Wagner and Stadler, 2003)), our work is unique in that it utilizes graph products to represent interactions between organisms and environments.

In this work, we use Cartesian products to represent couplings of environmental conditions and organismal types.

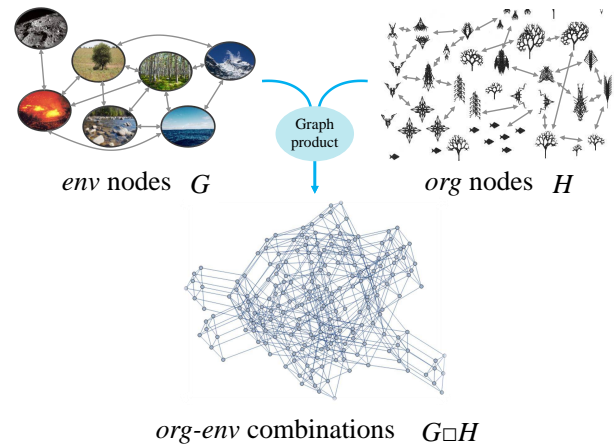


Figure 1: Graph product representation of organism-environment couplings in evolution. Graph G represents physical reachability among different options of environments, while graph H represents evolutionary reachability among different types of organisms. Their Cartesian product $G \square H$ (bottom) represents their couplings in a single product network.

Cartesian products are the most straightforward way to construct graph products, by connecting nodes (= elements of a Cartesian product of original node sets) to each other by following the edge structure of one factor graph or the other. We consider n_G different options (geographically separated habitats, co-located ecological niches, etc.) of environments as *env*-nodes, and represent their mutual physical reachability by a weighted graph G that connects those *env*-nodes (Fig. 1, top left). We denote the average amplitude of G ’s edge weights as δ , which represents the rate of spatial diffusion (e.g., migration). Similarly, we consider n_H possible types (genotypes, phenotypes, etc.) of organisms as *org*-nodes, and represent their mutual evolutionary reachability by another weighted graph H that connects those *org*-nodes (Fig. 1, top right). We denote the average amplitude of H ’s edge weights as μ , which represents the rate of type diffu-

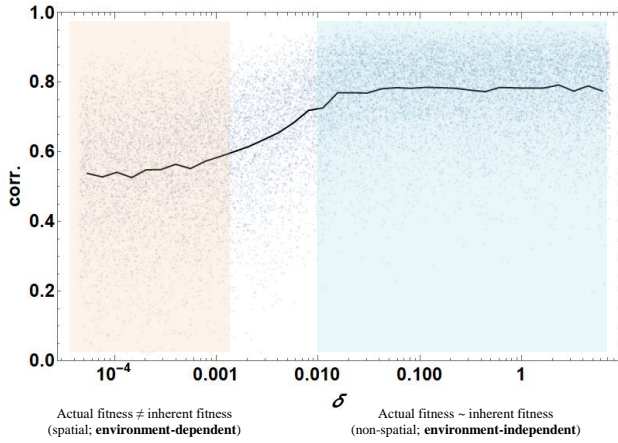


Figure 2: Spearman's rank correlation coefficients between organisms' inherent and actual fitnesses plotted against δ . Each dot represents a result from one trial (20,000 trials in total). The black curve shows means. $n_G = n_H = 20$, $\mu = 10^{-5}$, $\alpha = 0.6$.

sion (e.g., mutation).

Using these graphs, the organism-environment coupling space is given by a Cartesian product multilayer network $G \square H$, whose nodes represent specific *org-env* combinations (Fig. 1, bottom). In this framework, both the spatial diffusion on G and the type diffusion on H can be uniformly represented as a single diffusion process on $G \square H$. The Laplacian spectrum of $G \square H$ can be analytically derived from Laplacian spectra of G and H thanks to mathematical properties of graph product multilayer networks (Sayama, 2016).

Here we study the evolutionary dynamics on the organism-environment coupling space using a simple reaction-diffusion evolutionary dynamics

$$\frac{ds}{dt} = F(s) - Ls, \quad (1)$$

where s is a vector that represents activities (e.g., populations) of $n_G \cdot n_H$ *org-env* combinations, function $F(s)$ represents the local population dynamics, and L is the Laplacian matrix of $G \square H$. Here we consider a simple linear case where F is a diagonal fitness matrix

$$F = (1 - \alpha)I_G \otimes F_H + \alpha F_e, \quad (2)$$

where I_G is a $n_G \times n_G$ identity matrix, F_H is a $n_H \times n_H$ random diagonal matrix that represents the inherent (environment-independent) fitnesses of organisms, F_e is a $n_G \cdot n_H \times n_G \cdot n_H$ random diagonal matrix that represents the environment-dependent fitnesses of organisms, α is the strength of organism-environment coupling, and \otimes is the Kronecker product operator.

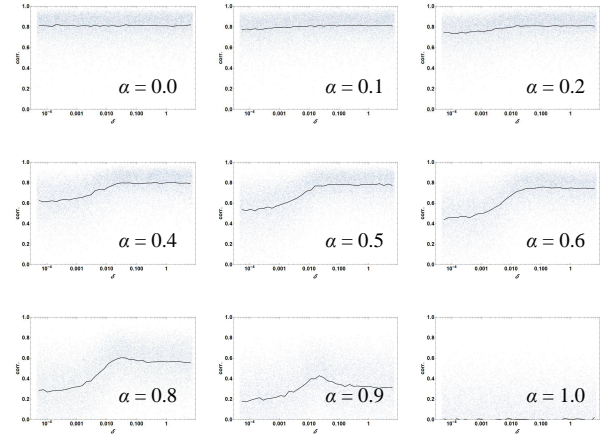


Figure 3: Dependence of correlations between organisms' inherent and actual fitnesses on α . Each panel shows a plot for a specific value of α in the same way as in Fig. 2.

We numerically calculated Spearman's rank correlation coefficients to detect (potentially nonlinear) correlations between the inherent fitness given in F_H and the actual average fitness obtained from the dominant eigenvector of $(F - L)$, varying spatial diffusion rate δ logarithmically while keeping the type diffusion rate μ small. Figure 2 shows the key illustrative results. It was observed that, when the spatial diffusion is sufficiently slow, the correlation between inherent and actual fitnesses drops significantly. More systematic large-scale numerical experiments with varied values of α (Fig. 3) further revealed that this transition was robustly observed for a wide range of α values.

These results indicate that, when spatial mixing is not sufficiently strong (i.e., low δ), attributing fitnesses to individual organisms alone is not a valid way to describe evolutionary dynamics. We derived this conclusion using graph product multilayer networks, successfully demonstrating the usefulness of this novel theoretical framework in evolutionary studies.

References

- Sayama, H. (2016). Estimation of Laplacian spectra of direct and strong product graphs. *Discrete Applied Mathematics*, 205:160–170.
- Sayama, H. (2018). Graph product multilayer networks: spectral properties and applications. *Journal of Complex Networks*, 6(3):430–447.
- Tilman, D. and Kareiva, P. (2018). *Spatial Ecology: The Role of Space in Population Dynamics and Interspecific Interactions (MPB-30)*, volume 30. Princeton University Press.
- Wagner, G. P. and Stadler, P. F. (2003). Quasi-independence, homology and the unity of type: A topological theory of characters. *Journal of Theoretical Biology*, 220(4):505–527.

Maximization of Transfer Entropy leads to Evolution of Functional Differentiation of Swarms

Hiromitsu Suganuma¹, Yuji Kawai², Jihoon Park² and Minoru Asada²

¹School of Engineering, Osaka University

²Institute for Open and Transdisciplinary Research Initiatives, Osaka University

Abstract

We aimed to investigate the principle of emerging interactions between swarms using the functional differentiation theory of the brain. We propose a heterogeneous swarms model, where two swarms having different parameters evolve to maximize transfer entropy between them. In our simulation, we found the emergence of heterogeneous behavior among the swarms, and the appearance of several interaction patterns depending on the degree of the transfer entropy. Our results imply that the same principle of functional differentiation may underlie both the brain and swarms, leading to a novel design of brain-inspired swarm intelligence.

Introduction

Diverse types of interactions between swarms, e.g., prey-predator and leader-follower, are observed in organisms' behavior. In such collective behavior, swarms have varying roles, i.e., they are functionally differentiated. This heterogeneity is reportedly a key in the emergence of interactions and pattern formation (Sayama, 2009).

In mathematical neuroscience, Yamaguti and Tsuda (2015) proposed self-organization with constraints as a principle of functional differentiation in the brain. They showed that maximizing the transfer entropies between neural modules allows the modules to have different dynamics and to interact with each other.

In this paper, we hypothesize that the principle of functional differentiation can also be applied to multi-swarm interactions. We extend the conventional boids model (Reynolds, 1987) to a heterogeneous swarms model and maximize the transfer entropies between the swarms using a genetic algorithm. Consequently, functional differentiation in the swarms is expected to emerge as in the brain.

Model

We propose a heterogeneous swarms model (Fig. 1), in which two swarms have different boid parameters. In a typical boids model, the velocity of the i th agent is updated based on its neighbor agents:

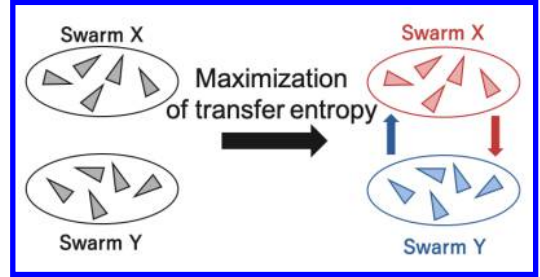


Figure 1: Heterogeneous swarms model. In an initial generation (left), all agents have the same parameters, and the labels of swarms, X or Y, are randomly assigned to agents. Agents belonging to a swarm have the same parameters which can be different from those of the other swarm. Then, the parameters for swarms X and Y are optimized to maximize transfer entropies between them. Consequently, the parameters become heterogeneous and the swarms begin to interact, suggesting that their functions have now differentiated.

$$\Delta \vec{v}_i = w_s \sum_{j \in S_s} \frac{\vec{x}_i - \vec{x}_j}{|\vec{x}_i - \vec{x}_j|} + w_a \left(\vec{v}_i - \frac{\sum_{j \in S_a} \vec{v}_j}{n_a} \right) + w_c \left(\vec{x}_i - \frac{\sum_{j \in S_c} \vec{x}_j}{n_c} \right) + \epsilon(t). \quad (1)$$

where \vec{x}_i and \vec{v}_i denote the position and velocity of the i th agent, respectively. The parameters, w_s , w_a , and w_c denote the weights for separation, alignment, and cohesion among agents, respectively. S_s , S_a , and S_c denote the set of neighboring agents for each rule, and n_a and n_c indicate the number of the neighboring agents. Noise $\epsilon(t)$ is added so as to make behavior complex.

Our model assumes two swarms, labeled as X and Y. They have different weights while all agents have the same weights in the typical model. For example, the weight for separation, W_s , consists of weights for the inter-swarms, $w_{X \rightarrow Y}$ and $w_{Y \rightarrow X}$, and weights for the intra-swarms,

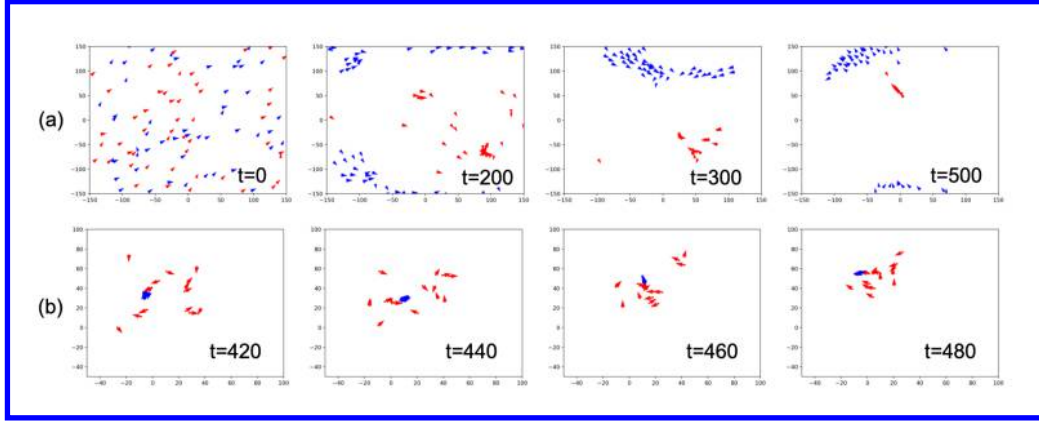


Figure 2: Behavior of the optimized heterogeneous swarms. (a): prey-predator like dynamics when the fitness was higher. (b): rotating leader dynamics when the fitness was lower.

$w_{X \rightarrow X}$ and $w_{Y \rightarrow Y}$:

$$W_s = \begin{bmatrix} w_{s_{X \rightarrow X}} & w_{s_{X \rightarrow Y}} \\ w_{s_{Y \rightarrow X}} & w_{s_{Y \rightarrow Y}} \end{bmatrix}. \quad (2)$$

The weights for alignment and cohesion are defined in the same manner.

The twelve weight parameters in total are optimized using a standard genetic algorithm. The fitness is defined as the product of transfer entropies between the averaged series of the velocity of the swarms: $T_{X \rightarrow Y} * T_{Y \rightarrow X}$, where $T_{X \rightarrow Y}$ denotes the transfer entropy from swarm X to swarm Y, and $T_{Y \rightarrow X}$ denotes vice versa. In the initial generation, the labels of swarms are randomly assigned to agents, and they have the same weight parameters. Then, the weights are optimized to maximize the transfer entropies through elite selection.

Results

We constructed a 2D simulator of 100 agents, in which a swarm consisted of 50 agents. The population in a generation was 96. The history length and delay for transfer entropy were 1.

Simulation results showed that the fitness converged in two values: approximately 0.3 and 0.04. Fig. 2 (a) and (b) illustrate behavior of the swarms when the consequent fitnesses were high and low, respectively. In both cases, the swarms had different parameters and seemed to have different functions. In Fig. 2 (a), the blue swarm appears to be escaping from the red swarm, which looks like prey-predator interaction. In Fig. 2 (b), the blue swarm appears to rotate and the red swarm loosely follows it.

Discussion

In this study, we proposed a heterogeneous swarms model, where two swarms with different parameters evolve to maximize transfer entropy between them. Our simulation showed

that functional differentiation and interactions between heterogeneous swarms were developed by maximizing the transfer entropy. The interaction patterns of the swarms varied depending on the fitness, suggesting that the same principle may underlie function differentiation in swarms and the brain. We suppose that the interaction between heterogeneous swarms is the foundation of division of labor, which is ubiquitously observed in biological systems, especially in social insects (Duarte et al., 2011). In addition to the conventional explanation that division of labor is due to efficient foraging, our model suggests that such an evolutionary process might involve an increase in information transfer or communication between swarms. In future, we plan to develop our model further by imposing tasks on the swarms to investigate the contribution of the functional differentiation to task performance, adaptability, and robustness.

Acknowledgements

This work was supported by JST CREST Grant Number JP-MJCR17A4, Japan.

References

- Duarte, A., Weissing, F. J., Pen, I., and Keller, L. (2011). An evolutionary perspective on self-organized division of labor in social insects. *Annual Review of Ecology, Evolution, and Systematics*, 42:91–110.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, 21:25–34.
- Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15(1):105–114.
- Yamaguti, Y. and Tsuda, I. (2015). Mathematical modeling for evolution of heterogeneous modules in the brain. *Neural Networks*, 62:3–10.

Evolving Recurrent Neural Network Controllers by Incremental Fitness Shaping

Kaan Akinici¹, Andrew Philippides²

^{1,2}Informatics, University of Sussex

¹ka373@sussex.ac.uk

²andrewop@sussex.ac.uk

Abstract

Time varying artificial neural networks are commonly used for dynamic problems such as games controllers and robotics as they give the controller a memory of what occurred in previous states which is important as actions in previous states can influence the final success of the agent. Because of this temporal dependence, methods such as back-propagation can be difficult to use to optimise network parameters and so genetic algorithms (GAs) are often used instead. While recurrent neural networks (RNNs) are a common network used with GAs, long short term memory (LSTM) networks have had less attention. Since, LSTM networks have a wide range of temporal dynamics, in this paper, we evolve an LSTM network as a controller for a lunar lander task with two evolutionary algorithms: a steady state GA (SSGA) and an evolutionary strategy (ES). Due to the presence of a large local optima in the fitness space, we implemented an incremental fitness scheme to both evolutionary algorithms. We also compare the behaviour and evolutionary progress of the LSTM with the behaviour of an RNN evolved via NEAT and ES with the same fitness function. LSTMs proved themselves to be evolvable on such tasks, though the SSGA solution was outperformed by the RNN. However, despite using an incremental scheme, the ES developed solutions far better than both showing that ES can be used both for incremental fitness and for LSTMs and RNNs on dynamic tasks.

Introduction

While deep feed-forward neural networks have been used very successfully in static problems where there is no temporal dependence between inputs, non-Markovian problems such as controllers for robots or games could potentially benefit by temporally extended networks (networks with a temporal element). Long short-term memory (LSTM) networks – which have complex forms of memory – are interesting networks because of their potential to capture long term temporal dependencies and have been used successfully in a number of tasks. (Sutskever, et al. 2014). However, it is harder to find the optimal settings for these networks using things like back-propagation due to exploding gradient problems (Salimans, et al. 2017). Indeed, to train LSTMs, people typically use back-propagation through time or reinforcement learning (Bakker, 2001). Recently, evolutionary optimisation has been used as an alternative to reinforcement learning to develop solutions since it requires less computational power per episode and memory (Salimans,

et al. 2017). Here we therefore see if it is possible to evolve an LSTM for a lunar lander game using either a steady-state genetic algorithm (SSGA) or an evolutionary strategy (ES). While ES's have been shown to outperform GAs on a number of tasks, it is not clear if it will be true for such dynamic networks operating in irregular, noisy, fitness landscapes.

Many different evolutionary algorithms have been effective at finding solutions for robotics tasks, not least as they provide a very flexible approach. For example, it is relatively easy in such problems to encourage robustness to problem variations and generalisability across starting conditions through evaluating the agent on multiple trials. In the lunar lander problem, this robustness is necessary as the landing surface and starting conditions are randomly generated for each instantiation. Steady-state genetic algorithms are particularly good at encouraging robustness because solutions that perform well stay in the population and are evaluated many times over the course of evolution, meaning they experience a very wide range of starting conditions without incurring the computational cost of evaluating all solutions in the population over the same number of trials. As evolutionary strategies work by having multiple copies of a single individual, which then moves through fitness space, it is not clear if they will lead to a similar level of generalization. Another way in which the flexibility of evolutionary algorithms helps in dynamic optimisation tasks, is that the issue of local optima in the search space can be ameliorated by techniques such as fitness shaping and incremental evolution in which the designer can guide the solution to the types of behaviour desired. Again, such schemes have been used successfully with SSGAs, it is not clear that they will also function well with ESs. As the lunar lander game with the default fitness function has local optima issues (as the agent can gain a reasonable score by doing nothing) as well as requiring solutions robust to starting conditions, we here use it as test-bed to see if an incremental fitness scheme can be as effective for an ES as it is for an SSGA and if so, whether the solutions generated are robust to changes in conditions.

Despite the issues of a large local optima and very noisy fitness evaluations, we show that LSTM networks can successfully be evolved with both evolutionary algorithms. However, through comparison with an RNN evolved with NEAT (used as a benchmark and to derive the network morphology for the evolutionary algorithms) and ES, the SSGA, while able to produce a viable network, does not appear to be taking advantage of the rich dynamics provided by the LSTM. In contrast, despite not being population-based in the same way as the SSGA, the ES generates robust controllers demonstrating both that these algorithms can

function effectively with both incremental fitness functions, noisy evaluations and highly dynamic LSTM networks.

Background and Methods

In this section we briefly describe the methods that were used during the experiments.

Long-Short Term Memory (LSTM) Networks

Long Short-Term-Memory (LSTM) networks are advanced versions of RNN networks that can selectively forget and update hidden states. In a basic LSTM perceptron there are four different gates that determine the output and hidden states (Gers, at al. 2002). These gates are commonly referred as “forget”, “input”, “recursive memory” and “output”. Additionally, LSTM networks have a hidden state and a memory. These properties allow LSTM networks to be aware of past actions and experiences thus enabling long-term temporal dependencies in the decision making process.

During the experiment, all of the parameters controlling the shape of the LSTM network were kept constant as evolving the LSTM shape as well would require a bigger study and more computational power and we were here interested in whether ESs would be able to work in noisy spaces with fitness shaping. In order to choose the morphology, initial experiments were conducted using NEAT and RNNs. Following tests, the size of the layers and depth of the network were chosen to be slightly bigger than the best NEAT derived RNN. Specifically, the LSTM comprised two ReLU layers of 20 and 25 units followed by a fully connected softmax layer.

Genetic Algorithm

Genetic algorithms are random heuristic search algorithms that are inspired by biology. Genetic algorithm are commonly used in multi-variable optimisation tasks (Gers, at al. 2002). They rely on continually evaluating different combinations of variables on an optimisation task and using the result as feedback to improve the variable choice. In order to use these algorithms, variables to be optimised are represented by genes and solutions are represented by genomes. A genome thus consists of all variables and their corresponding values. The result of the optimisation for the given genome is referred to as the fitness of the genome and the set of the genomes are referred to as the population (Whitley, 2001). Genetic algorithms use an analogue of an evolutionary process to iteratively improve the population. Many variations of genetic algorithms exist, but the majority work by using the fittest individuals in a population to generate the next population via selection, crossover and mutation.

In our experiments, the weights of the LSTM or RNN are used as the genotype meaning it is 7,040 variables long for the LSTM and 1,835 long for the RNN. As a crossover operator we used uniform crossover which produces two offspring, where the first offspring has 80 percent chance per gene to get genes from first parent and 20 percent chance to get from second parent, and second offspring has 80 percent chance from first parent and 20 percent chance from second parent. For mutation we used uniform mutation in the range of [-1,1] with 0.35% mutation chance per genome. We also used a steady state genetic algorithm, which instead of replacing all

of the population at once, as in a generational GA, iteratively selects two parents and produces two offspring which replace two members of the current population. The original replacement method of the population was replacing the parents, as shown in Pramanik and Setua (2014). However, we found the method proposed by Gilbert Syswerda (1991) to work better in our experiments (Pramanik and Setua, 2014; Syswerda, 1991). Specifically, in order to choose the individuals for mating, proportional roulette wheel selection was used and in order to choose the individuals to replace, reverse proportional roulette wheel was used. The architecture of the networks were as described above and a population size of 30 was used.

Evolutionary Strategy

An Evolutionary Strategy (ES) is a nature-inspired algorithm (Salimans, at al. 2017) and is a variant of a GA in which new individuals are not generated by random variation of one or two parents via crossover and mutation. Instead, an ES uses the statistics of the current population within the fitness space to determine new individuals and thus the direction of improvement in fitness space. An ES thus effectively uses mutation only and no crossover. Here we used the ES algorithm that is described by Salimans (Salimans, at al. 2017). Specifically, at each evaluation there is only one ‘real’ individual and N-1 (where N is the population size) simulated individuals which surround the real individual. Each gene of each simulated individual is constructed by adding a small amount of Gaussian-distributed noise to the corresponding gene of the real individual. The noise of each gene of each individual is scaled by its fitness value. The product is added to the genes of the real individual resulting in a new real individual (Salimans, at al. 2017). N-1 simulated individuals are created around this one individual and the process repeats.

The architecture of the networks was the same as the steady state network and a population size of 30 was used.

Algorithm 1: Evolutionary Strategies

```

for t=0,1,2,... do
  Sample  $\epsilon_1, \dots, \epsilon_i$  individuals using Gaussian noise
  Compute fitness  $F_i = F(\theta_t + \sigma \epsilon_i)$  for  $i=1..n$ 
  Set  $\theta_{t+1} = \theta_t + \alpha \frac{1}{n \sigma} \sum_{i=1}^n F_i \epsilon_i$ 
end for

```

Algorithm 1 displays how each generation was created. n represents the population size, θ_t represents the real individual’s genes, α represents learning rate, σ represents variance of the Gaussian noise, F_i represents the fitness value of i ’th individual. The variance of the Gaussian distribution was chosen to be two and the learning rate was chosen to be 0.5.

NEAT (Neuroevolution of Augmenting Topologies)

In order to get an idea of what network size might be effective for evolution and thus avoid complications due to over/under sized network structure, we first trained an RNN with NEAT, an approach which has proved successful previously (Stanley and Miikkulainen, 2002). NEAT is an advanced version of a

Weight Evolving Artificial Neural Network (TWEANN), a specialised genetic algorithm which focuses on optimisation of the weights of an artificial neural network (Stanley and Miikkulainen, 2002). The major difference between NEAT and TWEANN is topological evolution. In other words, the composition and the architecture of the network evolve over time together with its weights. Since the NEAT library does not support LSTMs, RNNs were used for these initial experiments, using the NEAT algorithm described by Stanley and Miikkulainen (2002). Due to limited computational power, we used a relatively small population of 30 with minimum species size 3 and 2 elites per deme. ReLU was used as the activation function.

Lunar Lander Game

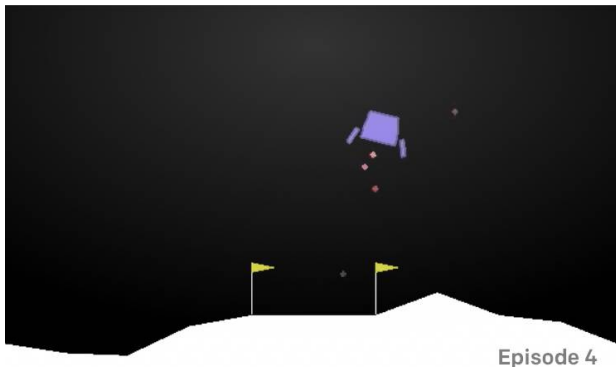


Figure 1: OpenAI Gym Lunar Lander Environment

The Lunar Lander game is a simulation provided by OpenAI (Brockman, et al. 2016) in which controllers have to land a spacecraft in a designated landing area indicated by the flags in Figure 1. Specifically the flags are at coordinates (-0.25,0) and (0.25,0) where (0,1) is the starting position of the spacecraft. The problem is made more difficult as the surface of the moon is randomly generated on every evaluation. The simulation area is bounded and if the spacecraft exits the boundary, the simulation stopped. While the spacecraft always starts from the same coordinates, (0,1), the starting angle is random as is the initial velocity (both magnitude and direction).

The environment is frictionless but subject to a constant gravity force towards the surface of the moon. The spacecraft has three thrusters, bottom, left and right. The bottom thruster increases spacecraft’s speed in the direction it is facing, while left and right make it rotate clockwise/counter-clockwise respectively. Each thruster produces a constant thrust and the spacecraft is subject to basic momentum constraints. In any given instance, the spacecraft could only use one of its thrusters. The goal was to land the spacecraft without crashing (landing with more than -0.6 vertical velocity or not upright) and using the minimum amount of fuel. The total fuel was unlimited however; the spacecraft was limited to 1000 action commands.

The Lunar Lander game expects one command for input at each step. These commands are coded as “0”, “1”, “2” or “3” which correspond to: *do nothing*, *use left thruster*, *use bottom thruster* and *use right thruster*, respectively. After the game receives one of those commands, it returns four parameters, which are *observation*, *score*, *done* and *info*. The observation

parameter, which represents the state of the environment, has eight variables: “x” and “y” coordinates, speed in vertical and horizontal axes, facing angle, angular velocity, and two Booleans, leg 1 and leg 2, which state whether a leg touches the ground in the goal zone (and which we do not use in network training though the default fitness function does). The *score* parameter is the value of default fitness function and *done* is a Boolean indicating the simulation’s stop condition.

Fitness Shaping Through Incremental Evolution

The default fitness function (displayed below) provided by the simulation discourages fuel consumption (*TrusterPower* terms) while encouraging every action that shortens the distance between the spacecraft and the goal position and penalising every action that increases the distance between the spacecraft and the goal location (first two terms of the equation). Touching the goal position rewards 10 points per leg of the spacecraft. Also attempting to takeoff after landing is discouraged and causes negative points. The formula of the default fitness function is given as;

$$fitness_t = -100\sqrt{x_t^2 + y_t^2} - 100\sqrt{x_{acc(t)}^2 + y_{acc(t)}^2} - 100angle_t + 10leg1 + 10leg2 - fitness_{t-1} - 0.3MainTrusterPower - 0.03SideTrusterPower$$

Because of the penalization of fuel consumption (Main/Side Truster Power term in the equation), attempts to evolve a solution with the default fitness were subject to issues with local optima (see Results) in which the spacecraft does not apply any thrusters and simply falls down. We explored different fitness functions to get around this issue (e.g. applying multiple thresholds to the action commands and discouraging use of the “0” action command) but were not successful.

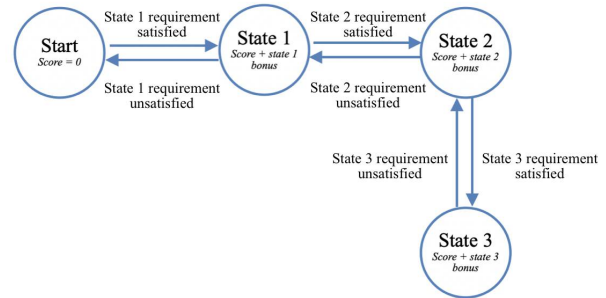


Figure 2: Incremental Fitness Function State Diagram

Thus we designed a fitness function based on different behavioural states, as illustrated at Figure 2. This fitness function was designed to define the problem in a more behavioural way, eliminate the issue of fuel consumption and decreasing the number of variables in the fitness function. Instead of trying to measure the properties of the agent and to determine if it is good or not, this fitness function categorises the current state of the agent and gives bonuses to certain actions/behaviours. With this approach we aimed to add action priorities into the evolutionary process. However the key

feature is that the agent has to perform well at a given state, meaning a certain type of behaviour has been developed, before the next state is evaluated. These states and transitions are:

State 1: Flying state. Detected if the agent was flying. The detection was made by thresholding. If the vertical velocity of the agent was between 0.0 and -0.80, the agent was rewarded 0.5 points and state 2 was enabled. If the agent's velocity was less than -0.80 but it was using its bottom thruster, it was rewarded 0.1 point. However, the transition to state 2 was disabled.

State 2: Horizontal stabilisation state. The agent was rewarded 0.5 points and state 3 was enabled if the angle of the agent was between 0.8π and -0.8π , which translates to being perpendicular to the ground as 0 is the angle of the normal to the ground, with some offset. Also the agent was rewarded 0.1 points if the angle was off-perpendicular but it used one of its side thrusters to correct. If neither of these conditions were met, 0.1 was subtracted.

State 3: Flight route minimisation state. This state rewards the flight trajectory of the agent. Every step that took the agent closer to the landing location gave $100 * displacement$ points. If the displacement was negative (going away from the landing area), nothing was rewarded.

The rationale behind these states were that the agent should be in control of its speed in order to fly, to be able to fly to control its orientation, and to be able to guide its orientation to fly towards the landing pad, which is the desired final behaviour.

We initially tried implementing the behavioural fitness function with all states used together i.e. without gating the states by only evaluating if it passed the previous one, but it is overly complicated and does not produce good results. In order to reduce the complexity of the heuristics, as suggested by Barlow (at al. 2004), an incremental fitness function was implemented. In this approach the behavioural fitness function is gradually modified. The heuristic fitness function was thus divided into three stages, hover, stabilise and orient (States 1, 2 and 3). The first stage of the incremental fitness function prioritises the flight time and the transition between State 1 to State 2 is blocked. For the second stage, the stabilisation phase, the fitness function was allowed to move from State 1 to State 2, but transition to State 3 was blocked. At the third stage, orient, the transition to the third state was enabled. This approach enabled the agent to develop certain behaviours easily while retaining previously learned behaviour.

Resampling and Noise Reduction

As the simulation environment randomly changed at every evaluation, some environments were more suitable for landing while some environments were less. As Pietro (2004) states, noise in the environment decreases the learning rate and the population has a chance to forget what it has learned (Di Pietro, at al. 2004). However, completely eliminating randomly generated landscapes and performing trials on a single static landscape would encourage overfitting and would prevent generalisation of the network. Thus, each network performed on n different environments and the fitness scores were sorted in an ascending order and combined as described below:

$$fitness = \sum_{i=1}^n \frac{fitness_{i-1}}{i + 1}$$

This function decreases the effect of the best episode and increases the effect of the worst episode. The individuals that perform well at all episodes perform significantly better than those that are only good at certain situations. This fitness function contributes to the generalisation of the network. The result of 10 episodes will be referred as a trial.

Since the environment was regenerated in every trial, the complexity of the solution varies between each trial. There were three different elements that contribute to the characteristics of the environment which were: the landscape, the starting angle of the agent and the starting vector. The diverse set of starting angles and the starting vectors ensured the network could not overfit to any action command or to find a fixed set of actions that lead to success. Also, the landscape alters the vector effect of the thrusters, e.g. the bottom thruster may have produced velocity on the horizontal axis due to an obstacle in the environment. All of these factors increase the complexity and high variance due to noise of the simulation. In order to solve the issue with the landscape, every generation performed "n" different episodes, where the k'th trial of i'th individual had the same random factors as the k'th trial of j'th individual. "n" randomly generated landscapes were selected with "n" different starting vectors which were picked from uniformly distributed random values. In order to prevent bias in the starting angles, the starting angles were divided into three groups, left, right and middle. The starting angle varied in between " $-\pi$ " and " π ", where "0" is perpendicular and " π " and " $-\pi$ " is horizontal to the ground. The starting angles for the left group were chosen from randomly distributed values between " -0.53π " and " -0.2π ". The angles for the right group were chosen from randomly distributed values between " 0.2π " and " 0.53π ". The angles for the middle group were chosen from randomly distributed values between " -0.2π " and " 0.2π ". The number of left angles and right angles were distributed equally while the ratio of the middle group was one in seven trials. This method was implemented to give a range of starting angles of similar difficulty, hence as the middle angle was the easiest to solve the issue, the ratio of the middle angle was the lowest.

Results and Analysis

We started using the default fitness function but it soon became evident that there was a local minimum in the fitness space. We ran the Steady State Genetic Algorithm with LSTM network with the default fitness function, for 5,200 trials (52,000 evaluations), where each trial consists of 10 resampled episodes however, the fitness does not improve and behaviour of the resulting network was the same in all runs.

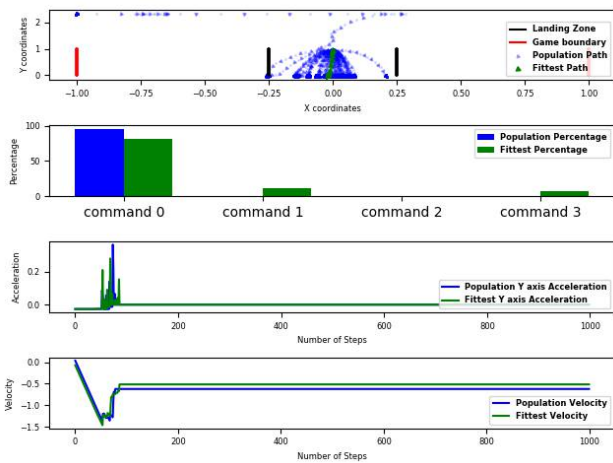


Figure 3: SS GA evolved by default fitness function. The top panel shows the flight trajectories, while the second panel shows the percentages of the actions that are being used during the flight. Third and fourth panels display the acceleration and velocity over time.

Figure 3 illustrates the behaviour of SS GA. The top panel shows the flight trajectories, while the second panel shows the percentages of the actions that are being used during the flight. The third panel displays the acceleration and velocity of the agent. The default fitness function has an unavoidable local optimum, where the agent abuses command 0 and does not use any other command and during the training process, the fitness of the population didn't improved. The reason behind the behaviour was the punishment of fuel usage. In other words, if the agent does not do anything, it will not consume any fuel and since fuel consumption was discouraged, the agent will perform better than the ones that were trying to fly.

With the introduction of the incremental fitness function and fitness shaping, we were able to avoid this local minimum. During the training with the incremental fitness function the fitness was improved. Thus, we decided to run the algorithm longer periods in order to obtain a well developed model. After, running the algorithm with the incremental fitness function for 20,000 trials (250,000 evaluations), the behaviour of the network had improved.

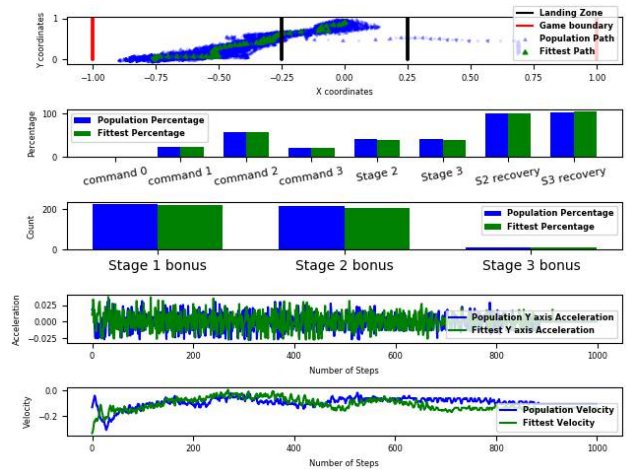


Figure 4: SS GA evolved by incremental fitness function. As in figure 3, The top panel shows the flight trajectories, while the second panel shows the percentages of the actions that are being used during the flight. However it additionally shows the amount of time the agents spends in stages 2 and 3 as well as the amount of steps that led to a transition to step 2 or 3 as step 2/3 recovery respectively, which gives insight into how much of the time the spends in these stages. The third panel shows the additional stage bonuses accrued the fourth and fifth panels display the acceleration and velocity over time. The same conventions are used for the following figures.

The behaviour of the steady state LSTM algorithm is displayed in Figure 4. The algorithm had significant improvements in acceleration and velocity controlling aspects as well as horizontal stabilisation. The agent learned to slow down its speed and slowly glide down. However, it failed to control its flight path. The points earned from each stage in the incremental fitness function are displayed in the middle panel of Figure 4. The stage 1 bonus (descending points) and stage 2 bonus (angle control) were high while stage 3 bonus (flight path) was very low. This behaviour was due to the lack of generalisation at stage 3.

To see if this behaviour was caused by a fundamental problem with using an LSTM network or network size, we ran the same experiment with an RNN network using the NEAT algorithm.

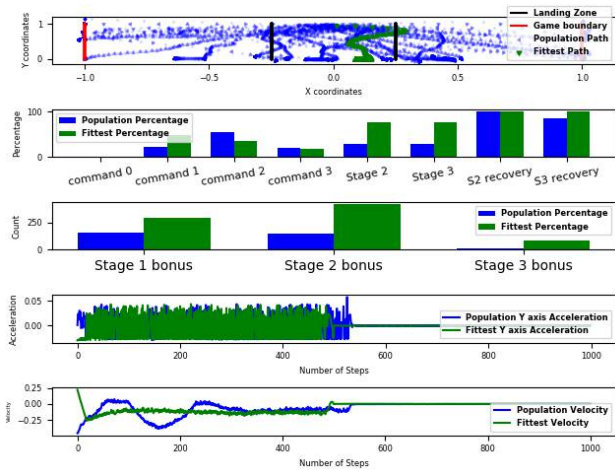


Figure 5: NEAT RNN evolved by incremental fitness function

Figure 5 displays the behaviour of the RNN NEAT network evolved using the incremental fitness function over ~360,000 trials (3,600,000 evaluations). The NEAT network managed to learn all stages showing that the fitness function is viable. In particular, when Figure 4 and Figure 5 are compared, while the LSTM performs similarly in stage 1, RNN with NEAT performs far better at stage 2 and stage 3, which leads to a better control over its orientation and flight path. This lead us to believe that using an LSTM is a viable option but that perhaps it is the type of evolution which is holding performance back. We thus repeated the same experiment with an ES algorithm using the LSTM and RNN network.

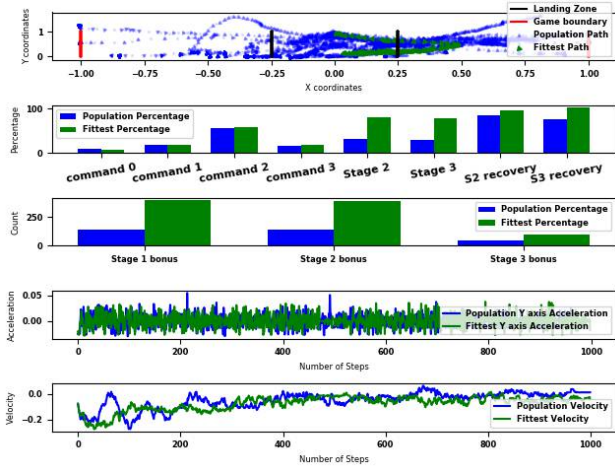


Figure 6: ES LSTM evolved by incremental fitness

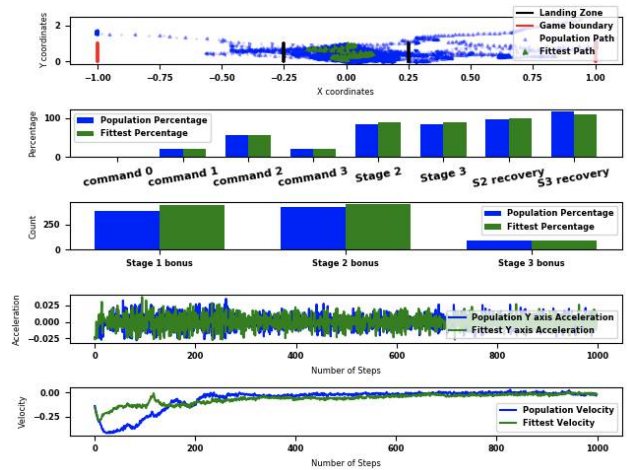


Figure 7: ES RNN evolved by incremental fitness

The behaviour of the evolutionary strategy algorithm with LSTM and RNN network is displayed in Figure 6 and Figure 7 respectively. The agent figured out how to change its flight path towards the landing point while kept its velocity and acceleration in control. The algorithm was run for 6,000 trials (60,000 evaluations) with LSTM and RNN.

The accumulated stage 1 bonus of the ES RNN and LSTM was far higher than NEAT RNN algorithm's while stage 2 and stage 3 bonus were similar. However, note that while the behaviour is complex, the networks actually finds a loophole in the fitness function. This trick was never landing and slowly tilting up and down near the landing area, which cheats the fitness function to think the agent is performing a landing and earns points. The behaviours of ES LSTM and ES RNN were similar. However, ES LSTM was utilising command 0 more than SS LSTM, NEAT RNN and ES RNN algorithms. Also, ES RNN algorithm's population wise fitness was far higher than ES LSTM. This might indicate existence of a wider optimum at a lower dimension.

	Stage 1 Average	Stage 2 Average	Stage 3 Average	Peak Fitness
SS GA LSTM (Population)	-244.37± 125.36	127.51±14.37	251.42±29.74	262.20±12.25
NEAT RNN (Population)	2.25±0.71	3.82±1.54	3.33±16.24	127.82±36.69
ES LSTM (Fittest)	212.21±28.23	783.92±78.91	1674.06± 119.85	1600.25± 1164.59
ES LSTM (Population)	65.38±13.42	332.55±123.18	330.62±171.42	471.68±446.78
ES RNN (Fittest)	167.06±30.69	1509.56±83.46	1662.15±80.90	2127.93± 1255.64
ES RNN (Population)	89.29±15.46	495.85±101.09	481.60±368.68	832.60±549.33

Table 1: Fitness Evaluations

To compare the performance of the algorithms more generally, Table 1 displays average fitnesses of ES LSTM, ES RNN, NEAT RNN and SSGA algorithms over 60,000 evaluations with five random restarts. While ES RNN, ES LSTM and SSGA had similar fitness values, NEAT RNN had lower fitness scores. NEAT algorithm alters the composition of the network during the process of evolution and it starts from one hidden layer which has one node. While this attribute enables the population to produce more unique individuals, it increases the time of convergence. However, ES was the fastest algorithm in both network types and reached a better optimum corresponding to a qualitatively better solution. Even though ES LSTM had lower average population fitness, fittest individuals of both ES RNN and ES LSTM had similar scores.

However, while the ES clearly develops better more dynamic behaviour than the SSGA, as noted in the introduction, one of the benefits of the SSGA is that as good individuals remain in the population and are re-evaluated, good individuals are tested on many different configurations of the task which increases robustness. As the ES works in a different manner, it is not clear whether this robustness accrues in as direct a way as it does for the SSGA. To test for dynamic stability, we thus decreased the left thruster power to 35% of its original power. This resulted in the following figures.

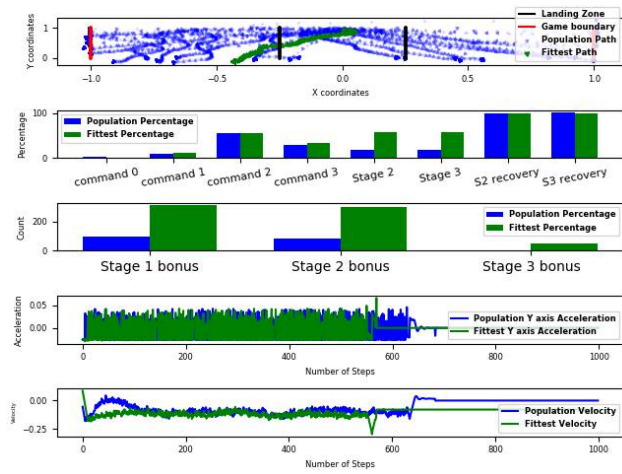


Figure 8: NEAT behaviour while left thruster was crippled

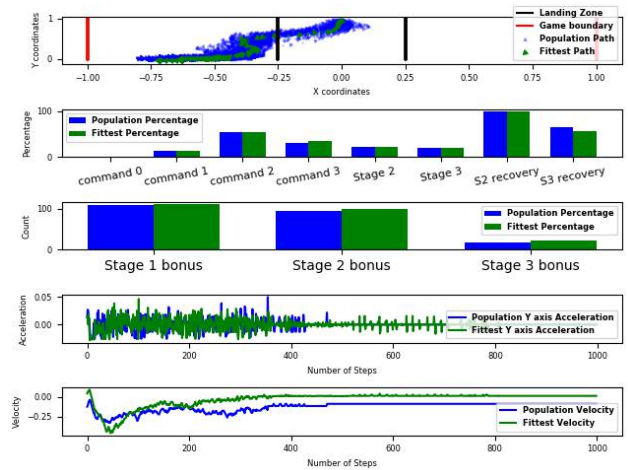


Figure 9: SSGA behaviour while left thruster was crippled

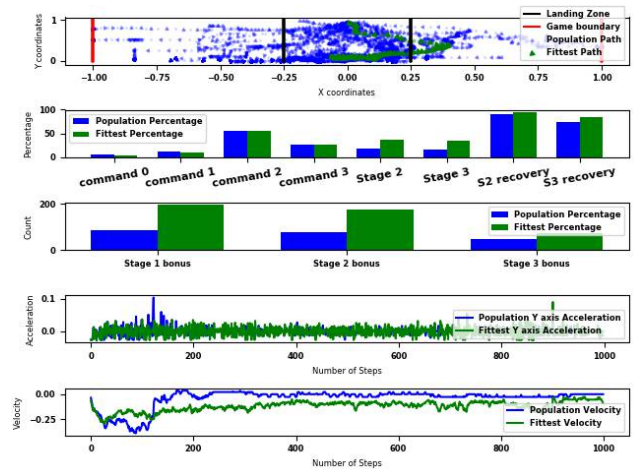


Figure 10: ES LSTM behaviour while left thruster was crippled

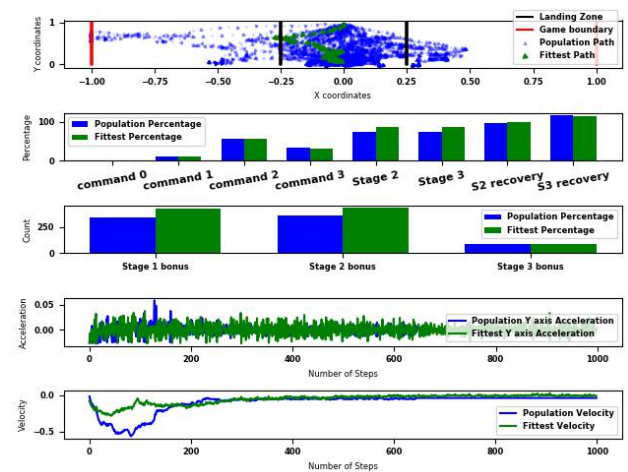


Figure 11: ES RNN behaviour while left thruster was crippled

Overall all algorithms used more Command 3 to compensate the loss of left thruster power (Figures 8, 9, 10,11). NEAT lost most of its control over the agent (Figure 8). SSGA performed similar behaviour as it had before the manipulation (Figure 9). However, when Figures 4 and 9 are compared, the velocity pattern of SSGA can be seen to be changed, indicating it is having a hard time controlling its velocity. In contrast, the ES LSTM and RNN were mostly unaffected (Figure 10, 11). Thus the ES has indeed developed robustness exhibiting homeostatic behaviour even to perturbations not experienced during evolution.

Summary and Discussion

In this experiment, LSTM and RNN networks were evolved using an incremental fitness function to solve a complex controller task: the Lunar Lander game. The steady state algorithm with LSTM was found to be less useful when the landscape was complex and noisy and found the local optimum faster than the NEAT algorithm with RNN network, as the faster converge rate of the population decreased the chance of finding other possible optima. The NEAT algorithm with RNN had a good balance between promoting diversity and was found to produce decent results. However using an ES with LSTM or RNN network was the fastest network and produced adaptive and dynamically stable behaviour even exploiting the fitness function to hover for a long time.

Overall the incremental fitness function was able to eliminate the issue of a local optimum in the fitness space both for a standard GA and for the ES. In addition increasing the complexity gradually helps the population to learn faster perhaps by reducing population convergence. Since the solution space of LSTM networks were more complicated (each neuron has eight weights excluding the bias weights) than RNNs, optimising LSTM networks is a harder task. However, reducing the complexity of the fitness landscape using incremental fitness function showed that LSTM networks can be evolved as fast as RNN networks.

The question of which type of memory in the LSTM networks was beneficial in this task, or whether different gates of the LSTM network require different evolution techniques, is not yet answered. To understand this, we would need to know more about why the LSTM network's behaviour is different to the RNN's. Since LSTM network cells are advanced versions of RNN cells, during the optimisation process they could be turned into RNN's by constraining the parameters correctly. In order to explore how the more complex memory might be valuable, the LSTM network can be initialised with parameters that make it act as an RNN or that allow different memory types only. By analysing the final evolved networks and seeing how much they vary from the standard RNN, the benefits that the different memory types of the LSTM bring to the table can be observed.

However, there are many other factors that affect the LSTM network's performance, such as network size, activation function and population size. The effect of these factors are not yet explored. A more flexible evolutionary method which explores these variables combined with ES algorithms, or a NEAT type process, might boost its performance and produce better results as suggested by Rawal Miikkulainen (2016) who showed that an LSTM network can be evolved using NEAT.

References

- Bakker, B. (2001). Reinforcement Learning with Long Short-term Memory. Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic.
- Barlow, G., Oh, C., and Grant, E. (2004). Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming. In IEEE Conference on Cybernetics and Intelligent Systems 2004. volume 2, pp. 689–694.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Zaremba, W. (2016). OpenAI Gym. CoRR.
- Di Pietro, A., While, L. and Barone, L. (2004). Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. Proceeding of Congress on Evolutionary Computation, 19-23 June 2004, pp. 1254-1261 10.1109/CEC.2004.1331041
- Gers, F.A., Schraudolph, N.N. and Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research (JMLR), 3:115–143.
- Pramanik, S. and Setua S. (2014). A steady state genetic algorithm for multiple sequence alignment. Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014. pp. 1095-1099. 10.1109/ICACCI.2014.6968251.
- Rawal, A. and Miikkulainen, R. (2016). Evolving Deep LSTM-based Memory Networks using an Information Maximization Objective. The Genetic and Evolutionary Computation Conference (GECCO'16), July 20-24, Denver, Colorado, USA.
- Salimans, T., Ho, J., Chen, X., Sutskever, I. (2017). Evolution Strategies as a Scalable Alternative to Reinforcement Learning. ArXiv e-prints, 2017.
- Stanley, K.O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. IEEE Trans. Evol. Comput. 10(2):99–127.
- Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to sequence learning with neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems.
- Syswerda, G. (1991). A study of reproduction in generational and steady-state genetic algorithms. In Gregory J. E. Rawlins, editor, Foundations of Genetic Algorithms. volume 1, pp 94-101, Morgan Kaufmann Publishers.
- Whitley, D. (2001). An overview of evolutionary algorithms: practical issues and common pitfalls. Information & Software Technology, 43:817-831.

Self-Replication in Neural Networks

Thomas Gabor¹, Steffen Illium¹, Andy Mattausch¹, Lenz Belzner², Claudia Linnhoff-Popien¹

¹LMU Munich

²MaibornWolff

thomas.gabor@ifi.lmu.de

Abstract

The foundation of biological structures is self-replication. Neural networks are the prime structure used for the emergent construction of complex behavior in computers. We analyze how various network types lend themselves to self-replication. We argue that backpropagation is the natural way to navigate the space of network weights and show how it allows non-trivial self-replicators to arise naturally. We then extend the setting to construct an artificial chemistry environment of several neural networks.

Introduction

Dawkins (1976) stressed the importance of self-replication to the origin of life. He argued that proto-RNA was able to copy its molecule structure within a soup of randomly interacting elements. This allowed it to reach a stability in concentration that could not be maintained by any other kind of structure. Eventually, life evolved more or less as an elaborate means to maintain the copying of structural information.

Since the early days of computing, the recreation of biological structures has been a target of research, starting from the early formulation of an evolutionary process by Turing (1950) and including famous examples like Box (1957), Conway (1970) or Dorigo and Di Caro (1999). Also see the overviews given by Koza (1994) or Bäck et al. (1997). Although conceived very early as well (Rosenblatt (1958) and Minsky and Papert (1972)), neural networks have only recently found broad practical application for advanced tasks like image recognition (Krizhevsky et al. (2012)), speech recognition (Hinton et al. (2012)) or strategic game playing (Silver et al. (2017)). The variety of uses shows that neural networks are a powerful tool of abstraction for various domains. However, in all these cases neural networks are used with a certain intend, i.e., equipped with a goal function. Through backpropagation, the distance of the network's output to the goal function can be systematically minimized.

The wide variety of application domains shows the power of neural networks as a functional abstraction. For other functional abstractions, such as expressions in the λ -calculus (Church (1932)) or a variety of assembler-like instruction

sets and automata (Dittrich et al. (2001)), it is known that, when a population of random instances of said functional abstractions are set up and allowed to interact, self-replicators arise naturally (see Fontana and Buss (1996) or Dittrich and Banzhaf (1998), respectively). For neural networks, Chang and Lipson (2018) have shown that self-application may lead to the formation of a self-replicating structure, albeit a rather trivial instance of one. In this paper, we (a) repeat these results for a broader range of neural network architectures and (b) extend the interaction model by the notion of self-training, which yields lots of non-trivial fixpoints. This allows us to (c) construct an artificial chemistry setup using neural networks as individuals that (under certain circumstances, of course) reliably produces a variety of non-trivial self-replicators.

Foundations

We provide a brief introduction to how neural networks function, then we proceed to discuss how to apply neural networks to other neural networks and how to train neural networks using other neural networks.

Basics

Neural networks are most commonly made from layers of neurons that are connected to the next layers of neurons and so on. As there are many kinds of neurons (fully connected, recurrent, convolutional) there are also many kinds of layers. Variations of this scheme go up to well established structures within such layers, consisting not only of single functional cells (LSTM, attention mechanism). What they have in common is the base functionality of accepting values (in form of a matrix or vector), application of weights or bias (a matrix of a similar shape, also known as the network's parameters), followed by a specific activation function (linear, sigmoid, relu, tanh, e.g.) that transforms the outputs. Note that while neural networks originated as a model of biological neurons, they cannot accurately fulfill that role anymore and instead serve as general function approximators.

The most basic form of a feed-forward network is a single-layer perceptron, consisting of many fully connected

cells that provide a transformation of the input on basis of its learned parameters. Mathematically, each cell in such a network is described by a function

$$y = f\left(\sum_i w_i x_i + b\right)$$

where x_i is the value produced by the i -th input cell, w_i is the weight assigned to that connection, b is a cell-specific bias and f is the activation function.

The *recurrent neural network* (RNN) structure allows to pass an additional vector h to the current calculation. This improves the performance when processing sequential inputs. The result of the evaluation step t is passed to the evaluation at step $t + 1$ as vector h_{t+1} . A recurrent cell's activation at every time step t is $h_t = f(Wx_t + Wh_{t-1})$ where w are the network weights (Chung et al. (2014)).

A neural network thus defines a function $\mathcal{N} : \mathbb{R}^p \rightarrow \mathbb{R}^q$ for input length p and output length q . A neural network \mathcal{N} is usually given by (a) its architecture, i.e., the types of neurons used, their activation function, and their topology and connections as well as (b) its parameters, i.e., the weights assigned to the connections. Whenever the architecture of a neural net is fixed, we can define a neural network by its parameters $\overline{\mathcal{N}} \in \mathbb{R}^r$. Note that $|\overline{\mathcal{N}}| =_{def} r$ depends on the amount of internal connection and hidden layers, but as all inputs and all outputs must be connected somehow to other cells in the network it always holds that $r > p$ and $r > q$.

Application

In the course of this work, we are interested in having neural networks that can be applied to other neural networks (and can output other neural networks). It is evident that if we want neural networks to self-replicate, we need to enable them to output an encoding of a neural net containing at least as many weights as themselves. We discuss multiple approaches to do so but first introduce a general notation covering all the approaches: We write $\mathcal{O} = \mathcal{N} \triangleleft \mathcal{M}$ to mean that \mathcal{O} is the neural network that is generated as the output of the neural network \mathcal{N} when given the neural network \mathcal{M} as input. When \mathcal{M} and \mathcal{O} are sufficiently smaller than \mathcal{N} , i.e., if $|\overline{\mathcal{M}}| \ll |\overline{\mathcal{N}}|$ and $|\overline{\mathcal{O}}| \ll |\overline{\mathcal{N}}|$, then we can simply define the output network \mathcal{O} via its weights $\overline{\mathcal{O}} = \mathcal{N}(\overline{\mathcal{M}})$. However, these conditions obviously do not allow for self-replication. Thus, we introduce several *reductions* that allow to define the operator \triangleleft differently and open it up for self-replication. Note that for these definitions, we assume that \mathcal{M} and \mathcal{O} have the same architecture and that the application of \mathcal{N} keeps the size of the input network, i.e., $\mathcal{M} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ for some p and $|\overline{\mathcal{M}}| = |\overline{\mathcal{O}}| = p$.

Reduction 1 (Weightwise). *We define $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}$ fixed. Let $\overline{\mathcal{M}} = \langle v_i \rangle_{0 \leq i < |\overline{\mathcal{M}}|}$. We then set*

$$\overline{\mathcal{O}} = \langle w_i \rangle_{0 \leq i < |\overline{\mathcal{O}}|}$$

where $w_i = \mathcal{N}(v_i, l(i), c(i), p(i))$

and $l(i)$ is the layer of the weight i , $c(i)$ is the cell the weight i leads into and $p(i)$ is the positional number of weight i among the weights of its cell. We use $\overline{\mathcal{O}}$ to define $\mathcal{O} = \mathcal{N} \triangleleft_{ww} \mathcal{M}$.

Note that l, c, p depend purely on the networks' architectures and the index of the weight i , not on the value of the weight v_i . Theoretically, we could pass on i to the network directly but it seemed more reasonable to provide the network with the most semantically rich information we have. Also note that we normalize $l, c, p : \mathbb{N} \rightarrow [0; 1] \subset \mathbb{R}$, i.e., the positional values are encoded by reals between 0 and 1 as is common for inputs to neural networks.

Intuitively, the weightwise reduction calls \mathcal{N} on every single weight of \mathcal{M} and provides the weight's value and some information on where in the network the weight lives. \mathcal{N} then outputs a new value for that respective weight. After calling \mathcal{N} for $|\overline{\mathcal{M}}| = |\overline{\mathcal{O}}|$ times, we have a new output net \mathcal{O} . This approach is most similar to the one used by Chang and Lipson (2018).

Reduction 2 (Aggregating). *Let $\text{agg}_a : \mathbb{R}^a \rightarrow \mathbb{R}$ be an aggregator function taking in an arbitrary amount of parameters a . Let $\text{deagg}_a : \mathbb{R} \rightarrow \mathbb{R}^a$ be a de-aggregating function returning an arbitrary amount of outputs a . Let $\overline{\mathcal{M}} = \langle v_i \rangle_{0 \leq i < |\overline{\mathcal{M}}|}$. Let*

$$\mathcal{M} \downarrow_b^{\text{agg}} = \langle \text{agg}_{a_i}(v_i, \dots, v_{i+a_i-1}) \rangle_{0 \leq i < b}$$

where $a_i = \lfloor \frac{|\overline{\mathcal{M}}|}{b} \rfloor$ for $i < b - 1$ and $a_i = \lfloor \frac{|\overline{\mathcal{M}}|}{b} \rfloor + (|\overline{\mathcal{M}}| \bmod b)$ for $i = b - 1$. Let

$$\langle w_i \rangle_{0 \leq i < b} \uparrow_b^{\text{deagg}} = \text{deagg}_{a_0}(w_0) \# \dots \# \text{deagg}_{a_{b-1}}(w_{b-1})$$

where a_i is defined as above and $\#$ is concatenation. We define $\mathcal{N} : \mathbb{R}^b \rightarrow \mathbb{R}^b$ for a fixed b . We then set:

$$\overline{\mathcal{O}} = \mathcal{N}(\mathcal{M} \downarrow_b^{\text{agg}}) \uparrow_b^{\text{deagg}}$$

We use $\overline{\mathcal{O}}$ to define $\mathcal{O} = \mathcal{N} \triangleleft_{\text{agg}} \mathcal{M}$ given fixed functions agg, deagg .

For our experiments, we use the average for aggregation

$$\text{agg}_a(v_0, \dots, v_{a-1}) = \sum_{i=0}^{a-1} \frac{v_i}{a}$$

and use a trivial de-aggregation function as defined by:

$$\text{deagg}_a(w) = \underbrace{(w, \dots, w)}_{a \text{ times}}$$

Intuitively, the aggregating reduction simply reduces the amount of weight parameters to a fixed amount b by aggregating sub-lists of the weight list into single values. Those single values are then passed to the network and its output values are copied to all previously aggregated weights. A lot of different aggregation and de-aggregation functions could be thought of, however, early tests with variants introducing more randomness or different topologies showed no differences in results. Thus, we focus on the simple instantiation of the aggregation reduction as given above.

Reduction 3 (RNN). We define $\mathcal{N} : \mathbb{R} \times \mathbb{R}^H \rightarrow \mathbb{R} \times \mathbb{R}^H$ as a recurrent neural network with a hidden state $h \in \mathbb{R}^H$ for some $H \in \mathbb{N}$. Let $\overline{\mathcal{M}} = \langle v_i \rangle_{0 \leq i < |\overline{\mathcal{M}}|}$. We then set

$$\overline{\mathcal{O}} = \langle w_i \rangle_{0 \leq i < |\overline{\mathcal{O}}|}$$

where w_i is given via

$$\mathcal{N}(v_i, h_i) = (w_i, h_{i+1})$$

where $h_0 = \mathbf{0}$. We use $\overline{\mathcal{O}}$ to define $\mathcal{O} = \mathcal{N} \triangleleft_{rnn} \mathcal{M}$.

Since recurrent neural networks are able to process input sequences of arbitrary length, the RNN reduction technically just needs to define \mathcal{N} as a recurrent neural network and simply apply it to the weights of another network. Even though this reduction appears most simple and natural, the explosion of gradients within larger recurrent neural networks means that they are very prone to diverge to very large output values if not sufficiently controlled. We reckon that an extension to recurrent neural networks (making them accessible to self-replication) should be possible, however, since vanilla recurrent neural networks are not so fit for self-replication, we refer this extension to future work.

We can use these several types of reduction to build a mathematical model of self-replication in neural networks.

Definition 1 (Self-Application). Given a neural network \mathcal{N} . Let \triangleleft be a suitable reduction. We call the neural network $\mathcal{N}' = \mathcal{N} \triangleleft \mathcal{N}$ the self-application of \mathcal{N} .

Definition 2 (Fixpoint, Self-Replication). Given a neural network \mathcal{N} . Let \triangleleft be a suitable reduction. We call \mathcal{N} a fixpoint with respect to \triangleleft iff $\mathcal{N} = \mathcal{N} \triangleleft \mathcal{N}$, i.e., iff \mathcal{N} is its own self-application. We also say that \mathcal{N} is able to self-replicate.

Since network weights are real-valued and are the result of many computations, checking for the equality $\mathcal{N} = \mathcal{N} \triangleleft \mathcal{N}$ is not entirely trivial. We thus relax the fixpoint property a bit.

Definition 3 (ε -Fixpoint). Given a neural network \mathcal{N} with weights $\overline{\mathcal{N}} = \langle v_i \rangle_{0 \leq i < |\overline{\mathcal{N}}|}$. Let \triangleleft be a suitable reduction. Let $\varepsilon \in \mathbb{R}$ be the error margin of the fixpoint property. Let $\mathcal{N}' = \mathcal{N} \triangleleft \mathcal{N}$ be the self-application of \mathcal{N} with weights $\overline{\mathcal{N}'} = \langle w_i \rangle_{0 \leq i < |\overline{\mathcal{N}'}|}$. We call \mathcal{N} an ε -fixpoint or a fixpoint up to ε iff for all i it holds that $|w_i - v_i| < \varepsilon$.

Training

As stated above, neural networks are commonly used in conjunction with backpropagation to adjust their weights to a desired configuration. We assume that we have a set of input vectors $\mathbf{x}_0, \dots, \mathbf{x}_n$ and a corresponding set of desired output vectors $\mathbf{y}_0, \dots, \mathbf{y}_n$. We want our neural network \mathcal{N} to represent the relation between these sets. The loss for a single sample $(\mathbf{x}_i, \mathbf{y}_i)$ is defined as $|\mathcal{N}(\mathbf{x}_i) - \mathbf{y}_i|$. Minimizing the loss of a neural network is called *training*. We use the SGD optimizer to apply gradient updates or rather weight changes to minimize the loss for given a given sample $(\mathbf{x}_i, \mathbf{y}_i)$, which

results in an updated network $\mathcal{N}' = \mathcal{N} \Leftarrow (\mathbf{x}_i, \mathbf{y}_i)$. We call \Leftarrow the training operator. For sets of sample points $\mathbf{x} = \mathbf{x}_0, \dots, \mathbf{x}_n$ and $\mathbf{y} = \mathbf{y}_0, \dots, \mathbf{y}_n$, we also write $\mathcal{N} \Leftarrow \mathbf{x}, \mathbf{y}$ as shorthand for $\mathcal{N} \Leftarrow (\mathbf{x}_0, \mathbf{y}_0) \Leftarrow \dots \Leftarrow (\mathbf{x}_n, \mathbf{y}_n)$.

We argue that training neural networks is another natural way of evolving them (as is application). Thus, we also want to train a neural network with other neural networks as input and output data. Of course, we again need to use reduction on said other neural networks. In short we write:

Reduction 4 (Weightwise Training). Given neural networks \mathcal{M}, \mathcal{N} with $\overline{\mathcal{M}} = \langle v_i \rangle_{0 \leq i \leq n}$ for some n . We write $\mathcal{N}' = \mathcal{N} \Leftarrow_{ww} \mathcal{M}$ iff

$$\mathcal{N}' = \mathcal{N} \Leftarrow \langle (v_i, l(i), c(i), p(i)) \rangle_{0 \leq i \leq n}, \langle (v_i) \rangle_{0 \leq i \leq n}$$

where l, c, p are defined as in Reduction 1.

Reduction 5 (Aggregating Training). Given neural networks \mathcal{M}, \mathcal{N} . Given a suitable aggregator function agg and aggregated size b . We write $\mathcal{N}' = \mathcal{N} \Leftarrow_{agg} \mathcal{M}$ iff

$$\mathcal{N}' = \mathcal{N} \Leftarrow (\mathcal{M} \downarrow_b^{\text{agg}}, \mathcal{M} \downarrow_b^{\text{agg}})$$

where the \downarrow operation is defined as in Reduction 2.

Reduction 6 (Recurrent Training). Given neural networks \mathcal{M}, \mathcal{N} . We write $\mathcal{N}' = \mathcal{N} \Leftarrow_{rnn} \mathcal{M}$ iff

$$\mathcal{N}' = \mathcal{N} \Leftarrow (\overline{\mathcal{M}}, \overline{\mathcal{M}})$$

where \mathcal{N} is trained on a sequence $\overline{\mathcal{M}}$ by being applied one by one recurrently.

Intuitively, these training reductions transform the input net \mathcal{M} to a smaller representation (as do the application reductions, cf. Reductions 1–3) and then train the network \mathcal{N} to accurately reproduce that representation.

Note that usually, when training a neural network, we derive training samples from a large data set or generate them automatically. However, we can use these training reductions to define the notion of self-training:

Definition 4 (Self-Training). Given a neural network \mathcal{N} . Let \Leftarrow be a suitable training reduction. We call the network $\mathcal{N}' = \mathcal{N} \Leftarrow \mathcal{N}$ the result of self-training \mathcal{N} .

We can apply self-training for many consecutive steps, however, in contrast to usual training in neural networks, the samples made available for training only depend on the network's own weights and introduce no randomness or additional coverage of the search space beyond their own (mostly pre-determined) evolution via self-training.

Experiments

We define three types of experiments, which test the two distinct approaches to self-replication based on application of neural networks to other neural networks and training using backpropagation on self-generated limited training points, respectively. Lastly, we show a strong connection between both approaches.

Note that for the sake of simplicity, we fixed all network architectures in the following experiments to only include

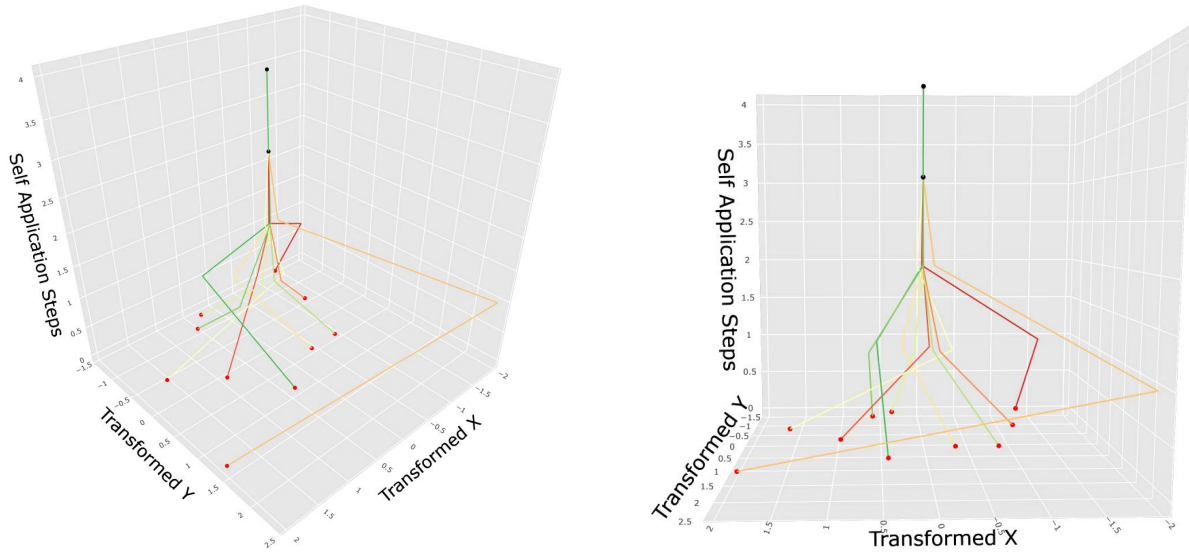


Figure 1: 10 independent runs of **self-application** with respect to the aggregating reduction \triangleleft_{agg} . 10 neural networks $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}^4$ with two hidden layers with two cells each were initialized randomly and then subjected to 4 self-applications each. The figure shows two perspectives on the same three-dimensional graph. The 20 weights in total per network were visualized in a two-dimensional space based on the transformed bases X and Y derived via PCA. All networks converge on $(X=0, Y=0)$, which corresponds to the weight vector $\mathbf{0}$.

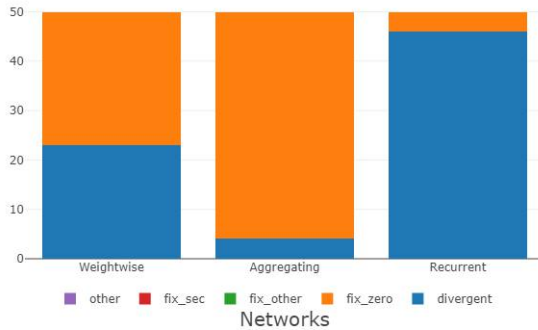


Figure 2: 50 independent runs of **self-application** each with respect to all three different types of reduction. We show an analysis of the networks (which were initialized randomly) after 100 steps of self-application.

two hidden layers with two cells each. Although evaluations were run with various activation functions, all plots show linear activation since we observed no qualitative difference between various activations. Similarly, bias was set to 0 in all plotted instances.

Self-Application

When subjecting a randomly initialized neural network \mathcal{N} to repeated self-application with respect to the weightwise reduction \triangleleft_{ww} , the weight vector $\overline{\mathcal{N}}$ tends to converge to the all-zero vector $\mathbf{0} = \langle 0 \rangle_{|\mathcal{M}|}$. This was already indicated by Chang and Lipson (2018) for a very similar reduction

approach. This effect probably stems from a phenomenon observed by Schoenholz et al. (2017): Randomly initialized neural networks tend to map their inputs to output values closer to $\mathbf{0}$. Figure 1 shows that the same effect also occurs for the aggregating reduction \triangleleft_{agg} . It shows the journey of several neural networks through the space of weight vectors.¹ Very few steps of self-application suffice to draw all neural networks to the coordinates $(X=0, Y=0)$, which in fact correspond to the weight vector $\mathbf{0}$.

The same plot for the weightwise reduction \triangleleft_{ww} looks rather similar. Figure 2 shows the resulting networks after several steps of self-application. Here, we discern five observations: A neural network \mathcal{N} is (a) *divergent* iff at any point in time any of its weights assumed the value ∞ or $-\infty$. Once this has happened, there is no returning from it. If the network assumes (b) the ε -fixpoint given by the weight vector $\mathbf{0}$, i.e., all its weights are sufficiently close to 0, we call the network a *zero fixpoint*. Note that for all experiments we set $\varepsilon = 10^{-5}$. If the network's weights resemble (c) any other ε -fixpoint we call it a non-zero, non-trivial or simply *other fixpoint*. At this stage, we also checked for (d) *second-order fixpoints*, i.e., networks \mathcal{N} fulfilling the weaker property $\mathcal{N} = \mathcal{N} \triangleleft \mathcal{N} \triangleleft \mathcal{N}$. However, we never found any such networks. Anything else falls into the category (e) *other*.

¹To be able to plot highly-dimensional weight vectors on paper, we derive the two principle components of the observed weight vectors using standard principle component analysis (PCA) and plot the weight vector as a point in that two-dimensional space. We use this technique for all such figures.

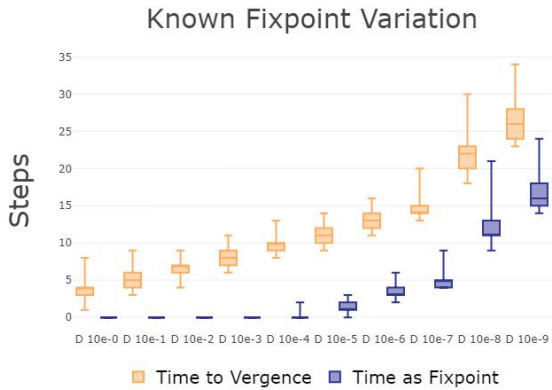


Figure 3: The robustness of a known 10^{-5} -fixpoint with respect to the weightwise reduction \triangleleft_{ww} . The x-axis shows the range of noise that the known fixpoint’s weights were subjected to. The y-axis shows for how many steps of self-application the network was still regarded as a 10^{-5} -fixpoint (purple) and after how many steps of self-application the network was regarded as diverged (orange).

Note that Figure 2 shows that no non-zero fixpoints are found for any reduction and that recurrent neural nets are most prone to diverge during repeated self-application.

We also checked for the chance to just randomly generate a neural network which happens to be a fixpoint. However, among 100,000 randomly generated nets for each type of reduction, we did not find a single fixpoint. Thus, we can clearly attribute the attraction towards $\mathbf{0}$ to self-application.

For the weightwise reduction \triangleleft_{ww} , it is rather easy to construct a non-zero fixpoint by hand: For a network \mathcal{I} , we set all leftmost weights per layer to 1 and all other weights to 0, thus implementing the identity function on the inputs of \mathcal{I} , which clearly fulfills the fixpoint property. This allows us to test if the non-zero fixpoints form an attractor in the weight space like the zero fixpoint does: We added small amounts of noise to all weights of \mathcal{I} and then subjected the resulting network \mathcal{J} to several steps of self-application, checking if $\overline{\mathcal{J}}$ would remain stable around $\overline{\mathcal{I}}$ or “verge”, i.e., either converge towards $\mathbf{0}$ or diverge towards infinite weights. However, even adding just at most 10^{-9} noise to each weight eventually caused all networks to “verge”. Figure 3 shows the experiment for various amounts of noise. Adding less noise unsurprisingly causes the network to longer fulfill the ε -fixpoint property and to “verge” later, which possibly means that the network fulfills the fixpoint property again when converging to $\mathbf{0}$ (but we did not count that).

Thus, while self-application on its own shows a stable intent to approach the fixpoint $\mathbf{0}$, it does not seem capable of creating any other fixpoints.

Self-Training

Subjecting randomly generated neural networks to self-training with respect to the aggregating training reduction \Leftarrow_{agg} yields results as shown in Figure 4. All networks evolve for a few steps of self-training, then their weights remain constant. Note that each network approaches a different point in the weight space. Most interestingly, these points are fixpoints, even though we only apply self-training and fixpoints are defined using self-application. Moreover, all of these fixpoints are non-zero.

Figure 5 shows a detailed analysis for all types of reduction: While recurrent neural networks still tend to diverge a lot, aggregating networks converge towards weights that do not represent a fixpoint. However, the weightwise networks converge to non-trivial fixpoints with utmost reliability.

In order to elaborate on the opportunities of interaction between self-application and self-training, we construct an experiment where the two appear in alternation. The results are shown in Figure 6: While aggregating networks reach the zero fixpoint so fast via self-application that self-training is not able to add anything to that, weightwise networks need about 200–300 steps of self-training between each self-application to converge to fixpoints as reliably.

Soups

As we have discussed several means of neural networks interacting with themselves, it seems a reasonable next step to open up these interactions and build a population of mutually interacting networks. A suitable combination of a population of individuals and various interactions is called *soup* and works like an artificial chemistry system (cf. Dittrich et al. (2001)). This means that a soup evolves over a fixed amount of epochs. At every epoch, several different interaction operators can be applied to networks in the population with a certain chance, resulting in new networks and thus a changed population.

Interaction 1 (Self-Train). *Applied to every single network \mathcal{N} for an amount of steps A , self-training substitutes its weights with $\mathcal{N}' = \mathcal{N} \underbrace{\Leftarrow \mathcal{N} \dots \Leftarrow \mathcal{N}}_{A \text{ times}}$.*

Interaction 2 (Attack). *Applied to two random networks \mathcal{M}, \mathcal{N} at a chance α , attacking substitutes the weights of the attacked network \mathcal{M} with the weights given via $\mathcal{M}' = \mathcal{N} \triangleleft \mathcal{M}$.*

Intuitively, attacking applies the function represented by the network \mathcal{N} to another network \mathcal{M} . Self-training remains basically unchanged from the non-soup scenario and provides a background evolution to every network in the population, even when it is not involved in any attack.

Figure 7 shows the evolution of a soup employing self-training and attacking. The networks start out randomly placed in the weight space and self-train towards fixpoints in the beginning. The big jumps in the networks’ trajectories

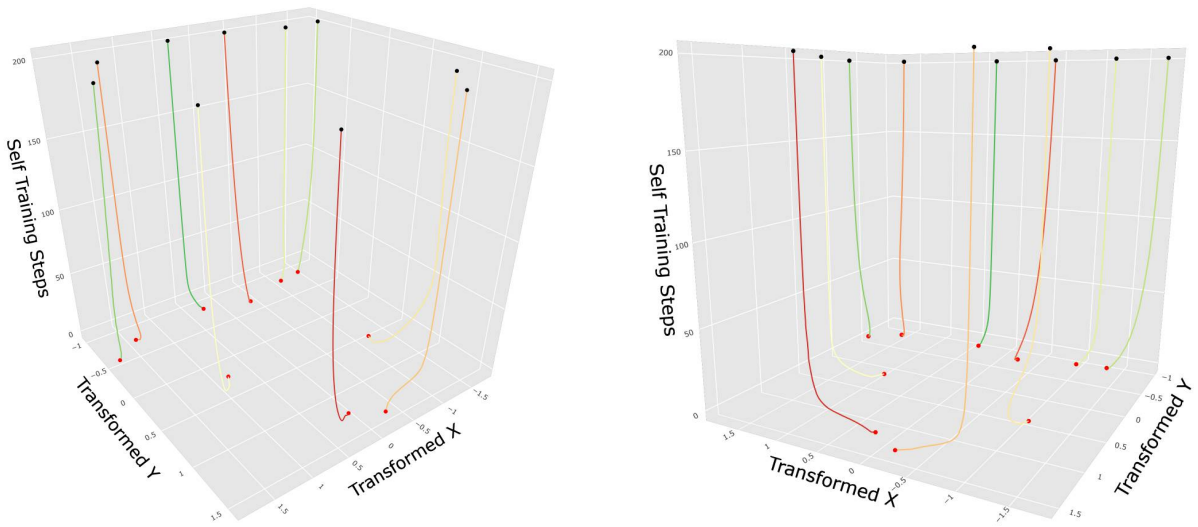


Figure 4: 10 independent runs of **self-training** with respect to the aggregating training reduction \Leftarrow_{ww} . 10 neural networks $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}$ with two hidden layers with two cells each were initialized randomly and then subjected to 200 steps of self-training each. The figure shows two perspectives on the same three-dimensional graph. The 20 weights in total per network were visualized in a two-dimensional space based on the transformed bases X and Y derived via PCA. All networks converge to different fixpoints with non-zero weights.

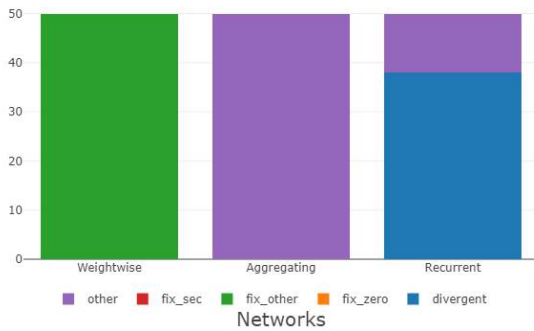


Figure 5: 50 independent runs of **self-training** each with respect to all three different types of reduction. We show an analysis of the networks (which were initialized randomly) after 1000 steps of self-training.

stem from being attacked by other networks; self-training then leads them to new fixpoints. Note that as self-training causes the networks to converge towards fixpoints, the impact of near-fixpoint networks' attacks becomes less and less. Most interestingly though, almost all attacks seem to drive the attacked networks towards the main cluster of the soup, where most networks gather in the end. This not only shows emergent behavior as the networks form a group as a cluster of fixpoints somewhere in the weight space (neither at the center of mass from the initial population nor anywhere near $\mathbf{0}$), but it can be also interpreted as a clear instance of (self-)replication within the networks of this soup.

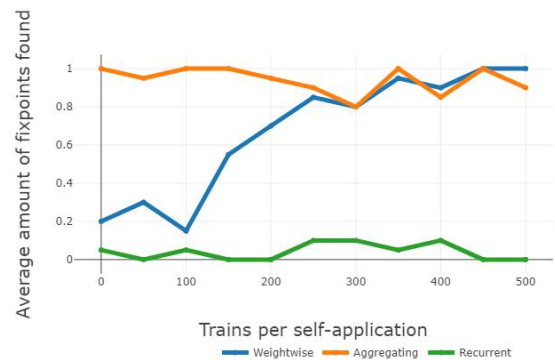


Figure 6: Evaluation of a mixed setting of self-application and self-training. For each type of reduction, 20 neural networks were generated at random and then subjected to 4 steps of self-application. In between those steps, 0, 50, ..., 500 steps of self-training were executed (see x-axis). The y-axis shows the average ratio of fixpoints (both zero and non-zero) found out of all runs, where a value of 1 means that all runs resulted in a fixpoint.

In Figure 8, we further evaluate the impact of parameter A in Interaction 1 for both weightwise and aggregating neural networks. (As recurrent networks already did not show sufficient compatibility with application, we omit these results.) More self-training manages to stabilize the weightwise networks' ability to find non-zero fixpoints. Still, even in a soup setting, aggregating networks converge to $\mathbf{0}$ to a distinctive degree.

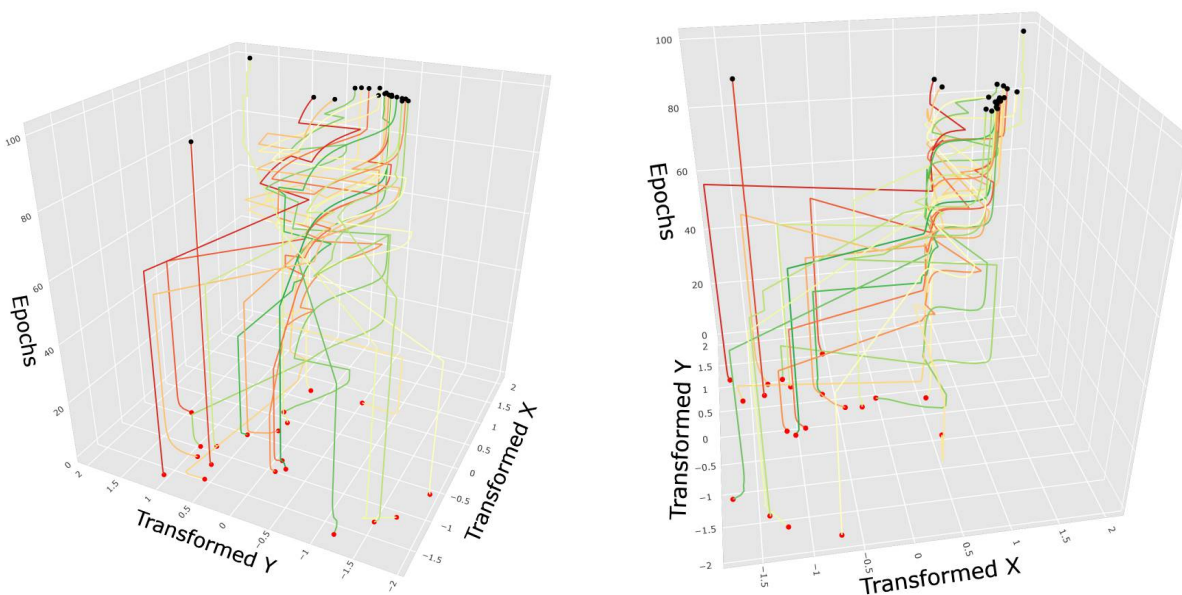


Figure 7: Run of one soup consisting of 20 neural networks using the weightwise reductions \triangleleft_{ww} and \Leftarrow_{ww} . The 20 neural networks $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}$ with two hidden layers with two cells each were initialized randomly and then evolved for 100 epoch. Per epoch, every network had a chance of 0.1 to attack another network and was subjected to 30 steps of self-training. This setup allowed for emergent behavior of the network forming a cluster at a region for all non-zero fixpoints. The figure shows two perspectives on the same three-dimensional graph. The 20 weights in total per network were visualized in a two-dimensional space based on the transformed bases X and Y derived via PCA.

Related Work

There is some research in generating neural networks using other neural networks (cf. Schmidhuber (1992); Stanley et al. (2009); Deutsch (2018), e.g.). However, without any suitable reduction operations, these approaches cannot be used to produce self-replicating structures.

Our results on self-application agree with Chang and Lipsion (2018) on the weightwise reduction. We extended the experiments with several means of reduction and managed to find non-trivial, non-zero fixpoints up to a very low error ε by introducing our weightwise reduction in combination with our notion of self-training. We augmented the approach by studying the combination of self-application and self-training. However, the inclusion of auxiliary fitness functions has not been considered in our work.

The idea to generate fixpoints via repeated self-application is based on Fontana and Buss (1996), who showed the emergence of fixpoints from having random expressions in the λ -calculus interact. They, too, construct an artificial chemistry system based on their functional abstraction and see complex structures of fixpoints arise. Sadly, we did not observe higher-order fixpoints as they did for λ -expressions. Possible connections between λ -fixpoints or larger organizational structures in general and fixpoints in neural networks may still be explored (Larkin and Stocks (2004)).

In general, a vast amount of research is dedicated to artificial chemistry systems, utilizing very different representations for the particles in the soup: Dittrich et al. (2001) and Matsumaru et al. (2005) provide excellent overviews, to which we refer for the sake of brevity.

Conclusion

We have presented various reduction operations without any claim of completeness. Interesting reduction possibilities like extracting the main frequencies of the weight vector using a Fourier transformation are still to be tested to full extent. Most importantly, all settings, architectures and parameters of the neural networks we constructed still allow for more thorough exploration and evaluation in future work.

We have also performed some exploration of the distribution of fixpoints within the weight space by generating lots of non-trivial fixpoints using our setup of self-training. Especially discovering some kind of measurement of how rare fixpoints actually are and if they can occur in all regions of the weight space would be helpful.

We think that perhaps the most interesting contributions are the distinct behaviors observed in the soup made of neural networks (Figure 7). While we evaluated some parameters, there exist many different ways to evolve such a soup and many different interactions whose effects are yet to be explored. Early results on an interaction called *learn*, which

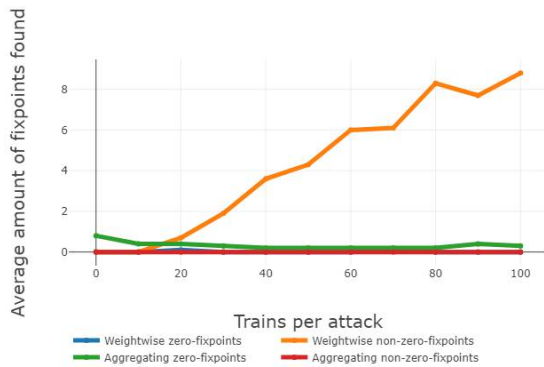


Figure 8: Evaluation of the impact of the number of trains per epoch on a soup consisting of 10 neural networks using the weightwise reduction \langle_{ww} or the aggregating reduction \langle_{agg} . Averaged over 10 runs. Per epoch, every network had a chance of 0.1 to attack another network and was subjected to a fixed amount of steps of self-training (see the x-axis). The y-axis shows the amount of (zero or non-zero) fixpoints present in the final population of the soup.

substitutes the weights of the learning network \mathcal{M} with the weights given via $\mathcal{M}' = \mathcal{M} \rightsquigarrow \mathcal{N}$ look most promising but were left out for brevity.

Eventually, we think that the dynamics of a soup might open up neural networks to a new kind of learning by not applying a goal function (and its respective loss) directly but by simply guiding a soup a certain way, perhaps achieving more diversity and robustness in the solutions reached (cf. Prokopenko (2013); Gabor et al. (2018), e.g.).

References

- Bäck, T., Hammel, U., and Schwefel, H.-P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE transactions on Evolutionary Computation*, 1(1):3–17.
- Box, G. E. (1957). Evolutionary operation: A method for increasing industrial productivity. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 6(2):81–101.
- Chang, O. and Lipson, H. (2018). Neural network quine. In *Artificial Life Conference Proceedings*. MIT Press.
- Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Church, A. (1932). A set of postulates for the foundation of logic. *Annals of mathematics*, pages 346–366.
- Conway, J. (1970). The game of life. *Scientific American*, 223(4):4.
- Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press, USA.
- Deutsch, L. (2018). Generating neural networks with neural networks. *arXiv preprint arXiv:1801.01952*.
- Dittrich, P. and Banzhaf, W. (1998). Self-evolution in a constructive binary string system. *Artificial Life*, 4(2):203–220.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries review. *Artificial life*, 7(3):225–275.
- Dorigo, M. and Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation (CEC99)*, volume 2. IEEE.
- Fontana, W. and Buss, L. (1996). The barrier of objects: from dynamical systems to bounded organizations.
- Gabor, T., Belzner, L., Phan, T., and Schmid, K. (2018). Preparing for the unexpected: Diversity improves planning resilience in evolutionary algorithms. In *2018 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29.
- Koza, J. R. (1994). Spontaneous emergence of self-replicating and evolutionarily self-improving computer programs. *Artificial life III*, 17:225–262.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.
- Larkin, J. and Stocks, P. (2004). Self-replicating expressions in the lambda calculus. In *Proceedings of the 27th Australasian conference on Computer science-Volume 26*, pages 167–173. Australian Computer Society, Inc.
- Matsumaru, N., Centler, F., di Fenizio, P. S., and Dittrich, P. (2005). Chemical organization theory as a theoretical base for chemical computing. In *Proceedings of the 2005 Workshop on Unconventional Computing: From Cellular Automata to Wetware*, pages 75–88. Luniver Press.
- Minsky, M. and Papert, S. (1972). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.
- Prokopenko, M. (2013). *Guided self-organization: Inception*, volume 9. Springer Science & Business Media.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Schmidhuber, J. (1992). Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.
- Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2017). A correspondence between random neural networks and statistical field theory. *arXiv preprint arXiv:1710.06570*.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.
- Stanley, K. O., D’Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212.
- Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.

The role of ambient noise in the evolution of robust mental representations in cognitive systems

Douglas Kirkpatrick^{1,2,4}, and Arend Hintze^{3,1,2,4}

¹ Department of Computer Science and Engineering

² BEACON Center for the Study of Evolution in Action

³ Department of Integrative Biology ⁴ Michigan State University

hintze@msu.edu

Abstract

Natural environments are full of ambient noise; nevertheless, natural cognitive systems deal greatly with uncertainty but also have ways to suppress or ignore noise unrelated to the task at hand. For most intelligent tasks, experiences and observations have to be committed to memory and these representations of reality inform future decisions. We know that deep learned artificial neural networks (ANNs) often struggle with the formation of representations. This struggle may be due to the ANN's fully interconnected, layered architecture. This forces information to be propagated over the entire system, which is different from natural brains that instead have sparsely distributed representations. Here we show how ambient noise causes neural substrates such as recurrent ANNs and long short-term memory neural networks to evolve more representations in order to function in these noisy environments, which also greatly improves their functionality. However, these systems also tend to further smear their representations over their internal states making them more vulnerable to internal noise. We also show that Markov Brains (MBs) are mostly unaffected by ambient noise, and their representations remain sparsely distributed (i.e. not smeared). This suggests that ambient noise helps to increase the amount of representations formed in neural networks, but also requires us to find additional solutions to prevent smearing of said representations.

Introduction

In all but the simplest of forms, artificial and natural minds need to have experiences, remember them, and use those memories to inform future actions. Previous work has defined the term “representations” to be the information an agent has about the environment that is not present in its sensors (Marstaller et al., 2013). We developed a measure to quantify these mental representations, referred to as R . Further work expanded on this method of quantifying representations and showed that R can even be used to augment a genetic algorithm's (GA's) performance to find better solutions in a shorter amount of time than an unaugmented GA (Schossau et al., 2015). In general, the amount of representation increases over the course of evolution and allows neural substrates ¹ to deal better with their environments –

¹e.g. MBs or ANNs, broadly referred to as brains or agents

clearly, knowing something about the world they live in is beneficial.

By measuring which components of a cognitive agent have information about specific aspects of the world, we can even pinpoint where representations are stored (Marstaller et al., 2013). Specifically, the measure of R (see Figure 2) is applied to every node and every concept separately, resulting in a matrix (M) of measurements. This method further allows us to determine how distributed (smeared) or localized these representations are (Hintze et al., 2018) (see Equation 2 for a quantification of the smearedness of matrix M). We also found that less smeared representations are more robust against sensor noise.

All these findings support the notion that representations are crucial for intelligence, and we can use evolutionary processes to create cognitive agents that take advantage of them. We also know that in deep learning of artificial neural networks noise is often used to increase the size or quality of the training data set (Brown et al., 2003; Lauzon, 2012), but it also directly can improve training when applied for example to the gradient descent back propagation (Neelakantan et al., 2015). Similarly, *dropout* is another method to improve the deep learning process due to randomness (Srivastava et al., 2014). However, noise within the input training set has also been identified to limit the performance of ANNs (Zhu and Wu, 2004) as it makes class discrimination difficult. In the context of these conclusions, the question that we seek to investigate is how we can improve the amount and quality of representations that a cognitive system has about its environment and what the role of sensor (ambient) noise plays in the evolutionary adaptive process.

In order to test these questions, we evolve different cognitive systems in the presence of different levels and forms of sensor noise, and study their performance, their ability to form representations, and their robustness to different types of noise. From previous experiments, we know that different systems behave very differently during evolutionary adaptation. Markov Brains (Marstaller et al., 2013; Hintze et al., 2017), for example, first have to evolve structures to retain information before they can evolve to take advantage

of them. Recurrent neural networks on the other hand retain information due to their connectedness and need to only evolve to use the information. At the same time, we know that Markov Brains have distinctively discrete representations, while neural networks tend to smear these representations across their recurrent nodes. As a consequence, recurrent neural networks in the long run lose their early evolutionary advantage of getting representations for free when compared to Markov Brains that need to also evolve mechanisms to retain information (Hintze et al., 2018). Since we also have not yet studied how representations form in other cognitive systems, here we take the opportunity and include recurrent neural networks that can evolve their topology and compare them to Markov Brains using deterministic logic gates, recurrent neural networks that use a fixed topology, and long-short term memory neural networks (LSTMs) (Hochreiter and Schmidhuber, 1997).

To study these questions of robustness and representations, the different cognitive systems are tasked to discriminate numbers (Nieder, 2018). Specifically, two different numbers are presented in sequence, and the cognitive system has to identify which of the two numbers is larger (the first or the second). This computation, however, takes the cognitive systems some time, and in this time the systems are additionally fed either noisy or static inputs. These systems cannot just avoid those noisy inputs (for example by looking away), but instead need to evolve some cognitive mechanism to filter out the noise, since the noise is always present in their sensors.

We will show that cognitive systems tend to increase the amount of representations that they have as they encounter sensor noise. They not only increase the amount of representations, but also smear those representations across their neural substrate. While being a proper response to sensor noise, it also makes them more vulnerable to internal noise of, for example, faulty components.

Material and Methods

Number Discrimination Task

For this work we adapted the Number Discrimination Task (NDT), used in biological (Nieder, 2018; Merritt et al., 2009) and psychological (Merritt and Brannon, 2013) experiments, for use *in silico*. In this task, participants have to discriminate two different amounts of items. For bees (Nieder, 2018) there were landing sites associated with different numbers of squares, and the bees were rewarded when they landed on the site associated with the correct value. For rhesus monkeys (Merritt et al., 2009) and humans (Merritt and Brannon, 2013), the reward is given when the person or monkey selects the correct value from two collections of different colored shapes presented on a screen. The basic principle *in silico* remains the same in that the agent must first look at one quantity (the number of ‘1’ signals in the input), then look at a second quantity (a second, different

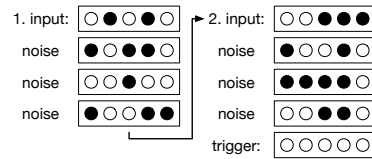


Figure 1: Example graphic of the input states for the number discrimination task. Each agent received a sequence of inputs, illustrated as black and white circles. The first and second input are used to determine the numbers to discriminate (here, 2 and 3, respectively), while the other intermittent inputs are ambient noise. Once the trigger is shown, the agent’s answer about which number is higher is evaluated.

number of ‘1’ signals in a second input), and finally make a determination of which quantity was larger or smaller.

For our experiments, the agents are given an input vector consisting of x ‘1’ signals and y ‘0’ signals, where $x + y$ is equal to 5, and are given 3 updates to process that number. Then they are given a second input vector consisting of a different amount of j 1s and k 0s, where $x \neq j$ and $j + k$ is also equal to 5. After a second set of 3 updates to process the second number, they are given a final input of 5 ‘0’s, after which they must indicate if x was greater than j . During the thinking period, the brains are either given a constant string of ‘0’s as an input vector, or with probability p each of the values in the input vector is set to a random value. The pattern of input vectors is illustrated in Figure 1.

In natural systems subjects are only tested on a handful of questions. Here we need to properly assess an agent’s ability to discriminate as part of the evolutionary process, and thus we test them on all possible input scenarios. For each number n between 0 and 5 inclusive, all permutations of n 1s and $5 - n$ 0s are tested against all permutations of the other values in $[0, 5]$. As an example, 01001 and 11000 are both valid permutations for $n = 2$, and all such permutations of $n = 2$ would be tested for all permutations of the amounts 0, 1, 3, 4, and 5. These are additionally tested with both the larger amount presented first and the smaller number second, and vice-versa. That is, 01011 as the first amount then 11111 as the second amount is tested as well as 11111 then 01011.

Evolutionary Algorithm The fitness for this task is determined by the number of times that the agent correctly identifies the larger amount (C) versus the number of times that the agent does not identify the larger amount correctly (I). The agent’s fitness is multiplied by 1.1 for every correct identification and divided by 1.1 for every incorrect identification. This results in a fitness function as shown in Equation 1.

$$W = 1.10^{(C-I)} \quad (1)$$

Once the fitness was calculated individually for the entire population, we used roulette wheel selection, which imple-

ments a fitness proportional Moran process (Moran, 1958), to determine which organisms reproduced and formed the next generation. At every reproductive event, mutations (see the following subsection for details) are applied to the offspring. The agents were evolved for 40,000 generations in populations of 100 organisms, all starting with different random genomes.

Shared Genome

Each of the brains used in this experiment was generated from the same type of genome. That genome type is a circular genome with an initial size of 5,000 sites, a maximum size of 20,000 sites, and a minimum size of 2,000 sites. A number of mutational operators were used to mutate the genomes. These were a point mutation operator that had a probability of 0.005 of changing each site, a copy insertion operator that had a per site probability of 0.00002 of copying a section of the genome with a size between 128 and 512 sites and inserting it into the genome at a randomly chosen site, and a deletion operator that had a per site probability of 0.00002 of deleting a section of the genome between 128 and 512 sites long. Brains read these genomes sequentially in order to determine their weights, computational elements, or topology when required.

Markov Brains

MBs take the form of a compact network of computational elements (called gates) that read from and write to a series of nodes. The nodes are divided up into three classes, input, output, and hidden. Input nodes take the input from the task, output nodes are used by the task to determine the agent's actions, and hidden nodes are used by the brain to store information. All of the MBs used in our experiments had 8 hidden nodes for memory. For our experiments we used two types of MBs, each using a single type of gate - Deterministic Logic or ANN. For a more detailed description see Hintze et al. (2017).

Deterministic Gates Deterministic Logic gates, as used in the experiments carried out here, read from between 1 and 4 inputs, perform a logic operation (e.g. boolean AND, boolean XOR) and write the results to between 1 and 4 outputs. If two gates write into the same node, their outputs experience a logical OR operation.

ANN Gates The ANN gates used here simulate a single-layer version of a simple feed-forward ANN (Russell et al., 2003). They read from between 1 and 4 inputs, multiply the inputs by weights stored in a table, and take the \tanh of the sum of the products to write to between 1 and 4 outputs. In case the outputs of these gates write into the same node, their outputs are added together.

The construction of Markov Brains with ANN gates technically allows for arbitrary topologies in the neural network,

and as such breaks the paradigm that ANNs must be organized in layers. The Markov Brain/ANN hybrids are much closer technically to RNNs or LSTMs but with a changing topology.

Recurrent ANNs

While the ANNs used in the ANN Gates are simple feed-forward mechanisms, more complex versions of ANNs are possible (e.g. Hochreiter and Schmidhuber (1997); Lauzon (2012)). As the presence of memory is necessary to examine representation, we elected to use an ANN augmented with recurrent nodes (i.e. memory) that was highlighted in previous work on representations (Hintze et al., 2018), referred to as an RNN. The structure of the RNN is similar to a standard ANN with one key difference. Like ANNs, the RNN is multilayer, feed-forward, and the layer to layer update is \tanh of the sum of weights times the previous layer values. The difference between ANNs and RNNs is that in RNNs the input and output layers have additional recurrent nodes that are copied from the output layer to the input layer at the end of each update. These recurrent nodes allow the RNN brains to have memory and store information about the environment. Each of the RNNs used in our experiments had 8 recurrent nodes, to allow a direct comparison to the 8 hidden nodes in the Markov Brains. The number of input and output nodes is defined by the task.

LSTM Networks

As opposed to the simple recurrence of RNNs, LSTM networks (Hochreiter and Schmidhuber, 1997) implement a more sophisticated memory model. LSTMs have been shown to perform well on tasks that require memory using Deep Learning (Schmidhuber, 2015), and thus seem ideal for studying representations. These networks have two streams of memory, (C and h) intended to replicate long and short term memory, as opposed to the simple recurring nodes of RNNs. LSTMs also use a more complicated set of equations to calculate the update values of the streams and output nodes. We modified the LSTMs to have 8 hidden nodes by expanding the memory streams, as described in previous literature (Hintze et al., 2018). The number of input and output nodes is again defined by the task.

Representations and R

Representations are the information stored in brain about environment not present in the sensors. An information-theoretic measure, R , was developed to measure representations (Marstaller et al., 2013), and is visualized in Figure 2. For this task, we defined the world states to be binary states for each number that are true if the agent is seeing that number as part of the current trial, and a binary state that is true if the first value is greater than the second. This information - which numbers the agent is seeing and the relationship between the numbers - are what we deem necessary for

the agent to solve the task.

As R is an information-theoretic measure, it is susceptible to the presence of noise (i.e. the presence of noise will affect the measurement of R). To ensure that the values of R would be comparable across different levels of ambient noise, we discarded the observations of World, Sensor, and Brain states when ambient noise was being applied to the sensor states.

We made one modification to the calculation of R , in how we dealt with the continuous hidden node values in MBs with ANN gates, RNNs, and LSTMs. To perform the entropy calculations, we used median discretization to first transform the hidden node values to bit values. This median discretization ensures that individual node states have maximum entropy to avoid accidentally introducing other sources of information into the calculation erroneously. With all brains having the same number and type of hidden states, we then calculated R .

Smearedness of Representations

While the measurement of representations is important, R measures nothing about the structure of those representations - how the representations are stored in the hidden nodes. Recent work (Hintze et al., 2018) identified a new measure regarding representations, the smearedness of representations. This measure records how much the representations are distributed across the hidden nodes of the brain. That paper had suggested two different types of smearedness - the smearedness of concepts across nodes (S_C) and the smearedness of nodes across concepts (S_N). We found a strong positive correlation between S_C and S_N (data not shown), so for the purposes of this paper we will only look at S_N .

To determine the smearedness of the representations, one must calculate atomic R - information stored about one aspect of the environment in one hidden state not including the information from the sensor. With the atomic R M_{ji} for every environmental concept j and hidden node i , we can

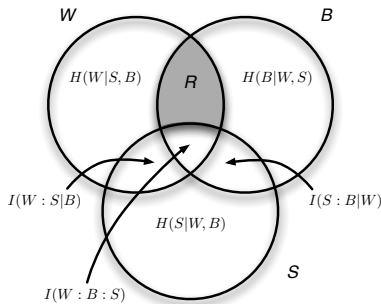


Figure 2: Venn diagram of entropies and informations for the three random variables W (world), S (sensor), and B (brain) describing all possible states the system can be in. The representation $R = H(W : B|S)$ is shaded.

then use Equation 2 to calculate S_N . To handle the continuous hidden node values in MBs with ANN gates, RNNs, and LSTMs, we used the same discretization process used in the computation of R (see the above section for details).

$$S_N = \sum_i \sum_{j>k} \min(M_{ji}, M_{ki}) \quad (2)$$

The approach used here measures the representations each node has about each concept separately. However, sets of nodes can share (encrypt) information about concepts, and *vice-versa*. One could compute all information between all subsets, which would allow for a more precise way to identify how information is smeared across nodes, and about which concepts such information is smeared. The complexity for a system of n nodes and c concepts is unfortunately 2^{n+c} which is technically impractical.

Robustness to Internal Noise

In order to determine if the agents are forming meaningful representations (i.e. representations that are useful to performance) we introduce a new measurement, referred to as robustness. This measure is designed to have a higher value when the agent is more robust to noise in its hidden states and a lower value when the agent is less robust to noise in its hidden states. Robustness to internal noise is designed to be an approximate measure of how well an agent could handle faulty memory or internal states. To calculate this robustness, we test each agent on the task while applying noise to the internal states (e.g. the hidden states in Markov Brains or the recurrent nodes in RNNs). The noise is quantified by a probability p , where p is the chance that for each hidden state that the agent has, on each update of the agent that state will be set to a random value between -1 and 1 . We test the agents with p at a range of values P between 0 and 1, and find the percentage of the trials that the agents get correct, defining the fitness function. The robustness is the sum of the percentages over all values of p (Equation 3).

$$Robustness = \sum_{p \in P} PercentCorrect_p \quad (3)$$

For our experiments, $P = [0.0, 0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.15, 0.2, 0.3, 0.4, 0.5, 1.0]$.

Experimental Setup

For each of the 4 brain types outlined above, we evolved populations on the Number Discrimination Task with varying levels of ambient noise. The noise levels tested (i.e. the probability that each sensor input becomes randomized instead of being kept at 0) were 0.0, 0.25, 0.50, 0.75, and 1.0. We ran 400 independent evolutionary experiments for each brain and noise level condition, each with a different random starting condition. At the end of evolution, we analyzed the

brains along the line of descent (Lenski et al., 2003), measuring their performance, R , the smearedness of nodes, and the robustness to internal noise.

Results

As we are dealing with a different environment than previously studied in the context of representations and R (Marstaller et al., 2013), we must first establish that the agents can evolve to perform and develop representations in this environment. We find that regardless of brain type (Markov Brain logic gates, Markov Brain ANN gates, RNN, or LSTM) and regardless of noise level (0, 0.25, 0.5, 0.75, 1.0), all brains generally evolved to perform the task (see Figure 3 left column). By the end of evolution, around 11% of the MB agents evolved with deterministic logic gates, 3.5% of the MB agents evolved with ANN gates, 35% of the RNN agents, and 4% of the LSTM agents had reached perfect performance, averaged across all noise conditions. However, it seems as if there is a specific difference between MBs (logic or ANN gates) and the two other types (RNN and LSTM) in the way evolutionary adaptation is affected by noise. While MBs experience a drop in performance when noise is introduced (0.25), they become better adapted the noisier the environments are (see Figure 3 right column). RNNs and LSTMs evolve immediately better with low noise (0.25) and struggle to become optimal the more noise they encounter.

The degree to which the different types of brains evolve representations on the other hand is affected differently by noise. The total information that MBs evolve to have about their environment seems to be unaffected by noise, while all other systems evolve to store more information the more noise they encounter (see Figure 4). As previously observed (Marstaller et al., 2013) MBs using deterministic logic gates end up having the largest amount of R initially, which might explain that additional sensor noise does not encourage them to evolve even more R . The other types of brains instead evolve to have more representations given more noise. We speculate that MBs with ANN gates, RNNs, and LSTMs need to do that in order to compensate for the noise. Intuitively, this is supported by the idea that one needs to remember better if one's sensors are flooded with noise. While this is a conjecture at this point, we will further investigate this question, but we need to first ask to what degree the representations that these systems evolve are smeared or distinct.

To measure the smearedness of representations, one first identifies which component of the brain has how much information about the concepts of the world. This is comparable with the idea of “grandmother neurons”, where specific neurons represent specific memories, here of the grandmother (for a more critical discussion see Quiroga et al. (2013)). Smearedness does not assume the existence of grandmother neurons, but only quantifies the degree to which represen-

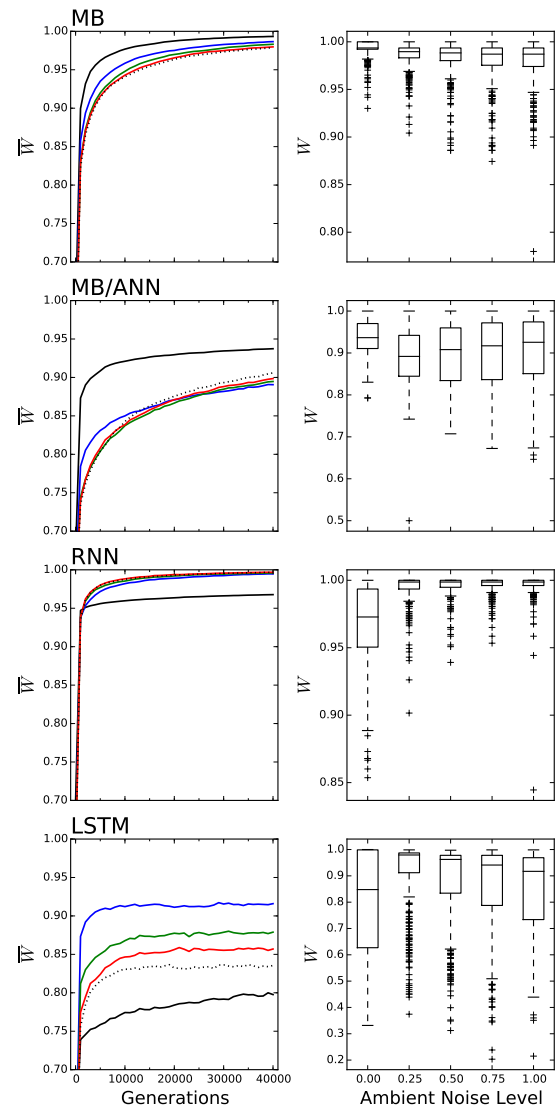


Figure 3: **Performance Over Time.** For each type of brain (MB, MB using ANN gates, RNN, and LSTM see labels) the average performance on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.5 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions.

tations are either stored in individual nodes (or neurons) or distributed over many. The less smeared R is the sparser and more distinct representations are. We find that, except for MBs using deterministic logic gates, all other brain types evolve to have more smeared representations the more noise they encounter (see Figure 5).

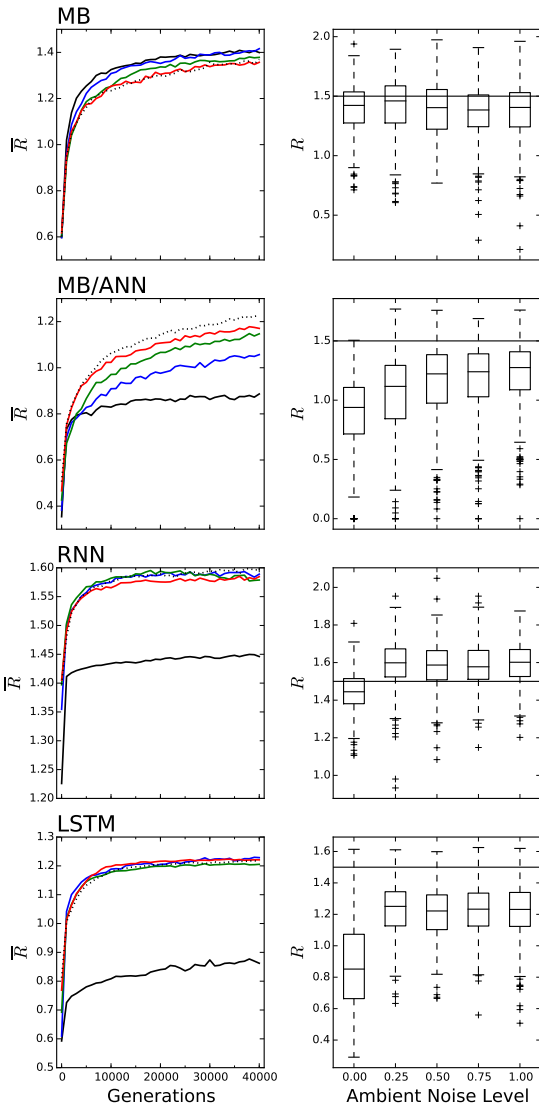


Figure 4: **R Over Time**. For each type of brain the average value of R on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.50 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions. There is a line in every subgraph in the right column at the height 1.5 to allow for a comparison between brain types.

We further find, that the degree of this smearing is correlated to the amount of R each agent evolved (see Figure 6). Those agents that evolve to have a lot of information about the environment, also smear these representations.

As mentioned before, we assume that brains evolve to

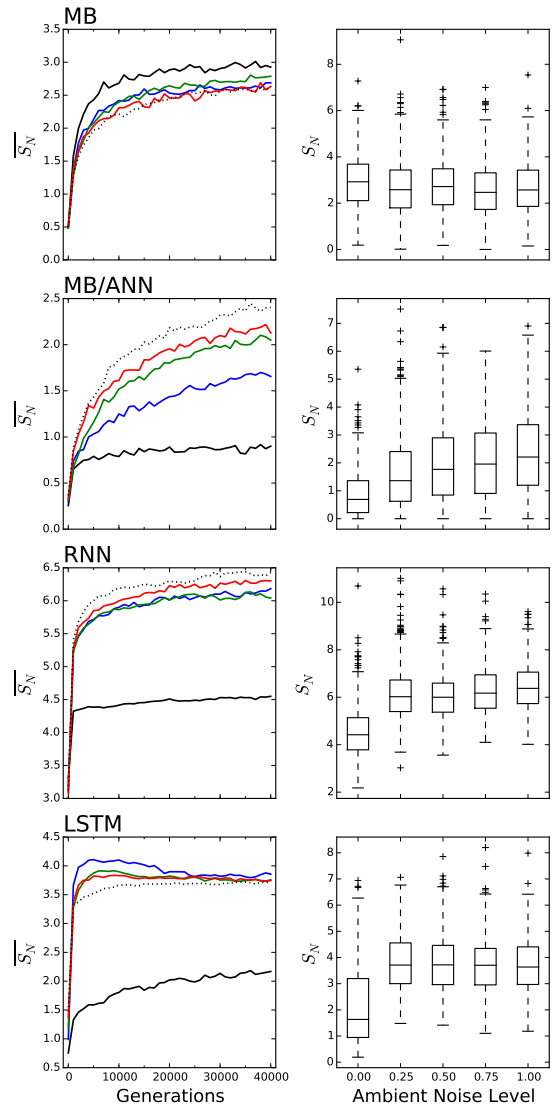


Figure 5: **Smeariness of Nodes Over Time**. For each type of brain the average value of S_N on the line of descent for all replicate experiments (left column). The black solid line is 0.0 noise, the blue line is 0.25 noise, the green line is 0.50 noise, the red line is 0.75 noise, and the black dotted line is 1.0 noise. The right column shows the distribution of performances including the 25% and 75% confidence intervals and their outliers for all brains under all experimental noise conditions.

have more representations as a functional response to more noise in the environment, and that the smearing of these representations over their hidden states is functional as well. If this is the case, the representations that get smeared need to contribute to performance, otherwise it could be coincidental or a measurement artifact. To test this, we determine the robustness to internal noise and correlate this robustness to

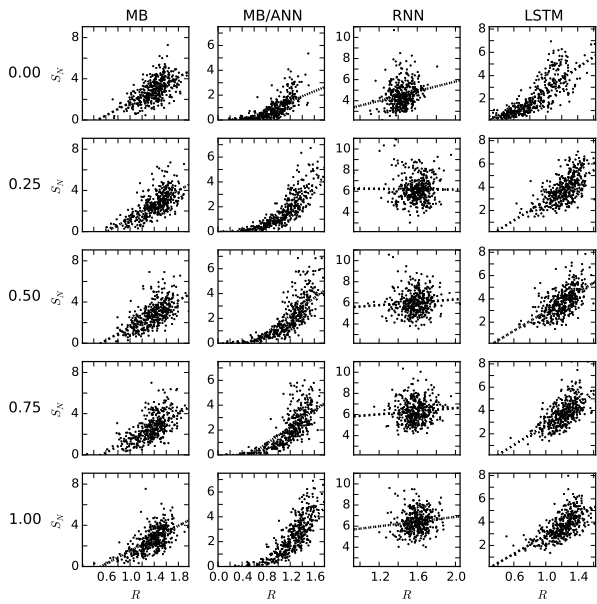


Figure 6: *R vs Smeariness of Nodes*. Each subplot is the scatterplot distribution of the value of S_N versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

R as well as the smeariness S_N . We find that in the majority of cases, the more representations the brains evolve the more they are effected by internal noise (see Figure 7). Only RNNs are somewhat unaffected and do not show a strong correlation between R and robustness to internal noise.

The same correlation can be found between the smeariness of representations and robustness to internal noise (see Figure 8 for the correlation of S_N and robustness). This shows that as hypothesized the increase of representations and their smearing over internal nodes is indeed functional and not just coincidental or a measuring artifact.

Discussion

Here we investigated the role that sensor noise has on the ability of different cognitive systems to evolve representations. Specifically, MBs with deterministic logic gates and ANN gates, RNNs, and LSTMs were used as examples of cognitive systems. We already knew that the smearing of representations, as happens in RNNs and LSTMs, makes these systems more vulnerable to internal noise. The hope was to improve our ability to evolve these systems by introducing sensor noise. While adding noise improves the performance of RNNs and LSTMs, MBs seem to be unaffected regardless of using logic gates or ANN gates. However, except for RNNs, those systems that improve their perfor-

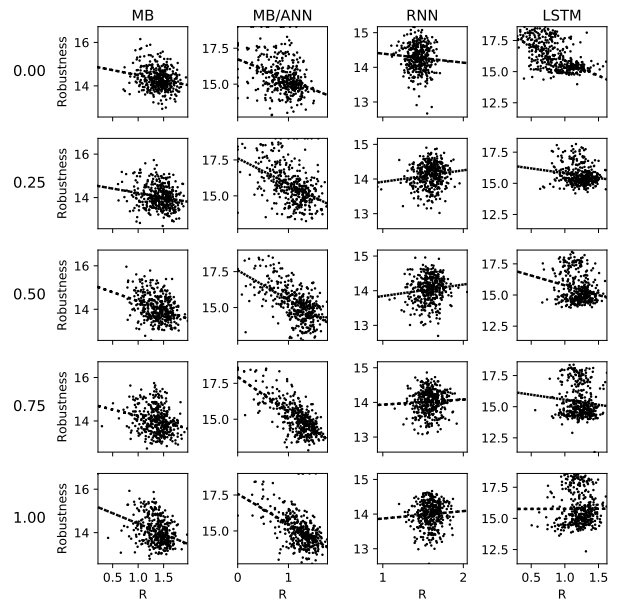


Figure 7: *R vs. Robustness*. Each subplot is the scatterplot distribution of the value of R versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

mance due to noise increase the amount of information that they have about the environment (R) and also smear these representations over their hidden states. Furthermore, these smeared representations are functional since they make the networks more susceptible to internal noise. Even though we did not test our hypotheses on more systems, we assume that our results can generalize to other types of computational cognitive models. Similarly, we only used the number discrimination task, and it is possible that other tasks respond differently. Both points suggest to test the effect of sensor noise in other cognitive systems and on other cognitive tasks. Here we also focused on evolutionary adaptation, and in most other cases RNNs and LSTMs are not trained by a genetic algorithm but instead gradient descent deep learning. This strongly suggests that we need to explore how these systems develop representations and to what degree the representations are smeared in the deep learning context.

Conclusion

We found our initial intuition confirmed that the evolution of cognitive systems can be improved by using sensor noise, more so with RNNs and LSTMs than with Markov Brains. However, this improvement comes with a price which one might only pay under certain conditions. Using sensor noise to improve a neural networks ability to evolve will only help

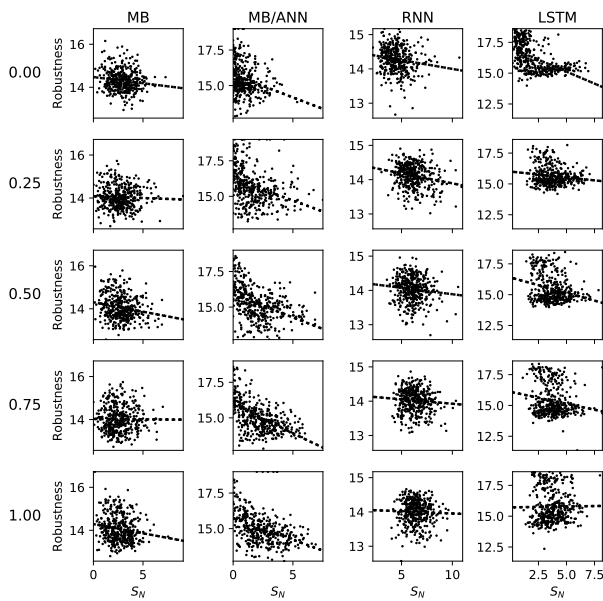


Figure 8: **Smearedness of Nodes vs. Robustness.** Each subplot is the scatterplot distribution of the value of S_N versus robustness to hidden noise, measured at the end of each evolutionary run. The dotted line in each subplot is the line of best fit of the data for each condition. Each column corresponds to a specific brain type, and each row corresponds to a given ambient noise level

if internal noise can be prevented.

Markov Brains using only deterministic logic gates experience a slight loss in performance due to noise, and do not form more representations and also do not further smear them. While this means that they do not experience a benefit from sensor noise, they also naturally do not smear representations. However, they also tend to have more representations at the end of evolutionary adaptation anyways. This suggests that one should use sensor noise in order to increase the amount of representations cognitive systems such as RNNs and LSTMs evolve. But at the same time, we should find another way to prevent systems from smearing said representations.

Acknowledgements

This work was supported in part by Michigan State University through computational resources provided by the Institute for Cyber-Enabled Research, and is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454.

References

Brown, W. M., Gedeon, T. D., and Groves, D. I. (2003). Use of noise to augment training data: a neural network method of mineral-potential mapping in regions of limited known deposit examples. *Natural Resources Research*, 12(2):141–152.

Hintze, A., Edlund, J. A., Olson, R. S., Knoester, D. B., Schossau, J., Albantakis, L., Tehrani-Saleh, A., Kvam, P., Sheneman, L., Goldsby, H., et al. (2017). Markov brains: A technical introduction. *arXiv preprint arXiv:1709.05601*.

Hintze, A., Kirkpatrick, D., and Adami, C. (2018). The structure of evolved representations across different substrates for artificial intelligence. In *Artificial Life Conference Proceedings*, pages 388–395. MIT Press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Lauzon, F. Q. (2012). An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 1438–1439. IEEE.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.

Marstaller, L., Hintze, A., and Adami, C. (2013). The evolution of representation in simple cognitive networks. *Neural computation*, 25:2079–2107.

Merritt, D. J. and Brannon, E. M. (2013). Nothing to it: Precursors to a zero concept in preschoolers. *Behavioural processes*, 93:91–97.

Merritt, D. J., Rugani, R., and Brannon, E. M. (2009). Empty sets as part of the numerical continuum: conceptual precursors to the zero concept in rhesus monkeys. *Journal of Experimental Psychology: General*, 138(2):258.

Moran, P. A. P. (1958). Random processes in genetics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 54, pages 60–71. Cambridge University Press.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Nieder, A. (2018). Honey bees zero in on the empty set. *Science*, 360(6393):1069–1070.

Quiroga, R. Q., Fried, I., and Koch, C. (2013). Brain cells for grandmother. *Scientific American*, 308(2):30–35.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., and Edwards, D. D. (2003). *Artificial intelligence: A modern approach*. Prentice Hall, Upper Saddle River.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schossau, J., Adami, C., and Hintze, A. (2015). Information-theoretic neuro-correlates boost evolution of cognitive systems. *Entropy*, 18(1):6.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210.

Probabilistic Program Neurogenesis

Charles E. Martin¹ and Praveen K. Pilly¹

¹HRL Laboratories, Malibu, CA 90265
cemartin@hrl.com

Abstract

We present a new method for addressing the challenge of continual learning wherein an agent must adapt to new tasks while maintaining high performance on previously learned tasks. To accomplish this, an agent must identify previously acquired information that generalizes to the new task while also adapting its internal model to learn information that is specific to the new task. Our approach is based on neurogenesis, which involves adding new neurons to a previously trained neural network in an intelligent way. To our knowledge, we are the first to leverage probabilistic programming within the framework of evolutionary computation to optimize the growth of neural networks for continual learning. Through a series of experiments, we show that our approach is able to consistently find better performing solutions than genetic algorithms and it is able to do so faster.

Introduction

In this work, we consider the challenge of continual learning. Agents that act and learn in the real world must develop the skills to handle a wide variety of different tasks. These tasks are inevitably associated with different distributions of observations and responses. The diversity of tasks encountered in complex environments may make the task of continual learning seem improbable at best, and impossible at worst. However, tasks are often related to one another. For example, some of the visual attributes that an autonomous vehicle learns for detecting cars on the road may also be used when learning to identify motorcycles. Thus, an agent may be able to avoid learning a new task from scratch. The key is for the agent to efficiently determine what information from previous tasks is reusable for a new task and to update its internal decision making models with new, task-relevant information while avoiding catastrophic forgetting. Succinctly, an agent must be able to learn new tasks efficiently while not forgetting how to perform previously learned tasks.

There are a variety of methodologies for tackling the continual learning problem [Parisi *et al.*, 2019]. Existing approaches can be roughly divided into three categories: those that focus on consolidating synapses in a neural network that are critical to retain previous knowledge [Kirkpatrick *et al.*,

2017; Zenke *et al.*, 2017; Aljundi *et al.*, 2018; Kolouri *et al.*, 2019]; those that employ either an explicit memory buffer or a generative model to be able to interleave the learning of new knowledge with that based on stored or generated data for the old tasks [McClelland *et al.*, 1995; Ans *et al.*, 2004; Atkinson *et al.*, 2018; Ketz *et al.*, 2019]; and those that dynamically change the structure of the neural network to accommodate new knowledge [Rusu *et al.*, 2016; Draelos *et al.*, 2017; Yoon *et al.*, 2017]. Here, we address the challenge of continual learning by intelligently adding new neurons to an existing neural network (neurogenesis) so that it learns to solve a new task without forgetting how to solve previously learned tasks. The challenge is in determining how many neurons to add and where to add them so that a desired accuracy on the new task is achieved with as few new neurons as possible. In our approach the network is able to learn to leverage knowledge that was acquired when learning previous tasks, but which is relevant to the new task. This information sharing can substantially reduce the number of new neurons that need to be added to achieve a desired level of performance.

Our approach takes a unique angle towards neurogenesis by treating it as an optimization problem and using a novel combination of probabilistic programming and evolutionary computation. Specifically, we leverage the Estimation of Distribution Algorithm (EDA) framework. However, instead of representing the distribution over solutions using a Bayesian network, which is most common, we use a probabilistic program designed specifically for neurogenesis.

This paper is organized as follows. We first discuss relevant background and then move on to our methodology, which covers neurogenesis and our estimation of distribution algorithm. We then describe our probabilistic program and learning process. Next, we describe our experimental setup and results, and lastly provide a discussion and concluding remarks.

Background

The approaches described here involve modifying the architecture of a neural network to enable continual learning

without catastrophic forgetting. In Rusu *et al.* [2016], when a new task arrives, a new sub-network is added to the parent network to facilitate learning. The sub-network receives connections from the parent network so that it can integrate previously learned useful information, but the weights in the parent network remain fixed. This approach prevents catastrophic forgetting, but does not scale well with increasing numbers of tasks. In contrast, our approach learns where and how many neurons to add.

Draelos *et al.* [2017] proposed the neurogenesis deep learning (NDL) model to incrementally train an autoencoder on new MNIST digits by adding new neurons at each layer and employing a pseudo-rehearsal process called “intrinsic replay” for preserve the performance on the old MNIST digits. Yoon *et al.* [2017] proposed a dynamically expanding network (DEN) that selectively adapts network weights and also expand network structure at each layer as needed using group sparse regularization in an online manner to facilitate the sequential learning of tasks. These approaches are computationally efficient because they only leverage *local* statistical properties of neurons and/or layers. However, this is also a disadvantage because they will often fail to identify the best solutions. In contrast, our approach encodes an efficient *global* search process and thus has a higher probability of finding optimal or near-optimal solutions.

Another popular approach is to use principles of self-organization to incrementally “grow” a neural network. Recent examples include Part and Lemon [2017] and Mici *et al.* [2018]. These approaches share a similar advantage in terms of computational efficiency, but also the disadvantage of not performing a global search like our approach.

Methodology

In this section, we describe the components of our approach. We start with a description of our neurogenesis-based approach to the continual learning problem. Next, we describe the estimation of distribution algorithm [Larraaga and Lozano, 2001], which is our overarching optimization method. Then, we discuss the probabilistic program learner (PPL) that generates new individuals and is the heart of our approach. Lastly, we describe the learning procedure used to adapt the parameters of the PPL so that it focuses on promising regions of the search space.

Neurogenesis for Continual Learning

In this work, we focused on adding new neurons to fully connected, feed-forward neural networks, however, our approach is equally applicable to recurrent and convolutional networks. Our method is initialized with a neural network that has been previously trained on one or more tasks. The neurogenesis process begins when a new task is presented. Here, we do not address the need to identify the arrival of a new task, but in practice, assuming that the input distribution for the new task is different, one can identify changes

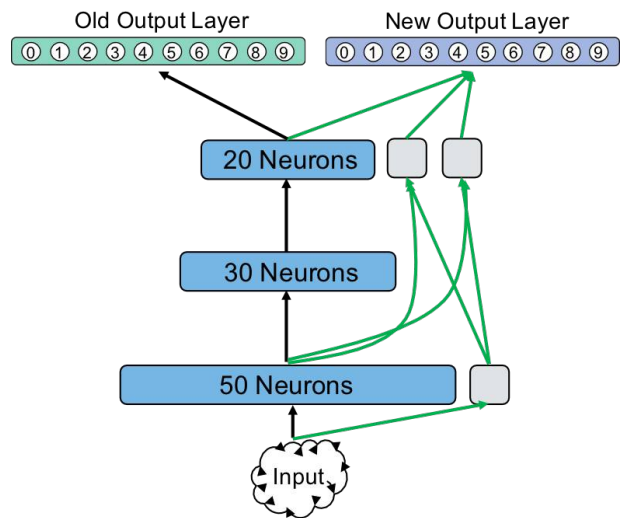


Figure 1: Example of neurogenesis used in our approach, illustrated here for continual learning of optical character recognition tasks. The layers of the parent network are shown in blue, while the newly added neurons are shown in gray. The black connections have fixed weights, while the green connections have trainable weights. A new head (shown in purple) is added to act as the output layer for the new task. This prevents the green connections, whose weights are trained on the new task, from interfering with the parent network’s performance on the old task.

in the statistics of the neuron activations in the penultimate layer of the network to infer that a new task has arrived. An example of neurogenesis is shown in Figure 1. The gray blocks are new neurons. The weights on the black connections were learned from a previous task and are fixed, while the green connections are new and have trainable weights. First, a new output (classification/prediction) layer is added to the network. Next, new neurons are added and connections are established. When new neurons are added to a layer it effectively creates a new layer that emits new connections (green) to the new neurons in the nearest layer above, which may be the output layer. Additionally, if new neurons are added to the first hidden layer, then new connections are established from the input layer to these neurons. Restrictions on the connectivity patterns of new connections ensure that the performance of the network on old tasks is not altered. This is because the activations of new neurons do not affect the activations of old neurons.

Estimation of Distribution Algorithm

Estimation of Distribution Algorithms (EDAs) are a class of evolutionary computation algorithms [Larraaga and Lozano, 2001]. Unlike most EC algorithms, EDAs explicitly encode a probability distribution over the search space from which

new individuals are generated. Rather than using mutation and crossover operators, an EDA adapts the parameters of a probabilistic model during the optimization process so that new individuals are increasingly likely to be selected from good regions of the search space. In our approach the probabilistic model is a probabilistic program, which is described in the next subsection.

Figure 2 shows a diagram of our EDA. The process begins by generating a new population of individuals using the program. Each individual is a neural network and is trained on a pre-defined task, such as a classification problem. Next, the fitness of each individual is computed. We use a fitness function that captures the trade-off between classification accuracy and model complexity. The fitness function is given by

$$Fitness = k \cdot accuracy - complexity, \quad (1)$$

where $accuracy \in [0, 1]$ and is the accuracy on the new task, $complexity$ is the model complexity given by the number of new neurons divided by the number of neurons in the parent network, and $k \in \mathbb{R}^+$. The selection process then proceeds in two separate steps. First, one individual is selected using tournament selection. This provides some degree of exploration since the individual with the highest fitness may not be selected. Second, we use a hall-of-fame that always and only contains the best performing individual found over the course of the entire optimization process. If the fitness of the best performing individual in the current population exceeds the best fitness found thus far, then it becomes the individual in the hall-of-fame. The last step is to update the parameters of the probabilistic program. This is done using the newly selected individual and the individual in the hall-of-fame, which may be the same. Updating the parameters tends to adjust the probability distribution represented by the probabilistic program in such a way that it is more likely to sample from good regions of the search space. At this stage the EDA either generates a new population using the updated program or terminates. In this work the termination criterion is that the elapsed wall-clock time has exceeded 21 min, though other criteria are possible, such as those based on a maximum number of iterations or fitness level.

Probabilistic Program Learner

Our approach uses a probabilistic program as a generative model to create new individuals. It has trainable parameters that are adjusted through a learning process, so that over the course of the optimization, individuals are sampled with increasing frequency from the best regions of the search space. We refer to this component as the *Probabilistic Program Learner* (PPL). Figure 3 shows a diagrammatic representation of the PPL. The purple boxes are random variables that specify the characteristics of the individuals, such as the number of new neurons to add. The light blue boxes are latent (unobserved) random variables that control the

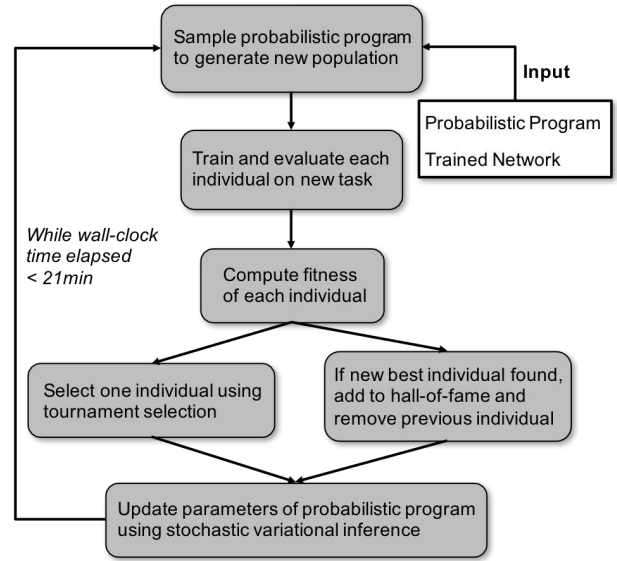


Figure 2: Our EDA used to execute neurogenesis. The uniqueness of our approach lies in the probabilistic program that learns to generate good performing individuals.

shapes of the probability distributions governing the aforementioned characteristics. These boxes each contain a descriptive name, the data type of the random variable, and the form of the probability distribution from which it is drawn. The dark blue boxes represent learnable parameters, hence the word “learnable” in the acronym PPL. They show the names of the parameters, their data types, and initial values. The program input (gray box) specifies a list of the maximum numbers of new neurons that may be added to any layer and the number of layers in the parent network, both of which are fixed throughout the optimization process.

We now describe the stochastic process of sampling from the PPL to generate a new individual. The sub-process shown at the top of Figure 3 begins by sampling a parameter $p_s \sim \text{Dirichlet}(c)$, where c is a learnable parameter. Next, an index is drawn from $Ind_{max} \sim \text{Multinomial}(1, p_s)$. This index is used later in the program to access the array `MaxSizes`. The next sub-process begins by sampling a list of parameters $p_n \sim \text{Beta}(alphas, betas)$, where $alphas$ and $betas$ are learnable parameters. The parameters p_n are the individual probabilities of selecting each of the layers in the parent network to receive new neurons. Next, the layers that will receive new neurons are selected according to $L \sim \text{Bernoulli}(p_n)$, where L is a Boolean array. The last sub-process selects how many new neurons to add to each layer $m \in N$. The control logic allows new nodes to be added only to those layers selected in the second sub-process (i.e., $L[m] == \text{True}$). The probability of adding a single new neuron to layer m is drawn from $p_a \sim \text{Beta}(alpha_m, beta_m)$, where $alpha_m$ and $beta_m$ are

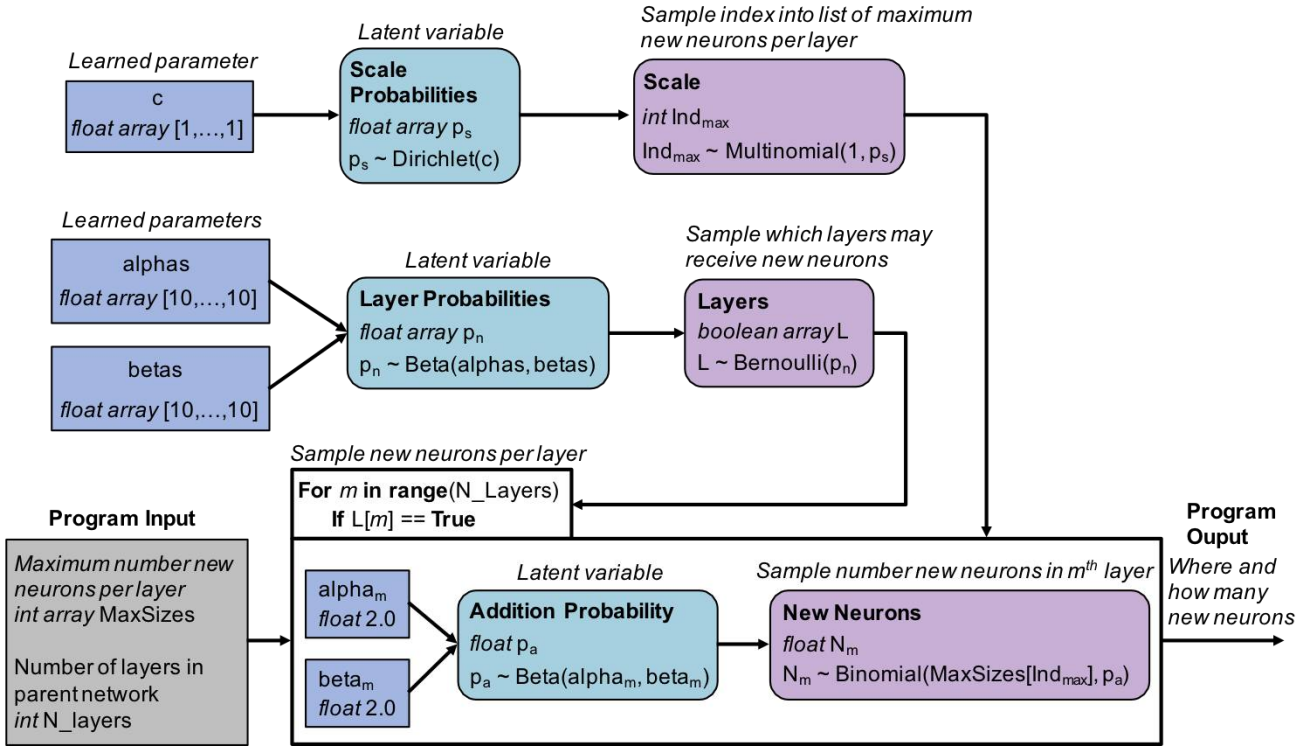


Figure 3: Illustration of the probabilistic program learner used in our approach.

learnable parameters. The total number of new neurons to add to the m^{th} layer is sampled from a binomial distribution according to $N_m \sim \text{Binomial}(\text{MaxSizes}[\text{Ind}_{\max}], p_a)$, where $\text{MaxSizes}[\text{Ind}_{\max}]$ is the maximum number of new nodes that can be added to any layer. Once the loop is completed, the program output specifies where and how many new neurons to add, which is sufficient to build a new individual.

Learning

In each generation of the optimization process, after selection has occurred, the learnable parameters of the probabilistic program are updated based on the attributes of the selected individuals. We refer to these attributes as observable variables and denote them by x . For each individual, the observables are the maximum number of new neurons per layer (Ind_{\max}), the layers that are permitted to receive new neurons (L), and the number of new neurons in the m^{th} layer (N_m). The random variables that are not observable are referred to as latent variables and are denoted by z . The latent variables in our model are p_s , p_n , and p_a , which are described in the previous section. For each individual, the learning algorithm updates the parameters (c , α , β , α_m , and β_m) so that the probabilistic program is more likely to generate similar individuals in subsequent generations. We used stochastic variational inference as the

learning algorithm [Hoffman *et al.*, 2013].

Experimental Setup

We applied our approach to a variant of the MNIST optical character recognition problem [LeCun *et al.*, 1998]. Our setup is shown in Figure 4. First, a “parent” neural network was trained to classify gray-scale images of handwritten digits from 0 to 9 (old task). The training dataset consisted of 1000 randomly selected images with approximately 100 images per class. The parent networks used in our experiments achieved an average testing accuracy of 87.4%. The images were flattened into a 748-dimensional vector. The parent network consisted of 3 hidden layers with hyperbolic tangent (tanh) activation functions. There were 50, 30, and 20 neurons in the first, second, and third hidden layers, respectively.

A new task was created by randomly generating a permutation mask and applying it to each of the images in the dataset. The training dataset for the new task consisted of 1000 randomly selected, permuted images with approximately 100 images per class. The permutation mask was created by randomly selecting two non-intersecting sets of pixel indices, and then swapping the corresponding pixels in each image. In our experiments, 50% of the pixels in each image were modified. The resulting new task was similar enough to the old task that some information from the parent network

was still valid, but different enough that adding new neurons significantly improved performance on the new task. If no new neurons were added, the parent network would experience catastrophic forgetting (loss of performance on the old task) due to the change in the input distribution. We set the maximum number of new neurons that could be added to any layer at 50. This leads to a total of $50^3 = 125,000$ unique individuals.

We set the parameter k in Equation 1 to 10.0. We found that smaller values of k resulted in the complexity term dominating the fitness, which resulted in a fairly simple fitness landscape with the global optimum being achieved by adding only 1 to 3 neurons at *any* layer. In contrast, setting $k = 10.0$ provided better balance between accuracy and complexity, and consequently, a more challenging optimization problem with many good, but suboptimal, local minima. For this setting, the global optimum is achieved by adding 16 new neurons to the first hidden layer and no new neurons to the second and third hidden layers. However, good, but sub-optimal, local minima can be achieved by adding new neurons to only the second or third hidden layers.

We used a genetic algorithm (GA) as basis for comparison with our approach. Genetic algorithms [Whitley, 1994] are a good fit to this problem due to the discrete nature of the search space. For the GA, an individual was encoded as a vector of length 3, where the values of the components indicated the number of new neurons to add in each of the three hidden layers. The maximum number of new neurons that could be added to any layer was 50. We used a population size of 30 and tournament selection with a tournament size of 3. Among the selected population, an individual was chosen for crossover with another randomly chosen individual with probability 0.5 and was chosen for mutation with probability 0.2. Once selected for mutation, each entry in the individual was mutated uniformly at random with probability 0.3 to a value in the interval $[0, 50]$. The relatively high mutation rate was found to prevent pre-mature convergence to poor solutions. We used two-point crossover with the crossover points being selected uniformly at random.

Results

We compared the performance of our approach to those of the GA using the experimental setup described in the previous section. We ran a total of 150 trials for the two approaches combined. Each trial started with a newly initialized population, and parameters in the case of the PPL, and then the optimization process was run for 21 minutes of wall-clock time. Wall-clock time was used because both the GA and our approach are randomized search algorithms and on average one generation of our approach takes longer than the GA due to the probabilistic inference. The average accuracy attained on the new task by the solutions from our approach was 87.7% and for the GA it was 87.6%. The

relationship between accuracy and fitness is shown in Eq. 1. The fact that the accuracy is close to that of the parent network on the old task (87.4%) illustrates the effectiveness of neurogenesis for continual learning. The typical solution found by PPN was 12-17 new neurons added only to the first layer. The typical solution found by the GA had greater variability and was either 11-20 new neurons added only to the first layer or 13-20 new neurons added only to the second layer. Based on a large number of experimental trials, we observe that the global optimal solution is approximately 16 new neurons added only to the first layer.

The results of the first analysis are shown in Figures 5 and 6. Each figure shows the average best fitness achieved by the PPN (blue) and GA (red) as a function of elapsed run time in minutes. Figure 5 is from 2 to 5 minutes and the Figure 6 is from 5 to 21 minutes. Based on the curves in Figure 5, it can be seen that on average PPN reaches near optimal solutions (a fitness of about 60.0) within the first 2 minutes of simulation time, whereas it takes the GA about 5 minutes to reach a comparable level of fitness. Figure 6 shows that in the long run our approach continues to improve and outperform the GA.

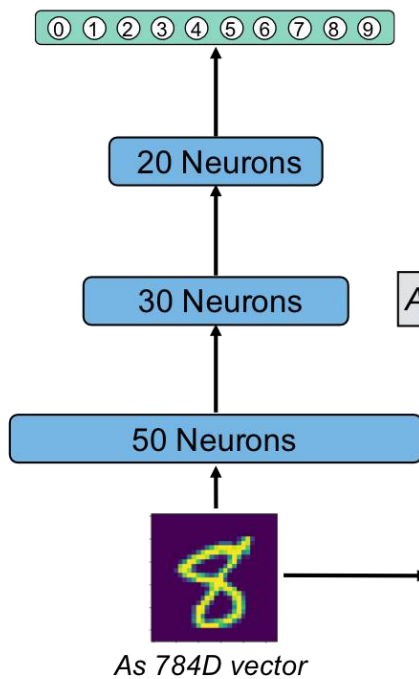
The next analysis examined the consistency with which the PPN and GA were able to find solutions that achieved particular fitness levels. Figure 7 shows the fraction of trial runs on which the best fitness found exceeded various lower bounds. The results for the PPN are in blue and those of the GA are in red. We can see that for each fitness lower bound on the x-axis, our approach exceeds the success frequency of the GA, and for the higher (more difficult to achieve) fitness levels (> 61) the success rate of our approach is at least double that of the GA. These results demonstrate that our approach consistently finds better solutions than the GA.

Discussion

We have demonstrated the effectiveness of our approach by showing that it is able to consistently find better performing solutions than the GA and it is able to do so faster. We believe that one of the primary reasons for this is that the probabilistic program represents a distribution over distributions, which promotes exploration and thus the discovery of better solutions more quickly. As can be seen in Figure 3, the parameters (p_s, p_n, p_a) of the distributions governing the observable random variables (Ind_{max}, L, N_m) , which control the structural attributes of an individual, are themselves (latent) random variables. When running the program in the forward direction to generate a new individual, we must sample the latent random variables first and then use them to determine the distribution over the observables.

Another reason for the effectiveness of our approach is that it is easy to incorporate prior knowledge about a domain directly into the search process. Our probabilistic program, rather than using crossover and mutation operators to implicitly define the search distribution, enables the designer/user

Old Task: Classify MNIST Digits



New Task: Classify Corrupted Digits

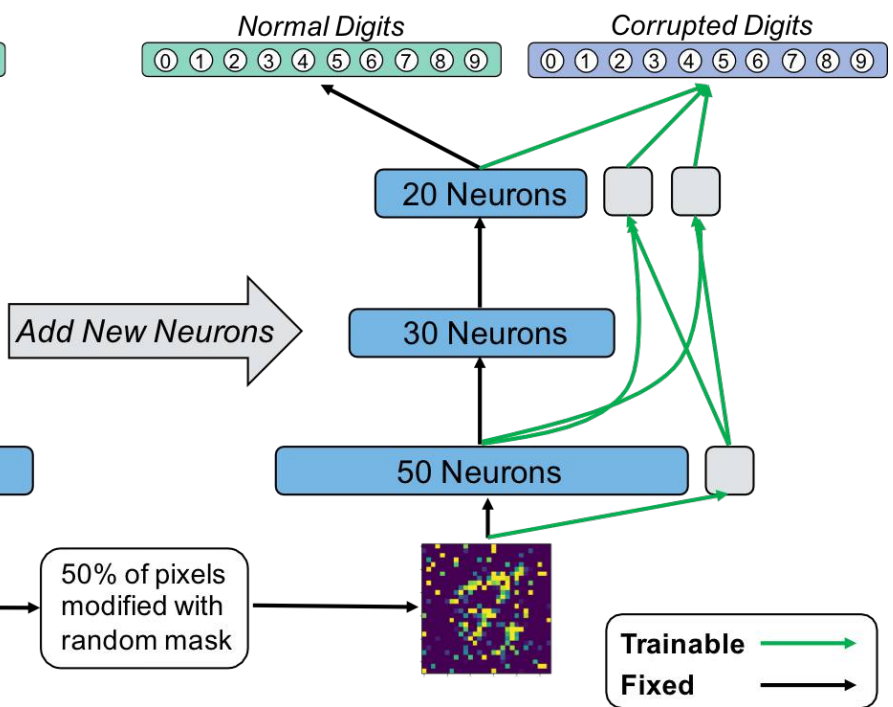


Figure 4: Illustration of our experimental setup. On the left, a *parent* network is trained to classify images of handwritten digits (0-9). A new task is created by randomly generating a permutation mask that randomly and uniformly selects 50% of the pixels to be swapped. The mask is then applied to each image. This way the new task is different, but still related to the old task. Neurogenesis is shown on the right. Our approach identifies where new neurons should be placed to minimize network complexity while achieving good classification performance on the new task (corrupted digits). The new (green) connections have trainable weights, while the weights on the old (black) connections remain fixed to prevent catastrophic forgetting.

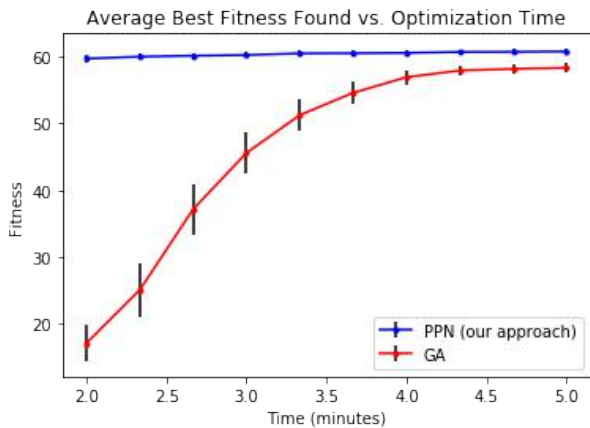


Figure 5: Comparison of the average best fitness found over the first 5 minutes of optimization time. The fitness values used for plotting have been shifted by -8.0 and scaled by 100.0. This was done solely for the purpose of improving visual interpretability of the results. Our approach (blue) quickly finds near optimal solutions, while the GA (red) is slower to find good solutions. Our approach starts much higher due to the informative prior in the probabilistic program. As a point of reference, when a parent network makes predictions on the new task prior to neurogenesis ($complexity = 0$ in Eq. 1) the expected fitness is -400 ($accuracy = 0.4$). The error bars are 90% confidence intervals.

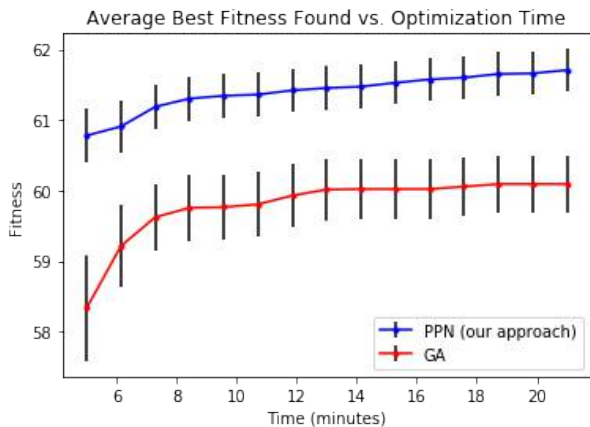


Figure 6: Comparison of the average best fitness found over the next 5 to 21 minutes of optimization time. The fitness values used for plotting have been shifted by -8.0 and scaled by 100.0 to improve readability. Our approach (blue) continues to improve, while the GA (red) stagnates. The error bars are 90% confidence intervals.

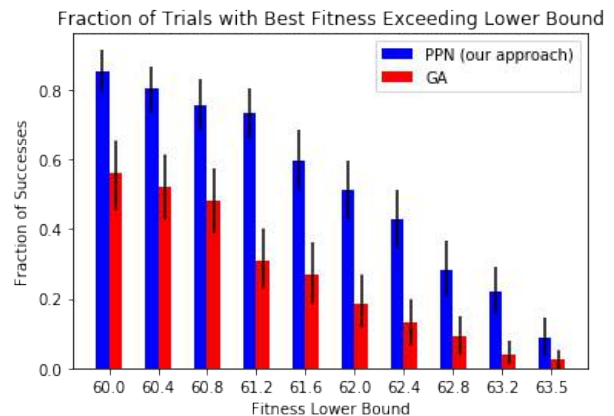


Figure 7: Comparison of how often our approach (blue) finds solutions with fitness values exceeding various lower bounds. The fitness values used for plotting have been shifted by -8.0 and scaled by 100.0 to improve readability. For each fitness lower bound, our approach exceeds the GA. The error bars are 90% confidence intervals.

to build in semantics that are believed to be important. We were able to take advantage of this with the portion of the program that selects the layers that are permitted to receive new neurons. Our original formulation of the PPL did not have this logic, but we quickly realized based on empirical observations that it is important to efficiently explore regions of the search space where one or more layers do not receive any new neurons. Subsequently, a simple modification to our program enabled this enhancement.

Conclusions

We have demonstrated a new approach to neurogenesis based on probabilistic program learning. Our method helps to address the continual learning problem wherein learning occurs sequentially on different tasks. In this setting, when learning a new task, it is of utmost importance to avoid catastrophic forgetting of previously learned tasks. However, previously learned information that generalizes to the new task should be leveraged to limit growth in the complexity of the learning model.

Past work on neurogenesis has largely focused on employing ad hoc rules or leveraging local statistical properties of neurons and/or layers [Draeos *et al.*, 2017; Yoon *et al.*, 2017] to decide the number and location of the new neurons. In contrast, our approach employs an efficient global search process using a probabilistic program, which has a higher probability of finding optimal or near-optimal solutions. Moreover, the probabilistic program process allows for easy integration of new features and rules to better regulate the addition of new neurons. In the future we will also validate our framework for continual learning us-

ing reinforcement learning tasks (e.g., those from the OpenAI Gym) and more challenging datasets such as ImageNet [Krizhevsky *et al.*, 2012], and make comparisons with additional methods such as evolution strategies. Another avenue we plan to pursue is extending our approach to convolutional neural networks (CNNs). In this application, instead of adding new neurons, PPL would add new feature maps to the convolutional layers.

Acknowledgments

We thank Risto Miikkulainen, Sebastian Risi, Andrea Soltoggio, and Jean-Baptiste Mouret for conceptual discussions surrounding the work. This material is based upon work supported by the United States Air Force and DARPA under contract no. FA8750-18-C-0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

- Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.
- Bernard Ans, Stephane Rousset, Robert M French, and Serban Musca. Self-refreshing memory in artificial neural networks: Learning temporal sequences without catastrophic forgetting. *Connection Science*, 16(2):71–99, 2004.
- Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. Achieving deep reinforcement learning without catastrophic forgetting. *arXiv*, 1812.02464, 2018.
- T.J. Draelos, N.E. Miner, C.C. Lamb, J.A. Cox, C.M. Vineyard, K.D. Carlson, W.M. Severa, C.D. James, and J.B. Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. *International Joint Conference on Neural Networks (IJCNN)*, pages 526–533, 2017.
- M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- Nicholas Ketz, Soheil Kolouri, and Praveen K. Pilly. Using world models for pseudo-rehearsal in continual learning. *arXiv*, 1903.02647, 2019.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.
- Soheil Kolouri, Nicholas Ketz, Xinyun Zou, Jeffrey Krichmar, and Praveen K. Pilly. Attention-based selective plasticity. *arXiv*, 1903.06070, 2019.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- P. Larraaga and J.A. Lozano. Estimation of distribution algorithms: A new tool for evolutionary computation. vol. 2. *Springer Science Business Media*, 2001.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- James L McClelland, Bruce L McNaughton, and Randall C O’Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419, 1995.
- Luiza Mici, German Parisi, and Stefan Wermter. Self-organizing neural network architecture for learning human-object interactions. *Neurocomputing*, 307:14–24, 2018.
- G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.
- Jose Part and Oliver Lemon. Incremental online learning of objects for robots operating in real environments. In *Proceedings of the Joint IEEE Int. Conf. on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 304–310, 2017.
- A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv*, 1606.04671, 2016.
- D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.
- J. Yoon, E. Yang, J. Lee, and S.J. Hwang. Lifelong learning with dynamically expandable networks. *arXiv*, 1708.01547, 2017.
- Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995. JMLR.org, 2017.

Self-optimization in a Hopfield neural network based on the *C. elegans* connectome

Alejandro Morales^{1*} and Tom Froese^{1,2}

¹Institute for Applied Mathematics and Systems Research, National Autonomous University of Mexico, Mexico

²Center for the Sciences of Complexity, National Autonomous University of Mexico, Mexico

*Corresponding author: alejandroe@ciencias.unam.mx

Abstract

It has recently been demonstrated that a Hopfield neural network that learns its own attractor configurations, for instance by repeatedly resetting the network to an arbitrary state and applying Hebbian learning after convergence, is able to form an associative memory of its attractors and thereby facilitate future convergences on better attractors. This process of structural self-optimization has so far only been demonstrated on relatively small artificial neural networks with random or highly regular and constrained topologies, and it remains an open question to what extent it can be generalized to more biologically realistic topologies. In this work, we therefore test this process by running it on the connectome of the widely studied nematode worm, *C. elegans*, the only living being whose neural system has been mapped in its entirety. Our results demonstrate, for the first time, that the self-optimization process can be generalized to bigger and biologically plausible networks. We conclude by speculating that the reset-convergence mechanism could find a biological equivalent in the sleep-wake cycle in *C. elegans*.

Introduction

Caenorhabditis elegans is a one-millimeter-long soil worm of the nematode family. It consists of only 959 cells, of which 302 belong to the nervous system. *C. elegans* is a case study model for biology (Girard et al., 2006), thanks to its small cell count, short lifespan and rapid reproduction. The full understanding of cellular development and the function of neural system mechanisms in human biology remains an open question, which makes *C. elegans* a good alternative. *C. elegans* is the first multicellular organism whose genome has been sequenced in its entirety (Waterston and Sulston, 1998), as well as the first animal from which the complete mapping of synaptic connections, called the connectome, has been completed (White et al., 1986). Knowledge of the connectivity of a complete neural system can help to better understand genetic and molecular mechanisms in neuroscience.

Network science provides further information concerning the structure of the *C. elegans* connectome, such as the statistical and topological properties of the network (Watts and Strogatz, 1998; Varshney et al., 2011). In terms of worm's

network dynamics, simulation models have begun to play a bigger role in the experiment-theory cycle in an attempt to better understand the neural underpinnings of worm behavior (Izquierdo, 2018). *C. elegans* has been studied also in the artificial life research community (Kitano et al., 1998; Winkler et al., 2009; Hattori et al., 2012; Izquierdo and Beer, 2015; Beer and Izquierdo, 2016; Aguilera et al., 2017). However, so far these simulation models have only worked with smaller circuits, and not with the whole connectome, and so the properties of the connectome's state space remain underexplored.

In this paper we therefore go the other way: we abstract away from the worm's situated behavior, and focus on the dynamics of its whole connectome. In particular, we are interested in determining if the connectome is able to self-optimize its connectivity, namely by forming an associative memory of its attractors such that the convergence of neural states is biased toward better attractors.

The self-optimization process

The self-optimization process used in this paper is based on the work of Watson et al. (2011), using the original network model proposed by Hopfield (1982). We have two discrete values for the states in the network $s_i = \pm 1$. Such values represent the activity of a neuron. Model dynamics include asynchronous neuron state updates, calculating with the following equation:

$$s_i(t+1) = \theta \left[\sum_j^N w_{ij} s_j(t) \right] \quad (1)$$

where w_{ij} is the connection weight between neuron i and j , and θ is the Heaviside threshold function (taking values -1 and $+1$ as negative and positive arguments, respectively).

In order to be able to test the connectome's convergent states with respect to its original connectivity, we differentiate two parts of w_{ij} , w_{ij}^O represents the original configuration of the edges at the beginning of the process, and w_{ij}^L store the Hebbian learning changes. Both make up the cur-

rent weights of the network:

$$w_{ij} = w_{ij}^O + w_{ij}^L \quad (2)$$

In the Hopfield network model, a node i satisfy a constraint posed by its interaction with node j if $s_i s_j w_{ij} > 0$. While actual interaction with neighbors determines the state update s_i , system energy represents the degree to which internal constraints with the original weight configuration, w^O , remain unsatisfied. It is calculated as follows:

$$E = - \sum_{ij} w_{ij}^O(t) s_i(t) s_j(t) \quad (3)$$

Hebbian learning should be applied to all system connections (i.e. the change in weight, $\Delta w_{ij} = \delta s_i s_j$, $\delta > 0$) to increase the attractor variation and reinforce learning.

$$w_{ij}^L(t+1) = w_{ij}^L(t) + \delta s_i(t) s_j(t) \quad (4)$$

The algorithm repeatedly goes through the following sequence of steps: (a) arbitrary assignment of states for the neurons (reset), (b) convergence of the network for a certain time period, most frequently resulting in an attractor, and (c) application of Hebbian learning.

Methods

The connectome. We ran the self-optimization algorithm on the most recent version of the *C. elegans* connectome produced by Jarrell et al. (2012). The database contains hermaphrodite neural system information (males arise infrequently, at 0.1%), such as synaptic direction, type of synaptic connection (synapse or gap junction), and the number of synapses between A and B neurons. We translated the connectome into a directed multigraph, with neurons as nodes and synapses as edges. This representation allows multiple synapses and gap junctions between the same two neurons, which naturally occur in the worm neural system.

Only 282 neurons belonging to a large somatic nervous system are taken into account. We did not consider pharyngeal neurons because they belong to another independent neural system (Albertson and Thomson, 1976; White et al., 1986). Neurons were arbitrarily assigned binary activation states $(-1, 1)$. In the neural network under study, chemical synapses are modeled as single-directed links between neurons (for example, $A \rightarrow B$ indicates that neuron A is presynaptic to neuron B , and B is postsynaptic to A). Gap junctions are represented in the connectome as double-linked neurons (if two neurons, C and D , have a gap junction between them, there are two links: $C \rightarrow D$ and $D \rightarrow C$). There are 5,611 connections in total. 62.5% of the total connections are chemical synapses, while 37.5% are gap junctions.

We assigned the number of synapses between neurons as the weight of each edge normalized in the interval $(0, 1)$.

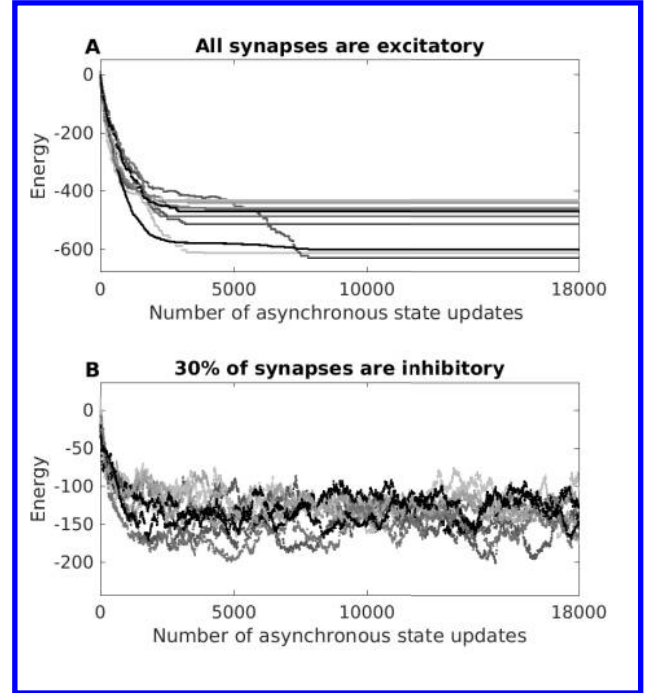


Figure 1: Illustrative examples of network state convergence without self-optimization. We show network energy after successive state updates for 10 independent convergences. **A** Scenario without inhibitory connections (negative weights). **B** Scenario with 30% inhibitory connections (negative weights). The addition of inhibitory connections increases the difficulty of coordinating neural activity across the connectome. Note the different y-scales when comparing panels A and B.

Jarrell et al. (2012) estimate the functional strength of synaptic interactions with the resulting number. Both links in gap junctions are assigned the same weight, and values vary between 1 and 81 (5.07 is the average weight per link). We clip to 1 the 15 high-weight values, which we determined with an arbitrary cut-off of weights greater than 44, before normalization. Edges with higher weight values are scarce (only 34 connections has a higher value than 44). Reduction of this outliers broadens the space-state exploration during the self-optimization; in other words, the range value of local attractors is wider.

Since the connectome does not contain information concerning inhibitory or excitatory connections, we tested two separate scenarios. First, a scenario in which all synapses are excitatory, and second, a scenario in which a percentage of synapses is inhibitory. For the second scenario we arbitrarily selected 30% of the edges and assigned negative weights to them, based on the fact that 30% is a percentage of inhibitory synapses that has been proven optimal in the

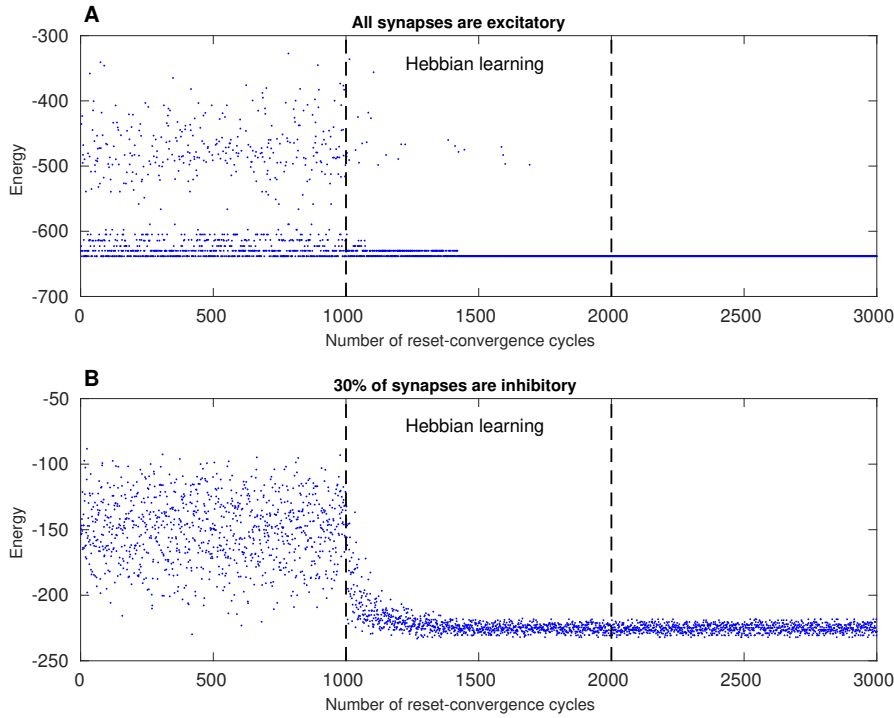


Figure 2: Illustrative examples of self-optimization in two different scenarios. **A** without inhibitory connections (negative weights) at the beginning of the process, and **B** with 30% inhibitory neurons (negative weights). Each panel shows the energy of the neuron states after 1,000 reset-convergence cycles during three distinct phases: before learning (1-1,000), during the self-optimization process (1,001-2,000), and after learning (2,001-3,000). Examples of state convergences of the phase before learning can be seen in Figure 1. Self-optimization can be observed in both scenarios. However, the processes can be distinguished in that better attractors tend to be found without inhibitory synapses, but inhibitory synapses preserve a greater diversity of attractors.

simulation of multi-task learning processes, at least for mammals (Capano et al., 2015).

The neural network. Once the connectome was mapped to a neural network, extra zero-weighted edges were added to make a complete directed graph so that Hebbian learning could add extra connections by changing their weights from zero to non-zero values. In other words, we added previously non-existing connections between neurons in two directions, and those extra edges represent potential connections, which may be needed for the removal of constraints regarding interactions during the process (constraints posed in terms of the satisfaction equation, $s_i s_j w_{ij} > 0$). There is an initial total of 5,611 connections in the network (from the connectome), which after adding extra edges rises to 80,213. These extra edges do not have anatomical equivalents in the connectome, but they could be conceived of in terms of functional rather than structural connectivity. We measured and reported the energy in terms of the original weight configu-

ration (5611 connections from the connectome). When the learning phase ends, the constraint satisfaction percentage is also based in this 5611 connections.

In a multigraph schema we must consider an identifier (k) for each multiple edge that shares the same direction. In light of this, the weight sum of all edges with the same direction was added to the state update equation (Eq. 1), rewritten as follows:

$$s_i(t+1) = \theta \left[\sum_j^N \left(\sum_k w_{ijk} \right) s_j(t) \right] \quad (5)$$

A multigraph schema is also reflected in the way we calculate system energy and Hebbian learning (Eq. 3 and 4, respectively), because we iterate above all edges including the multiple ones that share the same direction:

$$E = - \sum_{ijk}^N w_{ijk}^O(t) s_i(t) s_j(t) \quad (6)$$

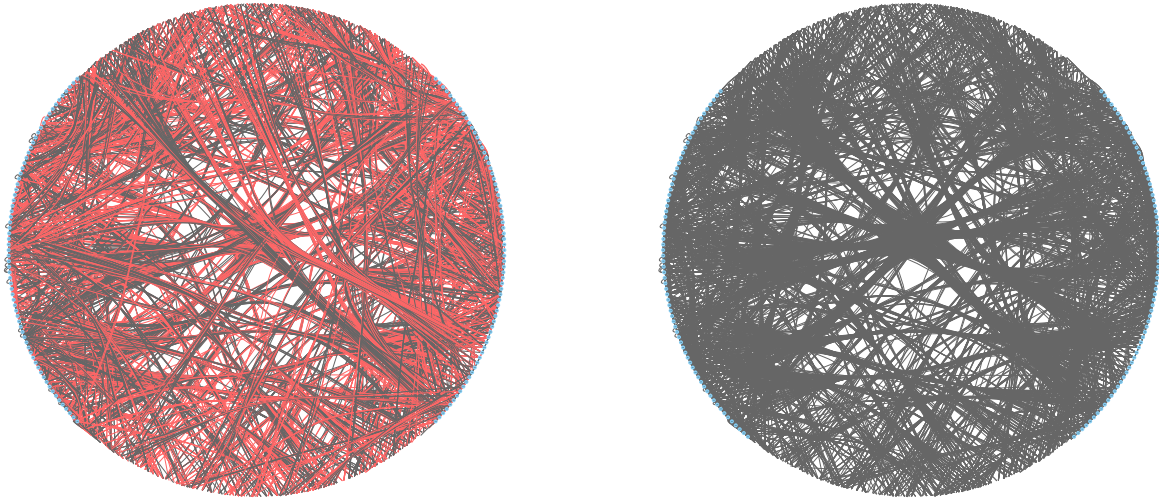


Figure 3: Visualization of the *C. elegans* connectome, consisting in 282 neurons with 5,611 excitatory connections, in a circle layout with a topological bundling procedure. Black connections represent edges that satisfy the constraints that were specified by the networks original weight configuration. Red ones represent edges that do not satisfy these constraints. We present two illustrative examples. The figure on the left shows the connectome with arbitrary states at the beginning of the self-optimization procedure, and the figure on the right shows the connectome with converged states after learning. Complete constraint satisfaction can be observed on the right side.

$$w_{ijk}^L(t+1) = w_{ijk}^L(t) + \delta s_i(t) s_j(t) \quad (7)$$

In summary, the network resulting from the connectome data differs from a traditional Hopfield neural network in the following ways:

- Asymmetrical connections are permitted.
- Self-connections are permitted.
- Since there are synapses and gap junctions between the same neurons within the connectome, multiple directed connections are permitted.

The first two connectivity restrictions were already lifted by previous models of self-optimization (Zarco and Froese, 2018a; Froese and Manzanilla, 2018), but this is the first time that the process is tested with a multigraph.

The self-optimization algorithm repeatedly goes through the following sequence of steps: (a) arbitrary assignment of states for the neurons (reset), (b) convergence of the network for a certain time period, most frequently resulting in an attractor, and (c) application of Hebbian learning.

The number of steps for (b) has been adjusted to 18,000. We observe this to be an adequate quantity to ensure either stability or convergence in each cycle. We fixed the learning rate δ ($\delta = 0.00001$) for all experiments.

Results

Overall, the results demonstrate the feasibility of self-optimization on the connectome. We report the results of both the scenario that starts with all positive weights, and the scenario with 30% negative weights.

We first explored the attractor dynamics of the network without self-optimization. Figure 1 A,B shows the network energy after successive neuron state updates. The network always reaches an attractor before the next state reset in the scenario with all excitatory synapses (Panel A). Unsatisfied network constraints also tend to decrease in the scenario with inhibitory synapses, but the network does not always reach an attractor (Panel B). This shows that the addition of inhibitory synapses increases the difficulty of neural coordination.

We then explored the network's self-optimization capacity. The experiment shown in Figure 2 A,B consisted in three stages. First, we set an initial weight configuration taken from the connectome and normalized (only positive values in A and 30% negative values in B) and performed 1000 reset convergence cycles without Hebbian learning. Then, we applied the self-optimization process using 1000 reset-convergence cycles. Finally, we apply the network 1000 cycles without Hebbian learning using the configuration obtained by the self-optimization process. This graph illustrates a tendency to decrease global energy in both A and B, although a global attractor is not reached in B.

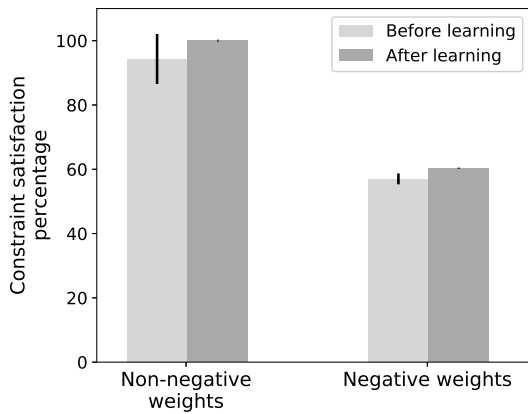


Figure 4: Percentage of constraint satisfaction before and after self-optimization (Hebbian learning), measured at the end of a reset-convergence cycle. For each case we consider the average of 1,000 cycles. A t-test was used for each data set (without negative weights and 30% negative weights) to see if the difference between before and after learning is significant. We obtain a p -value $\lll 0.05$ both for non-negative weights and 30% negative weights, indicating a highly significant improvement.

To illustrate the size and complexity of the connectome, we present in Figure 3 a diagram which shows the connectome structure with a circular neuron layout where the edges of different colors represent the satisfied and unsatisfied connections.

Figure 4 shows the percentage of constraint satisfaction before and after the self-optimization process. We applied a t-test to each data set to compare before and after learning.

Discussion

In the present work, we have tested the dynamics of the connectome of *C. elegans* with a Hopfield network and applied the self-optimization procedure in a multigraph structure. The results demonstrate that self-optimization can be generalized to this bigger and biologically more plausible topology. However, it is an open question to what extent this success has implications for understanding the operation of the nervous system of the worm. We respond to this open question by considering three issues.

First, is the connectivity required for the self-optimization process biologically plausible? We showed that under the right conditions the connectome network presents a tendency to optimize its own connectivity. It does so through Hebbian learning-based removal of constraints during repeated exploration of its state space. In the original Hopfield neural network model, this was achieved by restricting connections in such a way that the network state would nec-

essarily always converge on point attractors, namely by ensuring that connections are symmetric and that there are no self-recurrent connections. The restriction to point attractors and symmetric, non-recurrent connectivity is biologically implausible. An important step is therefore to show that self-optimization can still be achieved on an asymmetric, multiedged, recurrent network, which can give rise to a richer set of dynamics. We achieved this step with the present work.

Second, is the reset-convergence cycle required for the self-optimization process biologically plausible? In particular, what could be the mechanism that periodically relaxes the constraints of normal functioning, which permits the network to explore its state space? Following speculations by Woodward et al. (2015), in our model the neuron state resets could be interpreted biologically as a sleep-like state of *C. elegans*, especially when the worm is quiescent. *C. elegans* presents a quiescent behavior during lethargus, a sleep-like stage that occurs during larval development (Nelson and Raizen, 2013). Moreover, this sleep-like state also occurs during satiety and after exhaustion. Nevertheless, the way to define a wake-sleep cycle in the worm remains controversial. Our model suggests that it would be worthwhile to look for it, and to test its relationship to network self-optimization, perhaps akin to sleep-dependent learning.

Third, is the percentage and distribution of inhibitory connections biologically plausible? Again, we are limited by the lack of more detailed biological information. We introduced negative weights in the second scenario of the self-optimization process in an exploratory manner. The results suggest that their presence restricts the coordination between neurons. Such a restriction of neural coordination across the whole connectome could be biologically desirable, for instance by helping to avoid problems arising from excessive neural synchrony, such as in disorders like epilepsy. Another advantage is that there is an increased diversity of better attractors found at the end of self-optimization with inhibitory connections. In this way we overcome worries raised by Zarco and Froese (2018b) that convergence on single attractors could be limiting for applying the self-optimization process in cognitive robotics, which typically require the possibility to switch between multiple attractor configurations.

Future work

Further improvements to our current model can be made, especially because the real neurons of the worm tend to activate in a continuous rather than binary manner. In future work, we will therefore test the self-optimization connectome model with the dynamics of a continuous time recurrent neural network (CRTNN), which should in principle be possible (Zarco and Froese, 2018b). It would also be interesting to see what happens if we do not provide extra virtual or functional connections to the worm's connectome, and restrict Hebbian changes to the original anatomical connec-

tome only. Another possibility is to analyze neural coordination in terms of local clusters of neurons that are anatomically related (Nonet, 1999), rather than measuring success in terms of neural coordination across the entire connectome. Finally, further mathematical analysis of our model can be made, for instance by taking inspiration from the techniques employed by other Hopfield neural network models of sleep (Fachechi et al., 2019).

References

- Aguilera, M., Alquézar, C., and Izquierdo, E. J. (2017). Signatures of criticality in a maximum entropy model of the *C. elegans* brain during free behaviour. In Knibbe, C., Beslon, G., Parsons, D., Misevic, D., and Rouzaud-Cornabas, J., editors, *ECAL 2017: The Fourteenth European Conference on Artificial Life*, pages 29–35, Lyon, France. MIT Press.
- Albertson, D. G. and Thomson, J. N. (1976). The pharynx of *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 275(938):299–325.
- Beer, R. D. and Izquierdo, E. J. (2016). Propagation of rhythmic dorsoventral wave in a neuromechanical model of locomotion in *Caenorhabditis elegans*. In Gershenson, C., Froese, T., Siqueiros, J. M., Aguilar, W., Izquierdo, E., and Sayama, H., editors, *ALIFE 2016: The Fifteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 544–545, Cancún, Mexico. MIT Press.
- Capano, V., Herrmann, H. J., and de Arcangelis, L. (2015). Optimal percentage of inhibitory synapses in multi-task learning. *Scientific Reports*, 5:9895. doi:10.1038/srep09895.
- Fachechi, A., Agliari, E., and Barra, A. (2019). Dreaming neural networks: forgetting spurious memories and reinforcing pure ones. *Neural Networks*, 112:24–40.
- Froese, T. and Manzanilla, L. R. (2018). Modeling collective rule at ancient teotihuacan as a complex adaptive system: Communal ritual makes social hierarchy more effective. *Cognitive Systems Research*, 52:862–874.
- Girard, L. R., Fiedler, T. J., Harris, T. W., Carvalho, F., Antoshechkin, I., Han, M., Sternberg, P. W., Stein, L. D., and Chalfie, M. (2006). Wormbook: the online review of *Caenorhabditis elegans* biology. *Nucleic Acids Research*, 35:D472–D475. doi:10.1093/nar/gkl894.
- Hattori, Y., Suzuki, M., Soh, Z., Kobayashi, Y., and Tsuji, T. (2012). Modeling of the pharyngeal muscle in *Caenorhabditis elegans* based on FitzHugh-Nagumo equations. *Artificial Life and Robotics*, 17(2):173–179.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.
- Izquierdo, E. J. (2018). Role of simulation models in understanding the generation of behavior in *C. elegans*. *Current Opinion in Systems Biology*, 13:93–101.
- Izquierdo, E. J. and Beer, R. D. (2015). An integrated neuromechanical model of steering in *C. elegans*. In Andrews, P., Kaves, L., and Doursat, R., editors, *ECAL 2015: The Thirteenth European Conference on Artificial Life*, pages 199–206, York, UK. MIT Press.
- Jarrell, T. A., Wang, Y., Bloniarz, A. E., Brittin, C. A., Xu, M., Thomson, J. N., Albertson, D. G., Hall, D. H., and Emmons, S. W. (2012). The connectome of a decision-making neural network. *Science*, 337(6093):437–444.
- Kitano, H., Hamahashi, S., and Luke, S. (1998). The perfect *C. elegans* project: An initial report. *Artificial Life*, 4(2):141–156.
- Nelson, M. D. and Raizen, D. M. (2013). A sleep state during *C. elegans* development. *Current Opinion in Neurobiology*, 23(5):824–830.
- Nonet, M. L. (1999). Visualization of synaptic specializations in live *C. elegans* with synaptic vesicle protein-GFP fusions. *Journal of Neuroscience Methods*, 89(1):33–40.
- Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the *Caenorhabditis elegans* neuronal network. *PLOS Computational Biology*, 7(2):1–21.
- Waterston, R. and Sulston, J. E. (1998). The Human Genome Project: Reaching the Finish Line. *Science*, 282(5386):53–54. doi:10.1126/science.282.5386.53.
- Watson, R. A., Mills, R., and Buckley, C. L. (2011). Global adaptation in networks of selfish components: Emergent associative memory at the system scale. *Artificial Life*, 17(3):147–166.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440.
- White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society, Biological Sciences*, 314(1165):1–340. doi:10.1098/rstb.1986.0056.
- Winkler, D. A., Burden, F. R., and Halley, J. D. (2009). Predictive mesoscale network model of cell fate decisions during *C. elegans* embryogenesis. *Artificial life*, 15(4):411–421.
- Woodward, A., Froese, T., and Ikegami, T. (2015). Neural coordination can be enhanced by occasional interruption of normal firing patterns: A self-optimizing spiking neural network model. *Neural Networks*, 62:39 – 46.
- Zarco, M. and Froese, T. (2018a). Self-modeling in Hopfield neural networks with continuous activation function. *Procedia Computer Science*, 123:573–578.
- Zarco, M. and Froese, T. (2018b). Self-optimization in continuous-time recurrent neural networks. *Frontiers in Robotics and AI*, 5:96. doi:10.3389/frobt.2018.00096.

DNN Architecture for High Performance Prediction on Natural Videos Loses Submodule’s Ability to Learn Discrete-World Dataset

Lana Sinapayen^{1,*}, Atsushi Noda^{1,*}

¹Sony Computer Science Laboratories, Tokyo, Japan. *These authors contributed equally to the paper.
lana.sinapayen@gmail.com

Abstract

Is cognition a collection of loosely connected functions tuned to different tasks, or is it more like a general learning algorithm? If such an hypothetical general algorithm did exist, tuned to our world, could it adapt seamlessly to a world with different laws of nature? We consider the theory that predictive coding is such a general rule, and falsify it for one specific neural architecture known for high-performance predictions on natural videos and replication of human visual illusions: PredNet. Our results show that PredNet’s high performance generalizes without retraining on a completely different natural video dataset. Yet PredNet cannot be trained to reach even mediocre accuracy on an artificial video dataset created with the rules of the Game of Life (GoL). We also find that a submodule of PredNet, a Convolutional Neural Network trained alone, has excellent accuracy on the GoL while having mediocre accuracy on natural videos, showing that PredNet’s architecture itself might be responsible for both the high performance on natural videos and the loss of performance on the GoL. Just as humans cannot predict the dynamics of the GoL, our results suggest that there could be a trade-off between high performance on sensory inputs with different sets of rules.

Introduction

In a world where many tasks have been automated to quasi-perfection, the next big goal for Artificial Intelligence (AI) is Artificial General Intelligence (AGI) (Wang and Goertzel (2007)). Beyond domain-specific automation, AGI is often defined as AI with human-level performance, able to generalize its knowledge across different domains (Adams et al. (2012); Wang and Goertzel (2007)). Candidate algorithms to realize AGI vary with the evolution of the field (Goertzel (2010, 2014)), and recent theories include a combination of Deep Learning Networks to emulate human cortical networks (Yamakawa et al. (2017)), neural computation at the edge of chaos (Smith (2016); Cocchi et al. (2017); Garson (1996)), and of course different implementations of predictive coding (Hawkins and Blakeslee (2007); Friston (2010); Van De Ven and Schouten (2010)).

Here we are interested in predictive coding: the idea that brains generate models of the world by learning to predict their own sensory inputs. Although the specific mechanism

is up to debate (Friston (2009); Garalevicius (2007)), predictive coding has been found to take place in the human nervous system (Baldeweg (2006); Hosoya et al. (2005); Rao and Ballard (1999)), especially for visual and auditory processing. In many experiments, specific sensory illusions such as the auditory oddball effect are interpreted as hallmarks of predictive coding (Schindel et al. (2011)).

A recently proposed Deep Neural Network architecture, PredNet (Lotter et al. (2016)), has not only demonstrated high performance on natural video prediction, but has been shown to be susceptible to some of the same visual illusions as human beings (Watanabe et al. (2018)) as a side effect of its predictive abilities, despite contemporary explanations of these illusions not relying on predictive coding mechanisms. No one claims that PredNet accurately recreates the brain processes leading to predictive coding, but PredNet is currently the state of the art architecture replicating both the main consequences and side effects of predictive coding in human vision. Can PredNet be used as a general prediction machine, even in visual worlds that humans cannot predict, or is the algorithm’s implementation so attuned to human visual processing that it fails in the same circumstances as humans do? In short, does high performance on natural datasets cause a loss in generality for this specific architecture?

To test this hypothesis, we choose a visual task where simple local spatial rules lead to dynamics that are hard to predict for human beings: the Game of Life automaton (GoL) (Gardener (1970), summarized by Izhikevich et al. (2015)). At each time step, the GoL updates each cell depending only on the state of its direct neighbors at the previous timestep. It is simple enough to be predicted with accuracy by a simple Convolutional Neural Network (Rapp (2015)). Most dynamics are discrete but some have the feeling of continuous dynamics, for example the “glider”: a bird-like pattern that moves in a constant direction. The produced patterns can be hard to predict; even knowing the underlying rules, humans make mistakes in prediction. We can reasonably extrapolate that prediction without knowing these rules would be even harder for humans. The main dif-

ferences between a natural video such as those in the KITTI dataset (Geiger et al. (2013)) and a GoL-generated video are as follows:

- Theoretically, the GoL requires only 1 time step of memory for perfect prediction. Natural videos can show objects with speed or acceleration, which require several time steps to deduce.
- Natural videos have spatio-temporal continuity: objects maintain their shapes and there is no drastic change from one frame to the other. In the GoL, there is no continuity from one frame to the other and each cell is separate from the others. The GoL is a discrete world.
- Rules are extremely local in the GoL: a cell's next state only depends on its immediate neighbors.

If PredNet had succeeded at this task, it would suggest that the implementation of PredNet captures something fundamental beyond natural rules, and the idea of predictive coding as a general cross-domain learning rule would be strengthened. But our results show that PredNet is attuned to human performance even in its failure modes, as its sensitivity to visual illusions suggested. It comforts this architecture in its place as a replicator of human visual performance, but it also means that the improved performance on natural tasks comes to the cost of performance on tasks that can be solved by simpler networks. These results stack two different learning architectures (the human brain and PredNet) against predictive coding as a general cross-domain learning rule.

Methods

The source code used in this paper is available at https://github.com/LanaSina/prednet_gol. The datasets are available at https://figshare.com/projects/PredNet_Game_of_Life/60971.

PredNet

PredNet is a Deep Learning Neural Network with an architecture based on the principles of predictive coding (Fig. 1). The network is made of hierarchical layers. Each layer contains several modules, respectively for input, representation, prediction, and error calculation. The input to the first layer is the original image frame at time t ; the error between the prediction at t and the frame at $t + 1$ is passed as input to the next layer, which must therefore learn to predict the error signal from the layer below it. The representations from the upper layer are sent as feedback to the lower layer.

For our experiments, we use the code provided by Lotter et al. (2016) at this address: <https://github.com/coxlab/prednet>. The network has 3 layers, its convolutional modules have filters of size 3×3 pixels. We tested

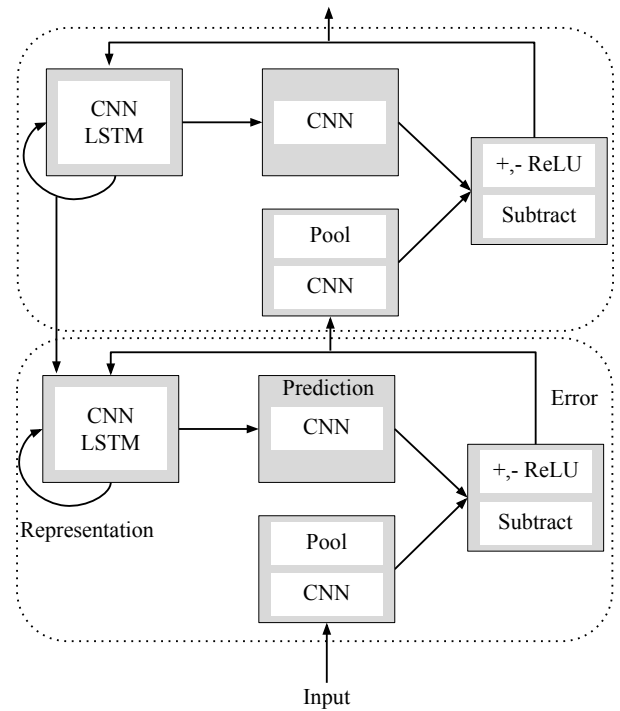


Figure 1: **The architecture of PredNet.** Figure modified from Lotter et al. (2016). The network is divided in modules (solid lines) that are stacked into layers (dashed lines). The input to the first layer is the original image frame at t ; the error between the prediction at t and the frame at $t+1$ is passed as input to the next layer. There are 3 layers in total (only 2 represented here.)

generalization between datasets by evaluating the KITTI-trained model on other datasets. These experiments are referred to as experiments **without retraining** in the remainder of this paper. For the experiments **with retraining**, we train and evaluate PredNet on one given dataset (as is typical in Deep Learning experiments, training and testing data are separate but extracted from one same dataset).

Simple CNN

We use an Auto-Encoder Convolutional Neural Network, a minimum model that can learn the GoL rules (referred to as simple CNN in the remainder of this paper). The model is similar to the CNNs in several modules used in the PredNet architecture (notably the Prediction module) and uses the same filter size. We don't use a Long Short Term Memory (LSTM) since it is unnecessary to predict GoL images. The architecture is as follows: the encoder consists of one convolutional layer and maps pixels to a latent representation. The filter size is 3×3 and the stride is set at 1. Zero padding is done to deal with boundary conditions of the image. The decoder uses the transposed architecture. The acti-

vation function is ReLU. We implemented this model using Chainer (Tokui et al. (2015)) and trained it with the negative log-likelihood function of the Bernoulli distribution. In this model, the data at the episode boundaries is excluded from the training and evaluation data. The number of epochs \times the size of training data is slightly less than one fifth of that of Prednet’s KITTI learning.

Datasets

We use two existing datasets composed of videos from the real world (called “natural datasets” in this paper) and one dataset of videos generated from the GoL (called “artificial dataset”).

- Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) Dataset The KITTI dataset is composed of videos filmed from a car driving on a road (Geiger et al. (2013)). The videos show traffic and pedestrians from the point of view of the driving car. We use the simple raw video part of the dataset (20 fps). As in the original PredNet paper, frames of the videos are down-sampled to 128×160 pixels. We used the processing script for outputting frames from videos made available by (Lotter et al. (2016)). The script is also available in the repository published for this paper.

- First Person Social Interaction (FPSI) Dataset The FPSI dataset (Fathi et al. (2012)) is composed of videos filmed from a first-person point of view. The videos were shot by fixing cameras to the head of several people who spent a day at a theme park. The original dataset’s url <http://cpl.cc.gatech.edu/projects/FPSI/> had become unresponsive at the time of the writing of this paper, so we used the video made available by Watanabe et al. (2018) at https://figshare.com/articles/Training_data/5483668/1. We also used their script for extracting and resizing images to 128×160 pixels, but found that the original video was encoded with the wrong FPS parameter. The corrected script is available at our published repository (output: 15 fps).

- Game of Life (GoL) Dataset We use Conway’s Game of Life (GoL) (Gardener (1970); Izhikevich et al. (2015)) as artificial dataset to test PredNet. The GoL is a 2D 0-player game where an initial state, decided by the programmer, automatically changes every timestep according to a set of rules. The GoL is played on a grid where each cell is in one of two possible states: *alive* (white cells in our implementation) or *dead* (black cells). The flow of time is discrete, and the state transitions are deterministic and memoryless: each cell has 8 neighbors (4 cells adjacent orthogonally and 4 diagonally); the state the cell depends on how many neighbors are *alive*. If there are exactly 2 *alive* cells in the neighborhood, the state of the central cell does not change. If there are exactly 3 *alive* cells in the neighborhood, the cell is *alive*

in the next frame. For any other number of living cells in the neighborhood, the next state of the cell is *dead*. These are the only rules. The interest of the GoL is that simple rules lead to complex patterns. We will briefly introduce the “glider” as an example used in experiments described later. The glider is a moving pattern composed of 5 cells. The movement of the glider has four stages; after 4 time steps, the glider has shifted one cell in the diagonal direction. In the GoL, there are patterns that move while keeping their shape like the glider, patterns that periodically repeat states, and so on.

We generate the video dataset as below. When generating an initial state, 10% of the cells are randomly set to *alive*; the other cells are *dead*. Each pixel represents a cell in the 128×160 images. There are no boundary conditions. The state often becomes fixed as time passes, so we generate a new initial state every 10 steps. We generate 10,000 images as training data.

Results

PredNet performance on KITTI, FPSI and GoL without retraining

We investigate how well the performance of PredNet trained exclusively on the KITTI dataset transfers to other datasets without retraining. We evaluate this model on the original KITTI dataset, on the natural video dataset FPSI, on random GoL patterns, and finally, on a GoL glider. In this experiment, the output of PredNet for the GoL was totally black frames, suggesting that the patterns were treated as noise. We therefore re-generated the GoL frames as 16×20 images and scaled them up to 8 times the original size.

As shown on Fig. 2-a), b), and Table 1, the performance transfers well between natural datasets. PredNet trained on the KITTI dataset still has better performance on the FPSI dataset than a simple “copy the last frame” model. This result is a testament to the robustness of the PredNet architecture. On the other hand, as shown on Fig. 2-c) and d), the performance does not transfer to datasets with artificial rules. As expected, the pretrained PredNet cannot predict the GoL, or even gliders. Even if gliders have a partly translational motion, the dynamics of the GoL are too far from the simple translations and rotations that might be responsible for the majority of variations in natural videos. In addition, we see an interesting error on frame 3 at Fig. 2-b): the translational motion of the cap that appeared at frame 2 is extrapolated, leading to the prediction of a floating cap at frame 3. This model cannot know that the cap should be on a human head, even if the person wearing the cap can be seen a few frames earlier. This would require either excellent use of short term memory, existence of an internal world model, or high level inference.

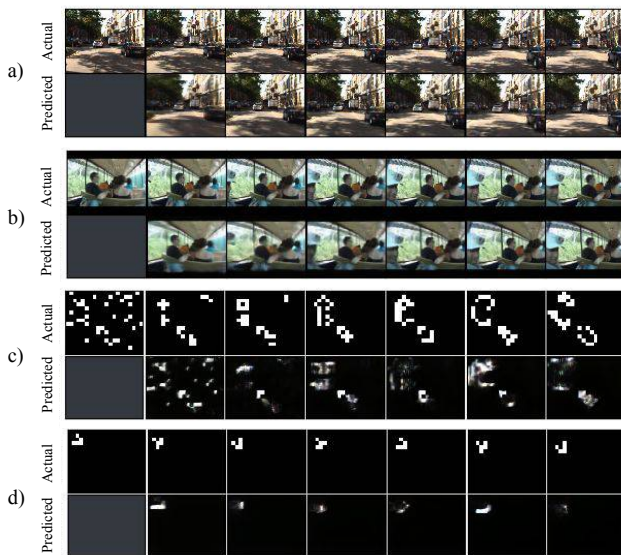


Figure 2: **Predictions generated by a PredNet trained on the KITTI dataset.** See Table 1 for measured performance. (a) Predictions for KITTI, the original training dataset. The performance is high. (b) Predictions for the FPSI dataset. The performance is also high. (c) Predictions for a random initial value of the GoL. The performance is poor, as expected. (d) Predictions for a GoL glider. The performance is also poor.

Retrained PredNet performance on the GoL

Since the performance of PredNet on natural images does not automatically transfer to the artificial dataset, in the first part of this experiment we re-train PredNet from scratch on the GoL dataset. Figures. 3 show zoomed-in results (clipped upper left images to make it easy to see). The network is unable to learn and performs poorly. This result is unexpected considering that this task is solved by a CNN as shown in the next section, and PredNet has several CNNs with the appropriate filter size of 3×3 in its architecture. Training on up-scaled images gave the same result. This suggests that high performance on natural videos comes to the cost of generality for videos exhibiting different types of dynamics (even if these dynamics are totally deterministic and predictable).

and 4,

Finally, we train and evaluate PredNet on a single sequence showing a glider moving from the upper left to the lower right of the frame. In this setting we use the same data for learning and evaluation (it is therefore a pure memorization task). As shown on Fig. 4, even with these extremely gentle conditions, PredNet is unable to make any predictions. Training on up-scaled images forced the network to have.

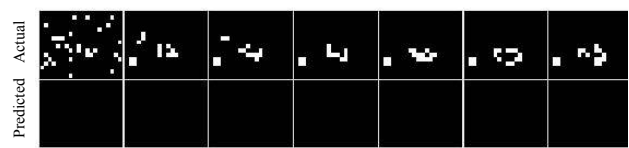


Figure 3: **Predictions generated by a PredNet trained and evaluated on random GoL patterns.** PredNet no longer outputs anything as predictions. The rules of the GoL seem not simply impossible to predict using PredNet, but impossible to learn.

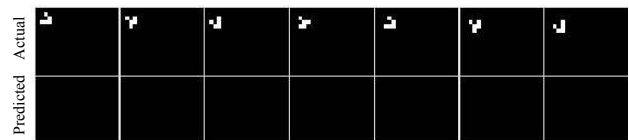


Figure 4: **Predictions generated by a PredNet trained only on the exact sequence shown here.** In this case, training data and test data are equal. Even for this simple task, PredNet no longer outputs anything as predictions.

Simple CNN performance on the GoL

We train the CNN on GoL videos generated from initial random patterns. We then test the CNN on two sequences: (a) a random pattern and (2) a glider pattern where a glider moves from the top left of the scene to the bottom right. As shown on Fig. 5, the CNN trained on the random patterns performs almost perfectly on both datasets, as previously reported in Rapp (2015). This result is not surprising considering that we fixed the surface of the convolution filters (receptive field of hidden neurons) to 3×3 cells: all the information necessary to predict the next step is contained into every individual filter. In addition, the rules are the same for the whole image, so that rules learned locally can be successfully applied anywhere in the image.

Simple CNN performance on KITTI

We train the simple CNN on the KITTI dataset. We do not expect good performance, as PredNet was implemented with the sole goal of achieving state of the art performance and simple CNNs are notoriously poor at natural video prediction. The results Fig. 6 show that indeed, the performance is poor. We stopped learning when the number of epochs was 10 because the value of loss did not fall any more. The output of the CNN resembles a fuzzy copy of the last presented frame. As recorded in Table 1, the simple CNN's error is an order of magnitude bigger than PredNet's error, and slightly lower than last frame copy. This result again shows a trade off, but opposite of PredNet: we have high performance on the artificial dataset and poor performance on the natural dataset.

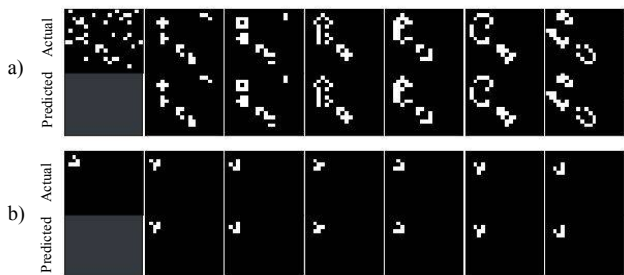


Figure 5: **Predictions generated by the CNN after training on random GoL patterns.** (a) Predictions from a random initial state. (b) Predictions for a initial state composed of one glider. The performance is close to perfect, an expected result considering that the convolutional filters are the same size as the local neighborhood of the GoL cells.



Figure 6: **Predictions generated by the CNN after training on the KITTI dataset.** The performance is poor, especially compared to PredNet (an expected result for natural videos).

Prednet-as-CNN: performance on the GoL

The goal of this experiment is to see if PredNet’s unexpected inability to predict discrete data might be due to implementation details (parameters or bugs), or if the architecture itself might be responsible for the inhibition of the ability of the inner modules to predict discrete data. We fix all the weights to their initial random values (i.e., we prevent learning on these weights by setting the learning rate to 0) except for one of the CNN modules. The resulting model, “Prednet-as-CNN”, is therefore structurally equivalent to a simple CNN: if it fails to learn GoL predictions, it will indicate that implementation details are to blame. If Prednet-as-CNN learns the GoL, it indicates that the failure of the original PredNet model is due to its architecture. The training data is the same as for the simple CNN experiment: random GoL patterns.

The results, shown on Fig. 7, show two noteworthy elements. First, the predicted cells are pink instead of white. This most likely suggests either an issue with the code, or less likely, an issue with the architecture introducing a random bias towards low loss on the red channel of the rgb input. Second, the network does output some predictions, not simply black frames. The performance indicates that the predictions are better than simple copies of the previous frame (Table 1), but to the naked eye they do mostly look like copies of the previous step. The results are the

Table 1: Mean squared error (MSE) of next frame predictions. As mentioned in the Dataset’s subsection, all datasets are set to the same pixel size, so it is meaningful to compare values side by side.

Dataset	Method	MSE
KITTI	Previous Frame	0.0256
	PredNet	0.0073
	Simple CNN	0.0164
FPSI	Previous Frame	0.0110
	PredNet (pretrained)	0.0059
GoL (random)	Previous Frame	0.0332
	PredNet (pretrained)	0.0330
	PredNet (re-trained)	0.0348
	PredNet-as-CNN	0.0265
	Simple CNN	0.0002

same for 150 (original parameter), 250, and 500 epochs, suggesting that no learning takes place beyond the pink-copy phase. This result is difficult to interpret, and additional experiments are needed; but this half failure (compared to the simple CNN), half success (compared to the unmodified PredNet and its blank predictions) suggests that at least part of the blame goes to the architecture itself.

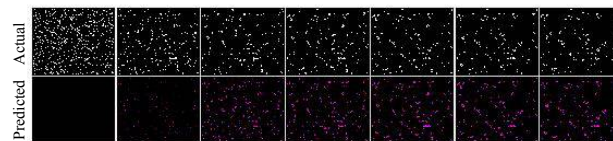


Figure 7: **Random GoL pattern prediction by the Prednet-as-CNN model.** The predicted cells are pink instead of white, suggesting an issue with the code, or an issue with the architecture introducing a random bias towards low loss on the red channel of the rgb input. Contrarily to PredNet, the network does output predictions and not simply black frames.

Discussion

We showed that PredNet can generalize without retraining on a completely different natural video dataset. Yet we also found that unmodified, PredNet cannot learn to predict the dynamics of the GoL. Worse yet, it cannot memorize a short 10-frame sequence of the GoL: a task that should be solvable rote memorization of the mapping between input and output. This result is surprising since a simple CNN with only one conv-deconv layer did learn the GoL rules, and PredNet was not able to learn despite internally containing modules that are themselves simple CNNs. When making PredNet structurally equivalent to a simple CNN (“Prednet-as-CNN”), the model improved its GoL predictions, albeit

still showing poor performance. This suggests that the architecture of PredNet, which makes it so good with continuous input, is also a cause of the inhibition of its submodules ability to predict discrete input.

This paper shows that there is at least one unsuitable task for the PredNet architecture as it is. In this models, past observations are abstracted and stored internally, and it seems that input images are shifted by an appropriate amount using estimated velocity and rotation information. This is especially striking in the prediction error on frame 3 in Fig. 2-b): the translational motion of the cap that appeared at frame 2 is extrapolated, leading to the prediction of a floating cap at frame 3. This explains why these models are good for continuous dynamics such as movement of objects in the real world. On the other hand, in the GoL world, every cell changes in a discrete manner according to the GoL rules. This can be captured by CNNs, but this ability seems to be somehow inhibited by the very architecture that makes PredNet good at shifting and rotating. To find out the cause of the difference in performance, we still need to test hybrid datasets that are missing some characteristics of real world video.

Note that we just assumed that PredNet, as current state of the art in video prediction, was a kind of "master algorithm" for natural videos; in reality its failure does not mean that all high-performance predictive algorithms would fail at predicting discrete datasets. Yet we have to wonder if a single architecture would be able to perform on both natural and artificial datasets, or if the trade-off in performance cannot be avoided. Supporting this latter hypothesis, the GoL is extremely difficult to predict for human brains; even its own creator, knowing the rules, had to use a physical board with go stones to compute states and still made errors. If there really is a divergence between optimizing predictions for natural and synthetic datasets, then when thinking about an hypothetical general artificial intelligence, we should also consider how to compensate for the counter-intuitive weaknesses that such superior models create for themselves compared to lower-performance models. More generally, it suggests that "prediction" does not have the same meaning in natural and artificial worlds.

References

- Adams, S., Arel, I., Bach, J., Coop, R., Furlan, R., Goertzel, B., Hall, J. S., Samsonovich, A., Scheutz, M., Schlesinger, M., et al. (2012). Mapping the landscape of human-level artificial general intelligence. *AI magazine*, 33(1):25–42.
- Baldeweg, T. (2006). Repetition effects to sounds: evidence for predictive coding in the auditory system. *Trends in cognitive sciences*.
- Cocchi, L., Gollo, L. L., Zalesky, A., and Breakspear, M. (2017). Criticality in the brain: A synthesis of neurobiology, models and cognition. *Progress in neurobiology*, 158:132–152.
- Fathi, A., Hodgins, J. K., and Rehg, J. M. (2012). Social interactions: A first-person perspective. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1226–1233. IEEE.
- Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127.
- Garalevicius, S. J. (2007). Memory-prediction framework for pattern recognition: Performance and suitability of the bayesian model of visual cortex. In *FLAIRS Conference*, pages 92–97.
- Gardener, M. (1970). Mathematical games: The fantastic combinations of john conway's new solitaire game" life,". *Scientific American*, 223:120–123.
- Garson, J. W. (1996). Cognition poised at the edge of chaos: A complex alternative to a symbolic mind. *Philosophical Psychology*, 9(3):301–322.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.
- Goertzel, B. (2010). Toward a formal characterization of real-world general intelligence. In *3d Conference on Artificial General Intelligence (AGI-2010)*. Atlantis Press.
- Goertzel, B. (2014). Artificial general intelligence: concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 5(1):1–48.
- Hawkins, J. and Blakeslee, S. (2007). *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan.
- Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047):71.
- Izhikevich, E. M., Conway, J. H., and Seth, A. (2015). Game of Life. *Scholarpedia*, 10(6):1816. revision #150735.
- Lotter, W., Kreiman, G., and Cox, D. (2016). Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning.
- Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79.
- Rapp, D. (2015). Learning game of life with a convolutional neural network. <https://danielrapp.github.io/cnn-gol/>. Accessed: 2019-02-28.
- Schindel, R., Rowlands, J., and Arnold, D. H. (2011). The oddball effect: Perceived duration and predictive coding. *Journal of Vision*, 11(2):17–17.
- Smith, L. S. (2016). Deep neural networks: the only show in town? a position paper for the workshop on can deep neural networks (dnns) provide the basis for artificial general intelligence (agi) at agi 2016, july 2016.
- Tokui, S., Oono, K., and Hido, S. (2015). Chainer: a next-generation open source framework for deep learning.

Van De Ven, A. and Schouten, B. A. (2010). A minimum relative entropy principle for agi. In *3d Conference on Artificial General Intelligence (AGI-2010)*. Atlantis Press.

Wang, P. and Goertzel, B. (2007). Introduction: Aspects of artificial general intelligence. In *Proceedings of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006*, pages 1–16. IOS Press.

Watanabe, E., Kitaoka, A., Sakamoto, K., Yasugi, M., and Tanaka, K. (2018). Illusory motion reproduced by deep neural networks trained for prediction. *Frontiers in psychology*, 9:345.

Yamakawa, H., Arakawa, N., and Takahashi, K. (2017). Reinterpreting the cortical circuit. In *Architectures for Generality & Autonomy Workshop at IJCAI*, volume 17.

Additional Details

Retrained PredNet performance on the upscaled glider

As in the Section “Retrained PredNet performance on the GoL”, we also upscaled the glider images and trained PredNet to memorize them. Fig. 8 shows that PredNet cannot predict perfectly even with these extremely gentle conditions. What it seems to have learned is a set of periodic translations of period 5 for simple patterns.

Simple CNN with smaller filters

When using 2×2 convolutional filters instead of 3×3 filters, the performance of the simple CNN model sharply declined as shown in Fig. 9. These filters are too small to represent the GoL rules: in the GoL, the next state of a cell is completely determined by the state of eight neighboring cells. Necessary information is contained in a 3×3 neighborhood.

Reversed video prediction

Although physical laws are time reversible, it is famously difficult for humans to predict videos where the flow of time is reversed. We tested whether a PredNet trained on regular-time videos from the KITTI dataset would be able to predict reverse-time videos. The model turned out to be just as good for regular- and reverse-time videos, with a MSE of 0.006 in both cases. Fig 10 shows the generated predictions.

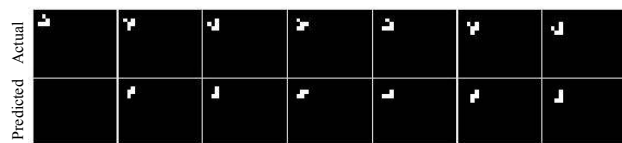


Figure 8: **Predictions generated by a PredNet trained only on the exact (upscaled glider) sequence shown here.** In this case, training data and test data are equal. Even for this simple task, PredNet seems to only be able to imperfectly predict the pattern, with a constant one-cell mistake. In reality, it seems that what was learned was a simple translation of elementary patterns of period 5.

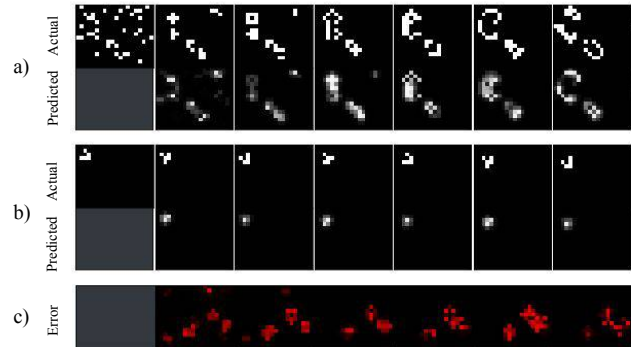


Figure 9: **Predictions with smaller convolutional filters.** The settings are the same as for Fig. 5, but the filter size is 2×2 instead of 3×3 . (a) Predictions from a random initial state. (b) Predictions for a initial state composed of one glider. (c) The difference between actual and predicted images. The performance is poor. The filters are too small to capture neighborhood information.



Figure 10: **Predictions generated on the FPSI dataset.** (a) Predictions on the regular-time video. (b) Predictions on the reverse-time video. The predictions are not exactly the same but the performance is similar in both cases.

Evolution of metamemory ability by artificial neural networks with neuromodulation

Yusuke Yamato¹, Reiji Suzuki¹ and Takaya Arita¹

¹Graduate School of Informatics, Nagoya University, yamato@alife.cs.i.nagoya-u.ac.jp

Introduction

Humans know whether, or how well, certain knowledge exists in their own memory. This subjective monitoring and control of one’s memory, *metamemory*, has been studied widely as a type of metacognition in cognitive psychology. This study, as a constructive approach, aims to evolve artificial neural networks that have a metamemory function. For this purpose, we use *evolved plastic artificial neural networks* (EPANNs) (Soltoggio et al., 2018). Specifically, we use *neuromodulation* (Fig. 1), that has been recognized as an essential element in cognitive and behavioral processes playing an important role in, for example, facilitating the evolution of learning, adapting to dynamic environments, and acquisition of mental representation. Using EPANNs, we showed in one of the evolutionary experiments that evolved neural networks clearly had capacity for metamemory (Sudo et al., 2014), in the sense that they satisfy a measure based on a type of delayed matching-to-sample tasks (DMTSS) (Hampton, 2001) that were developed to ask whether monkeys can have metamemory or not.

However, metamemory is not something so simple (Call, 2010), because it is extremely difficult to conclude that a monkey subject can monitor her memory just by observing her behavior. That difficulty depends also on the difficulty in defining metamemory in the first place. In principle, we could analyze and understand all mechanisms and processes involved in artificial neural networks evolved in simulation, unlike the cases of using living subjects. We take the previous evolutionary experiments (Sudo et al., 2014) as a starting point, and critically analyze the evolved networks and then refine the measure to exclude the evolution of networks whose mechanism or process seems different from that of metamemory. Our study scheme is based on the repetition of, evolutionary experiments, analysis of the evolved networks, and refinement of the measure.

Methodology

Fig. 2 shows an overview of the task (Sudo et al., 2014), composed of 4 phases. In the study phase, an agent receives a target pattern composed of 5 binary digits. The delay phase

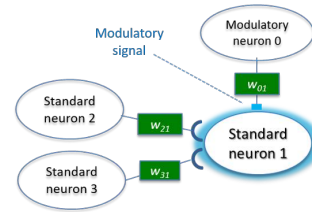


Figure 1: Metamemory.

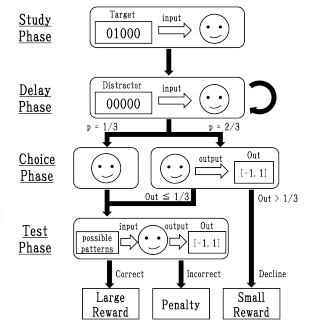
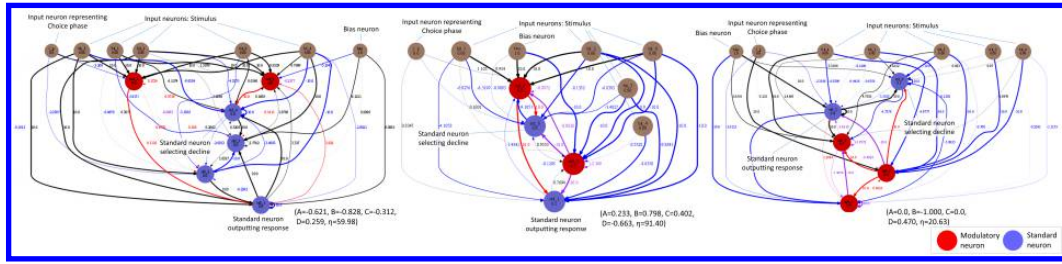


Figure 2: The delayed match-to-sample task.

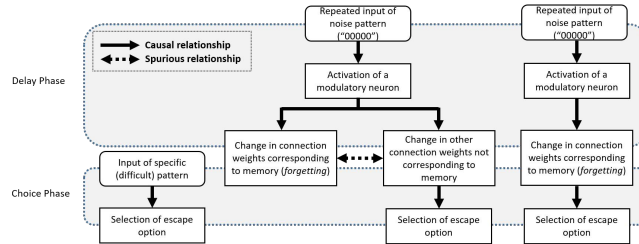
follows, in which the agent receives 00000 as a distractor pattern on several occasions. Then, with a probability of 2/3, the choice phase starts during which the agent receives a signal meaning that it is in that phase. One output from the agent will be interpreted as the intention to decline the trial. We set the agent receives a small reward (0.3), and the trial ends. On the other hand, with a probability of 1/3, the choice phase is skipped as a compulsory condition. In the test phase, the agent receives all patterns one by one in random order. An output is interpreted as a response for each pattern. If it matches the target pattern it memorized in the study phase, the agent is rewarded with a large reward (1.0). Otherwise, it is rewarded with nothing.

The neural network of an agent has 7 inputs and 2 outputs. The topology of the networks evolves while keeping the number of the neurons (including standard and modulatory neurons but excluding input neurons) not more than 16. Modulatory neurons are different from standard neurons, which affect the connection of standard neurons by changing their learning rate. We used an evolution strategy (ES) (Bäck et al., 1997) for evolution of topology and connection weights of neural networks, which is basically the same as the one used in Soltoggio et al. (2008).

We defined the following three measures of metamemory one by one responding to the repetition in the study scheme, the one used in the escape response paradigm, the one which



(a) Neural networks which meet BM (left), NM1 (center) and NM2 (right) (blue node: standard neuron, red node: modulatory neuron).



(b) Mechanisms of evolved neural networks satisfying BM (left), NM1 (center) and NM2 (right).

Figure 3: Evolved agents and mechanisms.

describes the minimum requirement for the neural network responding to the distractor pattern and the more strict measure of metamemory.

- Behavioral measure (BM): There is a difference in accuracy between chosen and forced trials in the escape response paradigm.
- Neural measure 1 (NM1): Behavioral measure is met, but not by changing the behavior according only to particular stimuli configurations.
- Neural measure 2 (NM2): Neural measure 1 is met, which is based on the self-reference on the memory.

Result

We obtained 17 successful trials among 20 in the sense that average fitness clearly increased through evolution. We analyzed the behavior of the evolved agents in the successful trials, and found agents that met BM and NM1 (Fig. 3(a)(left) and Fig. 3(a)(center)). However, we found that the evolved network (Fig. 2(b)) does not meet NM2 as it selects the escape option as a result of a spurious relationship shown in Fig. 3(b)(center). We further performed an extended evolutionary experiment targeted at NM2, and modified manually the best evolved network to successfully obtain a network which meets NM2 (Fig. 3(a, b)(right)). We investigated its network dynamics in detail and found that neuromodulation plays a crucial role in the evolved metamemory ability.

Conclusion

This paper reports on our attempt to evolve artificial neural networks with neuromodulation, that have a metamem-

ory function, based on the repetition of evolutionary experiments, analysis of the evolved networks and refinement of the measure. A straightforward direction is to further refine NM2 by using other measures based not on DMTs but other paradigms. Also, examining the generality of proposed measures would be interesting by applying them to cutting-edge cognitive models related to metamemory. We believe that our methodology contributes to the understanding of human metamemory and realization of artificial consciousness.

Acknowledgements

This work was supported in part by JSPS/MEXT KAKENHI: JP17H06383 in #4903.

References

- Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). *Handbook of evolutionary computation*. CRC Press.
- Call, J. (2010). Do apes know that they could be wrong? *Animal cognition*, 13(5):689–700.
- Hampton, R. R. (2001). Rhesus monkeys know when they remember. *Proc. of the National Academy of Sciences*, 98(9):5359–5362.
- Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P., and Floreano, D. (2008). Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. *Proc. of the 11th international conference on artificial life (Alife XI)*, pages 569–576.
- Soltoggio, A., Stanley, K. O., and Risi, S. (2018). Born to learn: The inspiration, progress, and future of evolved plastic artificial neural networks. *Neural Networks*, 108:48–67.
- Sudo, M., Suzuki, R., and Arita, T. (2014). Can agents with neuromodulation know when they remember? *Proc. of the 19th Int'l Symp. on Artificial Life and Robotics*, pages 330–334.

Robotic models of obstacle avoidance in bats

Carl Bou Mansour¹, Elijah Koreman², Dennis Laurijssen³
Jan Steckel³, Herbert Peremans³, and Dieter Vanderelst¹

¹University of Cincinnati
²Cornell University
³Antwerp University
boumancl@mail.uc.edu

Abstract

Echolocating bats can avoid obstacles in complete darkness relying on their sonar system. Under experimental conditions, these animals can infer the 3D position of obstacles. However, in cluttered and complex environments their ability to locate obstacles is likely to be largely reduced, and they might need to rely on more robust cues that do not degrade as the complexity of the environment increases. Here, we present a robotic model of two hypothesized obstacle avoidance strategies in bats, both of which model observed behavior in bats: a Gaze Scanning Strategy and a Fixed Head Strategy. Critically, these strategies only employ interaural level differences and do not require locating obstacles. We found that both strategies were successful at avoiding obstacles in cluttered environments. However, the Fixed Head Strategy performed better. This indicates that acoustic gaze scanning, observed in hunting bats, might reduce obstacle avoidance performance. We conclude that strategies based on gaze scanning should be avoided when little or no spatial information is available to the bat, which corresponds to recent observations in bats.

Introduction

Echolocating bats rely on their biosonar systems to avoid obstacles in complex environments (Griffin and Galambos, 1941). Several studies have documented that a bat's acoustic gaze often deviates from its flight direction. In particular, Ghose and Moss (2006) and Falk et al. (2014) found that, while hunting for prey, the bat's flight direction is determined by its gaze direction through a Delayed Linear Adaptive Law (DLAL): the bat's flight direction follows its gaze direction with a delay.

Recently, we have used simulations to propose a sensorimotor model of the prey capture by echolocating bats (Vanderelst and Peremans, 2018). The simulations in that paper incorporated the DLAL. The simulated bat was assumed to steer its gaze as to keep the prey in the center of its field of view. As in the experiments reported by Ghose and Moss (2006) and Falk et al. (2014), the flight direction followed the gaze direction. To assess the contribution of the DLAL to prey capture, we also ran simulations in which the head and body were rigidly coupled. Based on these results, we

concluded that a loose coupling between the flight direction and gaze direction allows bats to keep erratically moving prey in their field of view, and thereby, increases the probability of successful prey capture. This previous work hinted at a clear functional advantage of a loose coupling between flight and gaze direction through a DLAL during hunting. Here, we test whether a combination of acoustic gaze scanning and the DLAL is also beneficial to obstacle avoidance, using robotic experiments. We refer to the obstacle avoidance strategy as the Gaze Scanning Strategy.

In addition, to the Gaze Scanning Strategy we also evaluated a second strategy: the Fixed Head Strategy. Under the Fixed Head Strategy, the acoustic gaze and the flight direction are always aligned. This behavior has also been observed in bats (Knowles et al., 2015).

Methods

We instrumented a differential drive robot (fig. 1a,b) with a bat-like sonar system consisting of a narrowband ultrasonic emitter and two microphones, acting as ears. The sonar system was mounted on a pan-tilt system, allowing it to rotate with respect to the drive direction of the robot (fig. 1a). The microphones were embedded in 3D printed artificial pinnae to give the sonar system a realistic bat-like directionality. The robot emitted ultrasonic calls at a rate mimicking the pulse interval of bats. In addition, the model mimicked the speed and aerodynamic constraints of bats. The speed, head rotation, and body rotation corresponded with the flight dynamics of the *Eptesicus fuscus*. Echoes were processed using a model of the bat's cochlea (Wiegbe, 2008). We used the output of the cochlear model to implement two obstacle avoidance strategies in bats, the Gaze Scanning Strategy and the Fixed Head Strategy:

1. Gaze Scanning Strategy: Under this strategy, the interaural level difference was used to steer the gaze direction of the robot. The robot turned its sonar system towards the side from which the weakest echoes returned. The body followed the direction of the gaze aim, with a delay, as given by the DLAL. This strategy mimics the steering-by-hearing strategy observed by Ghose and Moss (2006)

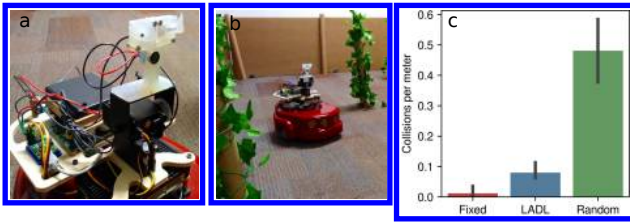


Figure 1: The robot (a) in one of the environments (b). (c) Obstacle avoidance results for the Fixed Head Strategy, Gaze Scanning Strategy and a Random Walk (as control).

and Falk et al. (2014).

2. Fixed Head Strategy: Under this strategy, the interaural level difference was used to steer the body rotation of the robot directly. The robot turned away from the direction from which it received the loudest echo(es). The gaze direction was always aligned with the driving direction of the robot. This strategy models a bat not using acoustic gaze steering (as observed by, e.g., Knowles et al., 2015)

The robot's ability to avoid obstacles was tested in a 3×4 m rectangular arena bounded by 50 cm high corrugated cardboard panels (fig. 1b). The obstacles consisted of cardboard tubes with a diameter of 4 cm. The tubes were wrapped in artificial ivy causing them to return multiple overlapping echoes. Our testing arena mimicked experimental conditions under which bat echolocation behavior has been studied before (for example, Falk et al., 2014; Knowles et al., 2015). Moreover, This environment can be assumed to model demanding obstacle avoidance conditions for bats (Falk et al., 2014; Knowles et al., 2015) under which localization of obstacles is not feasible (Warnecke et al., 2018; Knowles et al., 2015).

Results and Conclusion

Gaze scanning in bats has been likened to saccadic eye movements in mammals (Surlykke et al., 2009). Planning saccades requires the availability of spatial (angular) information. In the visual system, angular information is directly available. In contrast, angular information needs to be inferred computationally in hearing systems. In fact, bats might be assumed to often operate under cluttered conditions where computing angular information is impossible (Vanderelst et al., 2015). Under these conditions, only minimal spatial information, such as interaural level differences, might be available to bats to guide gaze scanning. The robust strategies implemented here are compatible with this kind of coarse spatial information.

In spite of their simplicity and the demanding environment, both strategies – observed in bats – successfully steered the robot away from obstacles (as compared with a Random Walk). However, we found that the Fixed Head

Strategy outperformed the Gaze Scanning Strategy (fig. 1c). We conclude that in spite of the benefit of this strategy during hunting (Ghose and Moss, 2006; Vanderelst and Peremans, 2018), (1) the delay introduced by the Gaze Scanning Strategy and (2) the fact that this strategy often looks away from the direction of driving makes it less suitable to navigate complex environments dense with obstacles.

We conclude that the Gaze Scanning Strategy can support obstacle avoidance using low-level, coarse spatial information to direct the gaze. However, the advantage of the Fixed Head Strategy leads us to conclude that gaze movements might reduce obstacle avoidance performance in highly cluttered environments. Our results would predict that the Gaze Scanning Strategy is traded for a Fixed Head Strategy under cluttered conditions. Indeed, Knowles et al. (2015) did not observe gaze scanning behavior in their experiments when flying bats through a matrix of chains.

References

- Falk, B., Jakobsen, L., Surlykke, A., and Moss, C. F. (2014). Bats coordinate sonar and flight behavior as they forage in open and cluttered environments. *Journal of Experimental Biology*, 217(24):4356–4364.
- Ghose, K. and Moss, C. F. (2006). Steering by Hearing: A Bat's Acoustic Gaze Is Linked to Its Flight Motor Output by a Delayed, Adaptive Linear Law. *Journal of Neuroscience*, 26(6):1704–1710.
- Griffin, D. R. and Galambos, R. (1941). The sensory basis of obstacle avoidance by flying bats. *Journal of Experimental Zoology*, 86(3):481–506.
- Knowles, J. M., Barchi, J. R., Gaudette, J. E., and Simmons, J. A. (2015). Effective biosonar echo-to-clutter rejection ratio in a complex dynamic scene. *The Journal of the Acoustical Society of America*, 138(2):1090–1101.
- Surlykke, A., Ghose, K., and Moss, C. F. (2009). Acoustic scanning of natural scenes by echolocation in the big brown bat, *Eptesicus fuscus*. *Journal of Experimental Biology*, 212(7):1011–1020.
- Vanderelst, D., Holderied, M. W., and Peremans, H. (2015). Sensorimotor model of obstacle avoidance in echolocating bats. *PLoS computational biology*, 11(10):e1004484.
- Vanderelst, D. and Peremans, H. (2018). Modeling bat prey capture in echolocating bats: The feasibility of reactive pursuit. *Journal of theoretical biology*, 456:305–314.
- Warnecke, M., Macías, S., Falk, B., and Moss, C. F. (2018). Echo interval and not echo intensity drives bat flight behavior in structured corridors. *The Journal of Experimental Biology*, 221(24):jeb191155.
- Wiegrefe, L. (2008). An autocorrelation model of bat sonar. *Biological Cybernetics*, 98(6):587–595.

A-life Evolution with Human Proxies

Vadim Bulitko and Kacy Doucet and Daniel Evans
Hope Docking and Mac Walters and Marilene Oliver
Julian Chow and Shelby Carleton and Natali Kendal-Freedman

University of Alberta, Edmonton, AB, Canada
bulitko@ualberta.ca

Abstract

Recent successes in Artificial Intelligence (AI) use machine learning to produce AI agents with both hand-engineered and procedurally generated elements learned from large amounts of data. As the balance shifts toward procedural generation, how can we predict interactions between such agents and humans? We propose to use Artificial Life to study emergence of group behaviours between procedurally generated AI agents and humans. We simulate Darwinian evolution to procedurally generate agents in a simple environment where the agents interact with human-controlled avatars. To reduce human involvement time, we machine-learn another set of AI agents that mimic human avatar behaviours and run the evolution with such human proxies instead of actual humans. This paper is an update on the on-going project.

1 Introduction

Artificial Intelligence (AI) is increasingly present in our lives in the form of smartphone assistants, smart appliances, self-driving cars, non-player characters in video games, etc. Consequently, emergent interactions between AI agents and humans can significantly affect our lives and help us address various societal challenges. Artificial life (A-life) is a powerful setting in which we can study behaviours emerging from interactions among AI agents. By adding human-controlled avatars to an A-life environment we can additionally study interactions between A-life and humans to gain insight into the continual impact of AI on society.

Most recent advances in AI use machine learning wherein the AI agents become progressively smarter by learning from existing data and their own experiences at various temporal scales. Evolutionary algorithms can learn across generations while reinforcement learning can learn from the agent's experience within its lifetime. Seminal work by Ackley and Littman (1991) combined two techniques into Evolutionary Reinforcement Learning.

Models of the evolution of AI agents in society must be informed by human involvement. As evolution can take many generations and human time is limited, we proposed in our earlier paper (Bulitko et al., 2018) to derive AI-controlled proxy agents to mimic human-controlled avatars in an A-life environment. Such proxies would be machine learned

from traces of actual human behaviour. These proxies could substitute for humans within the evolution for a set number of generations, with human-controlled avatars periodically re-introduced into the A-life environment where their behaviours are recorded to train the next set of proxies.

In this paper we present an update on this on-going project, including early proxy-creation results and the use of our approach in an interactive art installation.

2 Problem Formulation

The specific problem we focus on in this paper is to accelerate human-informed evolution of AI agents in an A-life setting. Our goal is to maximize the resulting reduction in evolution time while minimizing the loss of accuracy of the resulting AI behaviours relative to the baseline (i.e., using actual humans throughout the evolution).

To illustrate, consider evolving non-playable characters (NPCs) for a video game such as *Darwin's Demons* (Soule et al., 2017) or *No Man's Sky Next* (Hello Games, 2018). Suppose doing so with actual humans controlling avatars takes 100 hours for the NPCs to evolve to respond to the player in a desired fashion. Alternatively, we can substitute proxy agents for actual humans, reducing the evolution time down to, say, 10 hours (a 10x speed-up). However, since these NPCs were evolved against human proxies (i.e., *approximations* of humans), their evolved behaviour may be different from desired behaviour achieved in the former case. Our goal is thus to increase the speed-up while keeping both evolved response behaviours close.

3 Related Work

Surrogate models replace the actual, expensive fitness function with a more tractable approximation. Such models can be machine-learned (Kim et al., 2014; Rawal and Miikkulainen, 2018). Then evolution can be run with the surrogate model instead of the actual fitness function. This approach does not directly apply to our problem since the type of evolution we conduct in an A-life environment is asynchronous and does not have discrete generations on which a fitness function is used to sort the population towards forming the

next generation. Instead, A-life agents reproduce at will and the fitness of an agent is implicit in the environment (Ackley and Littman, 1991; Bulitko et al., 2017).

4 Learning Human Proxies

We are implementing the approach proposed by Bulitko et al. (2018) in a simple Netlogo environment (Wilensky, 1999). The environment is a modification of our previous work (Bulitko et al., 2017; Soares et al., 2018) and involves predator and prey AI agents on a two-dimensional 48×48 rectangular grid (Figure 1, left).

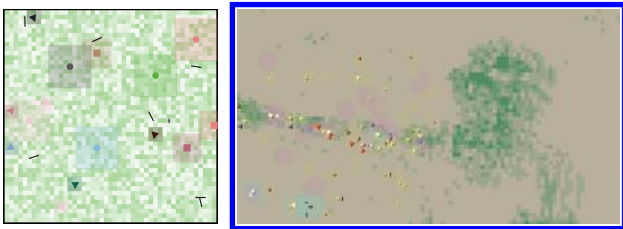


Figure 1: Our A-life environments: in Netlogo and in Unity.

Herbivores vary in shape and colour. They consume grass, shown as green cells. Predator agents consume herbivores and are visualized as short black line segments. A single human avatar is controlled by a player via keyboard. On each tick, the avatar movement along with a description of the 3×3 -cell neighbourhood centred on the avatar are recorded. Specifically, we record 99 numbers which represent the presence of obstacles, amounts of grass, and the numbers of each type of AI-controlled agents in each of the 9 cells (11 numbers for each cell). We then record the avatar’s action (i.e., moving to one of the 8 neighbours or staying put).

A run of N time ticks yields two matrices: the avatar’s inputs ($N \times 99$) and the avatar’s actions ($N \times 1$), which constitute the training data from which we learn a proxy to mimic the avatar’s actions on a previously unseen run of M ticks. Such a run yields test data with $M \times 99$ inputs and $M \times 1$ actions. We then calculate proxy accuracy as the percentage of rows in the test input matrix for which the proxy gives the actual action taken by the human avatar.

As a proxy, we use a memory-free linear model that computes a scalar utility of each of the 9 cells as a dot product of the 11 numbers describing the cell and 11 proxy parameters (i.e., a multichannel convolution with a $11 \times 1 \times 1$ kernel). The proxy agent then moves to the cell with the highest utility. The 11 proxy parameters are found via a simple genetic search with the fitness computed on the training data. Preliminary results suggest a proxy test accuracy of 78 to 91% after training on matrices containing 600 to 4000 rows.

5 Future Work & Conclusions

We are concurrently implementing a similar A-life environment using Unity game engine (Unity Technologies, 2018)

for an interactive art installation *Dyscorpia: Human in the Loop* to run in Edmonton, Alberta, Canada in April 2019: <https://www.dyscorpia.com>. Here, instead of a single keyboard-controlled avatar, multiple people are tracked with a camera as they walk around the installation. Their movements are painted into the A-life environment as grass. The right panel in Figure 1 shows a profile of a single human pointing to the left with his arm. We plan to record human movement data in the day time, use it to construct human proxies in the evening, run evolution of AI agents with the proxies overnight, and release the evolved agents in the installation next morning.

In conclusion, this paper presents an update on our ongoing project (Bulitko et al., 2018). Preliminary results with even rudimentary proxy learning are promising.

6 Acknowledgments

We appreciate support by The Kule Institute for Advanced Research, Social Sciences and Humanities Research Council and Edmonton Arts Council.

References

- Ackley, D. and Littman, M. (1991). Interactions Between Learning and Evolution. *Artificial Life II*, 10:487–509.
- Bulitko, V., Carleton, S., Cormier, D., Sigurdson, D., and Simpson, J. (2017). Towards positively surprising non-player characters in video games. In *Proceedings of the Experimental AI in Games (EXAG) Workshop at the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AI-IDE)*, pages 34–40.
- Bulitko, V., Walters, M., Cselinacz, M., and Brown, M. R. (2018). Evolving NPC behaviours in A-life with player proxies. In *Proceedings of the Experimental AI in Games (EXAG) Workshop at the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.
- Hello Games (2018). No Man’s Sky Next.
- Kim, Y.-H., Moraglio, A., Kattan, A., and Yoon, Y. (2014). Geometric generalisation of surrogate model-based optimisation to combinatorial and program spaces. *Mathematical Problems in Engineering*, 2014.
- Rawal, A. and Miikkulainen, R. (2018). From nodes to networks: Evolving recurrent neural networks. *CoRR*, abs/1803.04439.
- Soares, E. S., Bulitko, V., Doucet, K., Cselinacz, M., Soule, T., Heck, S., and Wright, L. (2018). Learning to recognize A-Life behaviours. In *Poster collection: The Annual Conference on Advances in Cognitive Systems (ACS)*.
- Soule, T., Heck, S., Haynes, T. E., Wood, N., and Robison, B. D. (2017). Darwin’s Demons: Does evolution improve the game? In *Proceedings of European Conference on the Applications of Evolutionary Computation*, pages 435 – 451.
- Unity Technologies (2018). Unity Engine.
- Wilensky, U. (1999). Netlogo.

Stability of Cooperation in Societies of Emotional and Moody Agents

Joe Collenette¹, Katie Atkinson¹, Daan Bloembergen² and Karl Tuyls¹

¹Department of Computer Science, University of Liverpool, UK

²Intelligent and Autonomous Systems, Centrum Wiskunde & Informatica, Amsterdam, NL
j.m.collenette@liverpool.ac.uk

Abstract

It is well documented that cooperation may not be achieved in societies where self-interested agents are engaging in Prisoner's Dilemma scenarios. In this paper we demonstrate, in contrast, that agent societies that use human-inspired emotions within their decision making, can reach stability in cooperation. Our work makes use of the Ortony, Clore, and Collins (OCC) model of emotions and we analyse the evolutionary stability of two different implementations that make use of key emotions from this model. Firstly, we consider an agent society that solely make use of this model of emotions for the agents' decision making. Secondly we look at a model that extends the emotional agents with a model for representing mood. We set out a proof that shows that our emotional agents are an evolutionarily stable strategy when playing against a worst-case scenario strategy. The proof demonstrates that our established model of emotional agents enables evolutionary stability to be achieved, without modification to this model. In contrast, the model of moody agents was shown not to be an evolutionarily stable strategy. Our analysis sheds light on the nature of cooperation within agent societies and the useful role that simulated emotions can play in the agents' decision making and the society as a whole.

Introduction

Models of emotion, particularly those based on the Ortony, Clore, and Collins (OCC) account of emotions, have been used as part of agents' decision making processes to explore their effects on cooperation within social dilemmas (Ortony et al., 1990; Lloyd-Kelly et al., 2014; Collenette et al., 2017). These studies include exploration of the model against a number of different agents within multiple environments. Knowing how these agents, using the OCC model of emotions, react in a simulated environment is an important study in understanding how they will behave against a specific subset of characteristic agents and environments. This paper expands our knowledge of agents using the OCC model of emotions in broader terms, by exploring the evolutionary stability of these agents in a Prisoner's Dilemma setting. The knowledge of whether these strategies can be considered an evolutionarily stable strategy (ESS) allows us to state that these human-inspired agents are able to flourish as a society against invading strategies.

We analyse two different interpretations of OCC agents. Firstly, Emotional agents that decide their action using only the model of emotions, based on a subset of the OCC model. To analyse the evolutionary stability of these agents we use the Prisoner's Dilemma game as this allows us to effectively look at the cooperation these societies achieve and whether the cooperation is sustainable against invading strategies.

We contrast the results with the second interpretation of an OCC agent, the Moody agent (Collenette et al., 2017), which uses the OCC model of emotions for decision making in addition to a psychology-grounded model of mood. Our analysis highlights the different strategies that are needed to achieve success as a society in terms of both stability and cooperation, in the iterated Prisoner's Dilemma.

We find that Emotional agents can be considered an ESS in the Prisoner's Dilemma when given time to converge against the opponent. This is in contrast to Moody agents, which are not an ESS. The Emotional agents take a more defensive strategy that allows cooperation to remain stable over time when playing against other Emotional agents. Opponents are able to take advantage of the Moody agents as they try to create cooperation. Actively trying to create cooperation with opposing strategies makes the Moody strategy more unstable when compared to the Emotional agents.

Analysing two different human-inspired agent strategies, has allowed us to show the inherent risk that cooperation brings in the Prisoner's Dilemma. The work has also showed how different strategies are better deployed in different scenarios. Emotional agents are better suited to a mixed group of agents with differing strategies than the Moody agents, while Moody agents are better suited than Emotional agents when only one strategy exists.

Background

The Prisoner's Dilemma is a social dilemma, popularised through the influential Axelrod's tournament (Axelrod and Hamilton, 1981), where two players pick between cooperating with the other player, or trying to take advantage. This choice is made simultaneously, independent of one another, and with no prior communications. If both players choose

C, C	D, D	D, C	C, D
R, R	P, P	T, S	S, T
3, 3	1, 1	5, 0	0, 5

Table 1: Prisoner’s Dilemma payoff matrix with example payoffs. *Cooperate, Defect.*

to cooperate this is the best joint payoff. If one player defects then the defector receives the best individual payoff, and the cooperator receives the worst individual payoff. If both agents choose to defect they both receive the joint worst payoff. The payoffs for the iterated Prisoner’s Dilemma (Table 1) must have the following restrictions hold (Rapoport et al., 1965) to be valid:

- $(T)emptation > (R)eward > (P)unishment > (S)ucker$
- $R > (S + T)/2$

A strategy can be described as an ESS when the majority of the agents in a society are using this strategy, and it cannot be invaded by any other strategy that is initially rare (Smith and Price, 1973); the definition is given below.

Definition 1. Let $V(A, B)$ be the expected payoff strategy A receives when playing against strategy B . A strategy M is considered an ESS where M is the dominant strategy and the following holds for all invasion strategies I where $M \neq I$.

$$V(M, M) > V(I, M) \text{ OR} \\ (V(M, M) = V(I, M) \text{ AND } V(M, I) > V(I, I)) \quad (1)$$

Evolutionary stability in the Prisoner’s Dilemma has been extensively analysed (Thomas, 1985; Bloembergen et al., 2015), with no pure strategy being an ESS in the iterated version of the Prisoner’s Dilemma (Boyd and Lorberbaum, 1987). Furthermore no TIT-FOR- n -TATS is an ESS (Farrell and Ware, 1989), nor are any reactive strategies (Lorberbaum, 1994). The predictability of these kinds of strategies allows invasion strategies to be successful. This shows that evolutionary stability is an extremely demanding criterion to place on a strategy in the iterated Prisoner’s Dilemma. Our Emotional and Moody agents differ from previous analyses of strategies in this problem set, as they are able to identify individual opponents and change their actions based on the individual, and also the memory space of these strategies extends beyond a single interaction.

Emotional agents

We analyse theoretically the evolutionary stability of Emotional agents described by Lloyd-Kelly (2014), whereas Lloyd-Kelly (2014); Collenette et al. (2017) focus primarily on experimental studies only. These emotional agents simulate a subset of the OCC model of emotions (Ortony et al., 1990) from the psychology literature.

The authors of the OCC model show how emotions influence a change in behaviour, with peoples’ actions being a result of their current emotional makeup. A person’s current emotional makeup is then influenced by interactions with other people in the environment. The OCC model defines 22 emotions, which are organised into a hierarchical structure with definitions on how each emotion responds to different actions (Ortony et al., 1990). The OCC model does not define how emotions are processed internally, but gives the outward effects of these emotions. Various agent designers have used this model successfully as part of their agents’ decision making process (André et al., 2000; Popescu et al., 2014; Lloyd-Kelly et al., 2014; Collenette et al., 2017).

Our focus is on two key emotions from the OCC model, *Gratitude* and *Anger*, within agents that interact in the Prisoner’s Dilemmas. The use of these two emotions affects the decision making process of the agents and changes whether an agent is currently cooperating or defecting. Work has also been conducted where additional emotions have been implemented, with the focus being on other areas of agent interactions such as agent replication (Lloyd-Kelly et al., 2014; Lloyd-Kelly, 2014). We are focusing on the decision making strategy of the agents, and whether this can be considered an ESS. This leads us to include only the emotions that affect the outcome of a decision, namely *Gratitude* and *Anger*, this is the same as in Lloyd-Kelly et al. (2014); Collenette et al. (2017).

For clarity we repeat how the two emotions, *Gratitude* and *Anger*, are implemented by Lloyd-Kelly (2014). Each emotion has a value and a threshold, both are represented by an integer. An interaction can increase an emotion’s value, up to the emotional threshold. When the threshold is reached, the agent’s action will change to reflect the emotional trigger, then the value of the emotion is reset. An agent’s threshold is determined by its “personality”. If an Emotional agent receives a cooperative move, the Gratitude value will increase. When the Gratitude threshold is reached the agent will then cooperate with that opponent. Similarly with *Anger*, if an opponent defects against the Emotional agent, the *Anger* value will increase. Once the *Anger* threshold is reached the Emotional agent will now defect against the opponent.

The possible personalities that the previous work has analysed are shown in Table 2. While the thresholds can be defined as much larger, we restrict them to this small set as this best reflects how emotions are short-term biological processes (Keltner and Gross, 1999).

Moody Agents

The Moody agents use simulated model of mood, in addition to the same subset of simulated emotions the Emotional agents use. This mood model informs decision making and is grounded in psychology (Collenette et al., 2016). The mood model uses mood as a real number, lower val-

Anger Threshold	Gratitude Threshold	Character
1	1	Responsive
1	2	Active
1	3	Distrustful
2	1	Accepting
2	2	Impartial
2	3	Non-Accepting
3	1	Trustful
3	2	Passive
3	3	Stubborn

Table 2: Personalities for agents using the OCC model of emotions, with their Anger and Gratitude thresholds

ues represent more depressed moods and higher values represent more positive moods, reflecting how psychologists have represented human mood (Diener et al., 1985; Hepburn and Eysenck, 1989; Hertel, 1999; Bilderbeck et al., 2016). Agents using the mood model use a real number between 0 and 100 to represent mood (Collenette et al., 2016), as this gives an intuitive understanding of whether the current mood is positive or negative. This value can be integrated more accessibly with the other parts of the model.

The model uses the Homo Egelis concept of fairness (Gintis, 2000) to control how the mood is affected after any given interaction with other agents. Fairness is an important concept when considering whether a given outcome can be considered positive or negative, as fairness has been shown to affect decision making in human societies (Fehr and Schmidt, 1999). Mood has an effect on what is considered fair by affecting perception and judgement (Mayer and Hanson, 1995; Forgas and Bower, 1987). When modelling multi-agent systems, we can capture a notion of human fairness (de Jong et al., 2008) and using this human-inspired model can be beneficial, as de Jong and Tuyls (2011) describes.

Definition 2 shows how the mood value is calculated after an interaction with another agent (Collenette et al., 2016). The value of the mood is affected by the payoff the agent has received and how “fair” the Moody agent believes this payoff to be in the context of both agents’ previous payoffs. Evaluating fairness is achieved using the Homo Egelis equation (Fehr and Schmidt, 1999). In Equation 2 the first line retrieves the value of α that will be used in the Homo Egelis equation. By definition this needs to be a value greater than 0 and less than 1, where lower values put less emphasis on the difference between the two agents. $\alpha = \beta$ represents an idealistic scenario where agents care equally about inequity between the opponents and themselves. Higher mood values should give a lower α , as being in a low mood represents that the agent “thinks” it is doing badly in the environment and as such will care more about inequity by design of the mood model (Collenette et al., 2016). This reflects how people also care more about in-

equity when doing poorly (Fehr and Schmidt, 1999). The value of the mood will lie between 0 and 100 and we “flip” the number and divide by 100, for example a mood of 75 will give an α of 0.25.

Definition 2 (Mood Calculation(Collenette et al., 2016)). *Let AG be the set of all agents, with i and $j \in AG$. Let p_i return the payoff of agent i . Let m_i return the mood of agent i , in the range $0 < m < 100$. Let μ_i denote the average payoff for agent i . Let j be the opponent of agent i . Let $\alpha = \beta$.*

$$\begin{aligned}
\alpha_i &= (100 - m_i)/100 \\
\Omega_i(j) &= \mu_i - \alpha_i \cdot \max(\mu_j - \mu_i, 0) - \\
&\quad \beta_i \cdot \max(\mu_i - \mu_j, 0) \\
m_i &\leftarrow m_i + (p_i - \mu_i) + \Omega_i(j)
\end{aligned} \tag{2}$$

The second line is the Homo Egelis equation. While traditionally the average between the agent and all opponents is taken into account, the model uses only the current opponent so that the agent’s mood is not affected by opponents it never interacts with. The result of Ω is the average payoff the agent has received with a weighted adjustment made based on the difference between the agent’s average payoff and the opponent’s average payoff.

The third and final line shows how the mood will go up or down based on the difference between the payoff the agent received and the agent’s current average payoff, adjusted by the value of the Homo Egelis equation. The value is then restricted to lie between 0 and 100. This value places the mood into one of five mood levels (Very High, High, Neutral, Low, Very Low).

The mood value will then affect the action selection of the agent, reflecting the psychology literature regarding low moods (Haley and Strickland, 1986; Hertel et al., 2000). For generally low moods, the agent will defect against any new agents. When the mood is high the agent cooperates with new agents, and when the mood is very high, the agent will always cooperate. When the mood is very low, the agent will always defect. The full description of the model along with the psychology grounding is given in Collenette et al. (Collenette et al., 2016, 2017). Table 3 also outlines the changes that Mood causes.

Evolutionary Stability Analysis

To analyse whether Emotional and Moody agents can be considered an ESS, we need to design an opponent strategy that will take the largest advantage of these agents and minimise their payoff. By designing such a strategy we can show that if Emotional and Moody agents are able to remain the dominant strategy, then no other strategy can invade the Emotional or Moody agents.

We will use a strategy termed the *Oracle*. The effectiveness of the strategy is achieved by breaking an assumption of

Mood Level	Moody Agent Cooperating	Moody Agent Defecting
Very High ($m > 90$)	No change	Cooperate
High ($70 > m \geq 90$)	No change	Cooperate against new opponent
Neutral ($30 \geq m \leq 70$)	No change	No change
Low ($10 \leq m < 30$)	Defect against new opponent	No change
Very Low ($m < 10$)	Defect	No change

Table 3: How simulated mood changes the action selection in Moody agents

the Prisoner’s Dilemma, namely that players have no knowledge of the opponent’s move, as reflected by the name. Intuitively the Oracle strategy will always cooperate with itself, and when faced with another strategy will choose the worst outcome for the opponent, effectively making it the worst case scenario for the opponent. The Oracle strategy targets the conditions needed to be an ESS, allowing effective analysis of evolutionary stability.

For example, if an opponent chooses to cooperate, the Oracle strategy is guaranteed to defect, giving the Oracle strategy the T payoff and the opponent the S payoff. For a society of agents to successfully survive an Oracle invasion, that society must have perfect cooperation among themselves, and protect themselves from the Oracle by always defecting against the opposing strategy.

We can now state that the expected value V after one round, for the Oracle strategy o against strategy b , where $Ac(b, o)$ returns the action b would use against o , can be calculated as:

$$V(o, b) = \begin{cases} R & \text{IF } b \equiv o \\ T & \text{IF } b \not\equiv o \text{ AND } Ac(b, o) = C \\ P & \text{IF } b \not\equiv o \text{ AND } Ac(b, o) = D \end{cases}$$

The Oracle is the most effective strategy at minimising the payoff of the Emotional agents, which we show in Theorem 4. To prove this theorem we need to use the fact that Emotional agents will not change their action if their opponent uses the same action (Lemma 3).

Lemma 3. *An Emotional agent will not change its subsequent action against an opponent if its opponent’s action mirrors the Emotional agent’s action.*

Intuitively this means that if an Emotional agent is cooperating and its opponent is also cooperating then the Emotional agent will not switch to defection and visa-versa.

Theorem 4. *The expected payoff of Emotional agents using the defined personalities, in the Prisoner’s Dilemma with the payoffs defined in Table 1, is minimised by the Oracle strategy, with no other strategy being able to lower the expected payoff further.*

Proof. If the Emotional agent is initially defecting then the payoff achieved by the Emotional agent is $V(e, o) = Pn$ where n is the number of rounds. Neither the Oracle nor the Emotional agent will ever change their action, as per Lemma 3.

When the Emotional agent is initially cooperating then the payoff the Emotional agent receives is S as the Oracle defects. By the definition of the Emotional agent we know that the Emotional agent will change to defection when the Anger level of that agent reaches the Anger threshold. The Emotional agent will change its action to defecting against the Oracle agent. Once the Emotional agent has changed its action the Oracle will continue to defect, now both agents are defecting. As both agents are defecting they will continue in mutual defection indefinitely as per Lemma 3, and the Emotional agent will receive the P payoff. We can now state that the expected value of an initially cooperative Emotional agent against the Oracle is $V(e, o) = Sm + P(n - m)$ where m is the Anger threshold and n is the number of rounds.

Assume there is a strategy x where the payoff achieved by the Emotional agent is $V(e, x) < Sm + P(n - m)$ when initially cooperating and $V(e, x) < Pn$ when the Emotional agent is initially defecting.

If the strategy x only defects then the payoff of an initially defecting Emotional agent is $V(e, x) = Pn$, and for an initially cooperative agent $V(e, x) = Sm + P(n - m)$. This contradicts the assumption as $V(e, x) < Sm + P(n - m)$, therefore only defecting is not the strategy x .

If the strategy x only cooperates then the expected payoff of an initially defecting agent is $V(e, x) = Tg + R(n - g)$ where g is the Gratitude threshold. The payoff for the initially cooperative agent is $V(e, x) = Rn$. This leads to a contradiction as T and R are both larger than S and P in the Prisoner’s Dilemma. Strategy x therefore cannot only cooperate.

Strategy x must therefore be a mixed strategy. By Lemma 3 we know that repeating the Emotional agent action leads to an indefinite repetition. Therefore we consider the strategy of doing the opposite of the Emotional agent. When the Emotional agent is cooperating the strategy x will defect and when the Emotional agent is defecting the strategy x will cooperate. Therefore the expected value of the Emotional agent is $V(e, x) = (Tg + Sm) \frac{n}{g+m}$.

This is the minimal strategy since if the strategy x switches a cooperative move for a defect, then the Emotional agent will receive a P payoff, but will not switch to cooperation, effectively removing the S payoffs it would have received. By definition of the Prisoner’s Dilemma $P > S$ so the Emotional agent’s expected value will increase. If the strategy x switches a defection for a cooperative move, then the Emotional agent will receive a R rather than a S payoff, and will not switch to defection leading to further R payoffs. By definition $R > S$ so the Emotional agent’s

expected value will increase.

As strategy x must be the mixed strategy of doing the opposite action of the Emotional agent, therefore the following must hold:

$$V(e, x) < V(e, o)$$

We evaluate this equation where $V(e, x) = (Tg + Sm) \frac{n}{g+m}$ and $V(e, o) = Sm + P(n - m)$ with an initially cooperating Emotional agent. We choose the initially cooperating agent as $Sm + P(n - m) < Pn$.

Given that we are looking at the possible personalities in Table 2 and using the values for the Prisoner's Dilemma in Table 1, we use the personalities that will play the maximum number of cooperative moves and the lowest amount of defection moves to minimize the number of T payoffs and maximize the number of S payoffs. This is the personality Trustful. Plugging in the values in the above equation results in the following:

$$\begin{aligned} (5 \cdot 1 + 0 \cdot 3) \frac{n}{4} &< 0 \cdot 3 + 1(n - 3) \\ \frac{5}{4}n &< n - 3 \\ \frac{1}{4}n &< -3 \\ n &< -12 \end{aligned}$$

For the equation to hold, strategy x must yield a lower expected payoff to the emotional agent than the Oracle, for any number of round n . We have reached a contradiction as n must be positive by definition.

Therefore strategy x is not the mixed strategy. We have also shown that only cooperating and only defecting are not strategy x .

\therefore The expected payoff of Emotional agents using the defined personalities, in the Prisoner's Dilemma with the payoffs defined in Table 1 is minimised by the Oracle strategy, with no other strategy being able to lower the expected payoff further. \square

The Oracle agent is the most effective agent at minimizing the expected payoff of the Emotional agents, for the given personalities and the given values for the Prisoner's Dilemma. By restricting the analysis to the given agent personalities and values, we are able to analyse the Emotional agents more effectively as we only need to look at one opposing strategy rather than both an Oracle strategy and the most effective mixed strategy.

Emotional Agents

We now move on to show that the Emotional agents are not an ESS when there are no restrictions on reproduction and interaction speed. We will show this both for the initially cooperative Emotional agent and the initially defecting Emotional agent. The values the strategies will receive for both

the initially cooperative Emotional agent, initially defecting Emotional agent, and the Oracle agent are given in Table 1, for an initial interaction.

Theorem 5. *Emotional agents are not an ESS in the initial phase.*

Proof. Assume that Emotional agents are an ESS. Given a majority of Emotional agents, with an invasion force of Oracle agents, by definition of an ESS, Equation 1 must hold for the Emotional agent strategy M and the invading Oracle strategy I . For the initially cooperative Emotional agent we have,

$$R > T \text{ OR } (R = T \text{ AND } S > R)$$

and for the initially defecting Emotional agent

$$P > P \text{ OR } (P = P \text{ AND } P > R).$$

A contradiction has been reached for each line in both the initially cooperative Emotional agent and the initially defecting Emotional agent. $P > P$ is a contradiction, $T > R$ and $R > P$ (from Table 1) contradict the equations.

\therefore Emotional agents are not an ESS in the initial phase. \square

Emotional agents are not an ESS, due to how the agents respond initially to the Oracle strategy. Emotional agents are able to respond to the opponent on an individual agent level, that is, the action the Emotional agent gives depends on who the opponent is. Collenette et al. (2016) has shown that all initially cooperative Emotional agents will cooperate with each other indefinitely.

Lemma 6. *All Emotional agents will converge to defection with all Oracle agents given a sufficiently high number of interactions and sufficiently high randomness in pairing.*

Emotional agents will eventually choose to defect indefinitely against the Oracle strategy given enough time to adjust. As other agents can not affect the action choice of either the Emotional agent or the Oracle, the Oracle will only defect which only increases the agent's Anger value. The Emotional agent will not switch back to cooperation.

Theorem 7. *Initially cooperative Emotional agents that have fast interactions and slow reproduction are an ESS*

Proof. Assume a fraction ϵ of the population is replaced by the invading Oracle strategy. We also assume that interactions between all agents are fast and reproduction of the population is slow. Given the fast interactions, and slow reproduction with respect to time, all Emotional agents will be defecting against any other Oracle agents that may be residual in the populous, as per Lemma 6. This gives both the Oracle strategy and the Emotional agents the P payoff. No Emotional agent has adjusted to the newly invading ϵ -Oracle agents, and as such are able to receive the S payoff.

Thus the expected payoff of the Emotional agents against the Oracle agents is $V(e, o) = S\epsilon + P(1 - \epsilon)$, and the expected payoff for the Oracle agents against the Emotional agents will therefore be $V(o, e) = T\epsilon + P(1 - \epsilon)$.

Using these values in Equation 1 gives us the following:

$$R > T\epsilon + P(1 - \epsilon) \text{ OR}$$

$$(R = T\epsilon + P(1 - \epsilon) \text{ AND } S\epsilon + P(1 - \epsilon) > R)$$

The equation will therefore hold, given that ϵ is sufficiently small as per the definition of an ESS strategy (Eshel, 1983). The expected value that the Oracle agent gets from the Emotional agents will be sufficiently close to P such that the first line will always hold. The Emotional agents are protecting themselves from defection of the Oracle agents. The ϵ number of new Oracle agents are unable to take a large enough advantage of the Emotional agents that they can break the stability.

\therefore Initially cooperative Emotional agents are an ESS, when interactions are fast and reproduction is slow. \square

In summary, initially cooperative Emotional agents are an ESS, as no strategy is able to minimise the payoff of the Emotional agents more than the Oracle agent. The assumptions of fast interactions and slow reproduction, are to allow the Emotional agents to adapt to all the Oracle agents before the next reproduction. A sufficiently small epsilon in this case is less than half, if using Table 1 as the payoff matrix, given the assumption that the Emotional agents are the majority as per the definition of an ESS (Eshel, 1983). The assumptions of fast interaction and slow reproduction are part of an efficient evolution and learning process (Hinton and Nowlan, 1987), with fast interactions allowing the agent to learn which in turn guides the reproduction process.

When we consider the initially defecting Emotional agent, they have already adapted to the invading Oracle agents. However the initially defecting Emotional agent being able to protect its payoff is not enough for it to be considered an ESS. The initial defection will prevent the Emotional agents from cooperating as a group, and with no avenue of breaking the defection, this allows the Oracle agents which do work together, to be a fitter strategy.

Moody Agents

We will now be comparing the Emotional agents to Moody agents, again using the Oracle strategy. The Moody agents we are analysing are similar to Emotional agents as they both use OCC-inspired emotions as part of their decision-making process. However the addition of the Mood model on top of the emotions changes how the Moody agents react in certain circumstances. The Moody agents have been shown to perform better in self-play than the Emotional agents (Collette et al., 2017). Analysing the evolutionary stability of these agents shows us the effects the Mood model has, when compared to Emotional agents.

To analyse the Moody agents, we need to take into account that different mood levels affect how the Moody agents respond to the Oracle. Therefore we need to analyse each mood level individually to be able to gain insights into the Mood model as a whole. Table 3 shows each Mood level and when the Mood value will override the action selection of the agent. As we know that Emotional agents need to be initially cooperative to be considered an ESS, we will assume that the Moody agents are also initially cooperative. We will start the proofs from very high levels of mood down to very low moods. For simplicity in the analysis we also assume that the mood levels do not change.

Theorem 8. *Moody agents that are in an initially very high mood are not an ESS*

Proof. Assume Moody agents in a very high mood are an ESS. Given that a fraction ϵ of the population is replaced by the invading Oracle strategy, the expected payoff of two Moody agents in a very high mood is R since by definition all Moody agents are cooperating. Thus the expected payoff of an Oracle agent against the Moody agent will be T and the Moody agent will receive S . Equation 1 holds by definition of an ESS. Therefore the following equation is true:

$$R > T \text{ OR } (R = T \text{ AND } S > R)$$

This is a contradiction, both sides of the *OR* are false. By definition of the Prisoner's Dilemma, $T > R$ which contradicts both sides of the equation.

\therefore Moody agents in an initially very high mood are not an ESS. \square

Moody agents in very high moods are not an ESS. This is due to these particular agents being functionally equivalent to a fully cooperative strategy, which is known to not be an ESS (Boyd and Lorberbaum, 1987; Lorberbaum, 1994). Moving onto high moods, we will show that initially cooperative agents are equivalent to Moody agents in a neutral mood. This allows us to avoid repeating proofs.

Lemma 9. *Moody agents that are in an initially high mood are functionally equivalent to Moody agents in a neutral mood.*

In high moods we know that the only effect on decision making is that the Moody agent will always cooperate with a unknown opponent. By our assumption above, Moody agents are initially cooperative, and therefore Moody agents in a high mood are functionally identical to Moody agents in a neutral mood.

We will show that neutral moods are functionally equivalent to Emotional agents. This gives us that Theorem 7 holds for Emotional agents and Moody agents that are in either a neutral mood or a high mood.

Lemma 10. *Moody agents that are in an initially neutral mood are functionally equivalent to Emotional agents.*

The mood value has no effect on action selection; by definition of Moody agents, they will respond using the emotion model as defined for the Emotional agent. We now go on to show that initially very low, and low, mood levels are not an ESS. We do this as both types of Moody agents are functionally equivalent against an Oracle agent.

Lemma 11. *Moody agents that are in an initially low mood are functionally equivalent to Moody agents against an Oracle agent.*

The Moody agents will defect with all other Moody agents as the initial cooperation is broken. The low moods change the first action to defection, the Moody agents will continue to defect indefinitely as per Lemma 3. The Moody agents in a very low mood will defect by definition. When playing against an Oracle both the Oracle and the Moody agent in a low mood will play defection. The Oracle will also defect against a Moody agent in a very low mood. Regardless of whether the Moody agent is in a low mood or a very low mood, they will defect indefinitely with both other Moody agents and Oracle agents.

Theorem 12. *Moody agents that are in an initially low mood or are in an initially very low mood are not an ESS*

Proof. Assume Moody agents in low mood or a very low mood are an ESS. Given that a fraction ϵ of the population is replaced by the invading Oracle strategy, the expected payoff of two Moody agents is P . This is valid for both very low and low moods as per Lemma 11.

The expected payoff of an Oracle agent against a defecting Moody agent is P , and they will be in mutual defection indefinitely. Equation 1 holds by definition of an ESS. Therefore the following equation is true:

$$P > P \text{ OR } (P = P \text{ AND } P > R)$$

We have reached two contradictions, $P > P$ and $P > R$ since by definition of the Prisoner's Dilemma $R > P$.

\therefore Moody agents that are in an initially low mood or are in an initially very low mood are not an ESS \square

To conclude that Moody agents overall are not an ESS, we need to now show that the Moody agents' mood level will always lead to the evolutionarily unstable mood levels. As only neutral and high moods are possibly an ESS, we only consider these two mood levels.

Theorem 13. *Moody agents in an initially neutral or initially high mood will move to the very high mood level, when there is a sufficiently small ϵ invasion of Oracle agents.*

Proof. The expected payoff of a Moody agent(k) in either a neutral or high mood will be $V(k, o) = S\epsilon + P(1-\epsilon)$ against an Oracle agent and $V(k, k) = R$, as Theorem 7 applies as per Lemma 10, since Moody agents are functionally equivalent to Emotional agents. When a Moody agent receives a payoff its mood level updates. We state that $\Omega_i(j) \approx \mu_i$

when two Moody agents interact. The majority of interactions are between two Moody agents, therefore making their averages (μ_i and μ_j) approximately equal, which means Ω is not changing the perception of the reward in Equation 2.

The final calculation updates the mood $m_i \leftarrow m_i + (p_i - \mu_i) + \Omega_i(j)$. With the majority of the interactions being between two Moody agents we can state that $p_i - \mu_i \approx 0$. We know that $\Omega > 0$, therefore the m_i will increase in the majority of cases indefinitely as the invasion of Oracle agents is ϵ small.

\therefore Moody agents in an initially neutral or initially high mood will move to the very high mood level, given a sufficiently small ϵ invasion of Oracle agents. \square

In conclusion Moody agents are not an ESS. While Moody agents may be an ESS in neutral and high moods, with the same conditions as the Emotional agents, the Moody agents will move into the other mood levels that are not an ESS. If the mood level of Moody agents was to stay stable over time, this would go against the design principles of the model (Collenette et al., 2016). The psychological grounding of the moody model requires that mood levels change over time as per the psychology literature (Mayer and Hanson, 1995; Diener et al., 1985).

Conclusion

We have shown that Emotional agents that use a model of emotions as part of their decision-making can be considered an ESS when they initially cooperate with new partners and are able to adapt to an invading strategy before reproducing. We showed that Moody agents using a simulated model of mood alongside the model of emotions as their decision-making process, are not part of an ESS. This is because some mood levels break the assumption that Moody agents cooperate together and will always protect themselves from invading strategies. We tested these human-inspired agents against an Oracle strategy, which we showed was the most effective at minimising the expected payoff of the Emotional agents and can successfully invade the Moody agents.

Collenette et al. (2017) showed that Emotional agents had stable levels of cooperation, which is reflected by these agents being an ESS. Collenette et al. (2017) also showed that Moody agents increased their cooperation over time, with a higher average payoff than the Emotional agents. The ability to increase their cooperation is what causes the mood model strategy not to be an ESS. This shows us how more protective strategies are able to succeed against invasive strategies, whilst when playing against more reciprocating strategies, a more responsive and risky strategy succeeds. In turn, this reflects the inherent risks and rewards in the Prisoner's Dilemma.

Our work poses further questions, namely whether we can reduce the assumptions needed for the Emotional agents. We will be aiming to find out whether the mood model can

be adapted to become an ESS, by changing the secondary decision making process, which is currently the OCC-based model and is not an inherent part to the mood model. Further alterations to the fairness metric, such as changes to the α and β variables may affect the evolutionary stability, which provides a further avenue of investigation.

This body of work is part of the wider literature that considers simulated emotions, mood, and personalities in agent systems, in terms of how they model these human-inspired concepts and the effects these aspects have on wider agent societies. The majority of the literature concerning these kinds of agents focuses on simulations and observing the effects. We have taken a broader view of human-inspired agents by analysing evolutionary stability in an account that implements both Emotional and Moody agents.

References

- André, E., Klesen, M., Gebhard, P., Allen, S., and Rist, T. (2000). Integrating models of personality and emotions into lifelike characters. In Paiva, A., editor, *Affective Interactions*, volume 1814 of *LNCIS*, pages 150–165. Springer.
- Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.
- Bilderbeck, A., Reed, Z., McMahon, H., Atkinson, L., Price, J., Geddes, J., Goodwin, G., and Harmer, C. (2016). Associations between mood instability and emotional processing in a large cohort of bipolar patients. *Psychological Medicine*, pages 1–10.
- Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: a survey. *Journal of Artificial Intelligence Research*, 53:659–697.
- Boyd, R. and Lorberbaum, J. P. (1987). No pure strategy is evolutionarily stable in the repeated prisoner’s dilemma game. *Nature*, 327:58 EP –.
- Collenette, J., Atkinson, K., Bloembergen, D., and Tuyls, K. (2016). Modelling mood in co-operative emotional agents. In *Proceedings of Distributed Autonomous Robotic Systems*.
- Collenette, J., Atkinson, K., Bloembergen, D., and Tuyls, K. (2017). Environmental effects on simulated emotional and moody agents. *The Knowledge Engineering Review*, 32.
- de Jong, S. and Tuyls, K. (2011). Human-inspired computational fairness. *Autonomous Agents and Multi-Agent Systems*, 22(1):103–126.
- de Jong, S., Tuyls, K., and Verbeeck, K. (2008). Fairness in multi-agent systems. *The Knowledge Engineering Review*, 23(02):153–180.
- Diener, E., Larsen, R. J., Levine, S., and Emmons, R. A. (1985). Intensity and frequency: dimensions underlying positive and negative affect. *Journal of personality and social psychology*, 48(5):1253.
- Eshel, I. (1983). Evolutionary and continuous stability. *Journal of theoretical Biology*, 103(1):99–111.
- Farrell, J. and Ware, R. (1989). Evolutionary stability in the repeated prisoner’s dilemma. *Theoretical Population Biology*, 36(2):161 – 166.
- Fehr, E. and Schmidt, K. M. (1999). A theory of fairness, competition, and cooperation. *Quarterly journal of Economics*, 114:817–868.
- Forgas, J. P. and Bower, G. H. (1987). Mood effects on person-perception judgments. *Journal of personality and social psychology*, 53(1):53.
- Gintis, H. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton university press.
- Haley, W. E. and Strickland, B. R. (1986). Interpersonal betrayal and cooperation: Effects on self-evaluation in depression. *Journal of Personality and Social Psychology*, 50(2):386.
- Hepburn, L. and Eysenck, M. W. (1989). Personality, average mood and mood variability. *Personality and Individual Differences*, 10(9):975–983.
- Hertel, G. (1999). Mood effects in social dilemmas: What we know so far. *Resolving social dilemmas: Dynamic, structural, and intergroup aspects*, pages 227–243.
- Hertel, G., Neuhof, J., Theuer, T., and Kerr, N. L. (2000). Mood effects on cooperation in small groups: Does positive mood simply lead to more cooperation? *Cognition & emotion*, 14(4):441–472.
- Hinton, G. and Nowlan, S. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Keltner, D. and Gross, J. J. (1999). Functional accounts of emotions. *Cognition & Emotion*, 13(5):467–480.
- Lloyd-Kelly, M. (2014). *Modelling Emotions and Simulating their Effects on Social Interactions in Agent Systems*. PhD thesis, University of Liverpool.
- Lloyd-Kelly, M., Atkinson, K., and Bench-Capon, T. (2014). Fostering co-operative behaviour through social intervention. In *Proc of SIMULTECH’14*, pages 578–585. IEEE.
- Lorberbaum, J. (1994). No strategy is evolutionarily stable in the repeated prisoner’s dilemma. *Journal of Theoretical Biology*, 168(2):117 – 130.
- Mayer, J. D. and Hanson, E. (1995). Mood-congruent judgment over time. *Personality and Social Psychology Bulletin*, 21:237–237.
- Ortony, A., Clore, G. L., and Collins, A. (1990). *The cognitive structure of emotions*. Cambridge university press.
- Popescu, A., Broekens, J., and van Someren, M. (2014). Gamygdala: An emotion engine for games. *IEEE Transactions on Affective Computing*, 5(1):32–44.
- Rapoport, A., Chammah, A. M., and Orwant, C. J. (1965). *Prisoner’s dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press.
- Smith, J. M. and Price, G. R. (1973). The logic of animal conflict. *Nature*, 246(5427):15.
- Thomas, B. (1985). On evolutionarily stable sets. *Journal of Mathematical Biology*, 22(1):105–115.

Effects of Visual Sensory Range on the Emergence of Cognition in Early Terrestrial Vertebrates: An Agent-Based Modeling Approach

Can Gurkan¹, Leif Rasmussen¹ and Uri Wilensky^{1,2,3}

¹Department of Computer Science, ²Department of Learning Sciences, ³Northwestern Institute on Complex Systems
Northwestern University, Evanston, IL 60208, USA
gurkan@u.northwestern.edu

Abstract

As water dwelling vertebrates began to progressively evolve features that enabled them to survive on land, they also developed larger eyes, which would have considerably increased their range of vision above water. This increase in visual range may have facilitated their exploitation of new food sources on land and promoted increased cognitive capacity in the form of planning (MacIver et al., 2017). In this study, we use a multi-level agent-based model to attempt to replicate the dynamics of the hypothetical evolutionary scenario described above. To do so, we use a novel method called agent-centric Monte Carlo cognition (ACMCC) (Head and Wilensky, 2018), which allows us to represent the agents' cognition in a quantifiable manner by performing micro-simulations in a separate agent-based model. In our simulations, we observe that as a population that is adapted to live on land emerges, their mean eye size and cognitive capacity increase.

Introduction

When vertebrates first began to move onto land, they experienced morphological changes, progressively trading their fins for limbs, and their gills for lungs. This better adapted them to their new terrestrial environment (Long and Gordon, 2004). As they emerged from the water, it is likely that they gained much more visual information about their environments. This information gain may have driven the emergence of more advanced cognition and complex planning abilities (Kohashi and Oda, 2008; MacIver, 2009; MacIver et al., 2017; Mugan and MacIver, 2018).

In this study, inspired by MacIver et al. (2017), we hypothesized that terrestrial tetrapods would develop higher planning capacities than their aquatic counterparts due to an increase in their visual perception space.

Methods

We use a multi-level agent-based modeling (ABM) approach in this study. ABMs have been a powerful tool for simulating complex systems (Wilensky and Rand, 2015). We use a novel method called agent-centric Monte Carlo cognition (ACMCC) (Head and Wilensky, 2018) in order to model animal cognition. Each agent has a mental representation of

its environment which is modeled using a separate cognitive ABM. Agents use this cognitive ABM to run "micro-simulations" in order to predict the outcome of different actions they may choose. The product of the number of actions they simulate and the length of these simulations give us an estimation of their cognitive capabilities. We used NetLogo (Wilensky, 1999) along with the recent LevelSpace extension (Hjorth et al., 2015) to implement our model.

Overview

In our model, half the world is underwater (low visibility), and the other half is on land (high visibility). The tetrapods in our model are initially adapted to water. In time, these aquatic tetrapods evolve and some of them transition to life on land. "Food" in our model represents invertebrates available as prey on both water and land. The tetrapods in our model have the following evolvable parameters:

Eye size: the radius of the agents' fields of vision.¹ For the sake of simplicity, we assume that agents have a field of vision of 360 degrees. The effect of eye size on visual range on land is much greater than that of underwater in accordance with previous studies (MacIver et al., 2017).

Mobility: how well adapted the tetrapod is to life underwater or on land. This characteristic aims to capture the morphology of the tetrapod as it undergoes the fin-to-limb transformation and how this relates to its locomotion (Long and Gordon, 2004).

Energy: the level of nutrition (energy) of the tetrapod. Tetrapods gain energy from consuming food and lose energy as they move, plan actions, or reproduce.

Simulation Number: how many micro simulations the agent performs at each time step. It models the ability to consider a number of different scenarios before deciding on the best course of action.

Simulation Length: the number of steps in each micro simulation that the agent performs at each time step. It models the ability to think multiple steps ahead before deciding on the best action.

¹We note that this is a proxy for how far a tetrapod can see and includes factors such as eye size, eye complexity and development.

Planning

To modeling animals' planning and decision-making skills we allowed the modeled tetrapods to run "micro simulations" in which they could simulate the consequences of a variety of actions and select the action (direction of movement) that resulted in the greatest expected energy gain. A micro simulation is a simplified version of the macro model.² Each tetrapod loses an amount of energy proportional to the product of its simulation number and simulation length parameters, representing the cost of cognition. Furthermore, the size of the world in a micro simulation is limited to the initiating agent's visual range. We can think of this as a Partially Observable Markov Decision Process, which has previously been used to model animal decisions (Head and Wilensky, 2018; Mugan and MacIver, 2018; Miller et al., 2017).

Actions

At each time step, each tetrapod moves, consuming energy. The food remains stationary. The speed of an agent's movement depends upon its mobility value and whether it is on land or underwater. The amount of energy consumed depends on an agent's mobility value, its eye size, and whether it is underwater or on land. If a tetrapod reaches food, it kills the food and gains energy. If the energy value of a tetrapod goes below zero, it dies. A tetrapod may reproduce, dividing its energy between it and its offspring. The offspring inherits its parent's parameters but undergoes mutation at a predetermined rate. At each time step, there is an adjustable probability that a new piece of food will appear.

Discussion

The initially abundant food sources on land become scarce following tetrapods transition to land. Then, as can be seen in Figure 1, the terrestrial tetrapods develop larger eyes and an increased cognitive capacity (simulation length \times simulation number), likely due to the selective pressure to remain competitive in consuming the now-scarce food.³ Even though we chose to model a scenario in which only the predators are able to plan, we expect our results to generalize to a broader context in which prey also gain an advantage from planning by being able to better avoid predators.

Conclusion

The results of our model support the idea that increased visual perception allowed the early tetrapods to look fur-

²We note that we chose a model of cognition in which we could explicitly define and measure intelligence. Our specific model of cognition relies on an explicit central representation of the environment that can be manipulated, however we expect that our results could still be observed with different models of cognition.

³Changes in the environment and further specialization and adaptation of species could explain the variation in eye sizes of contemporary intelligent species without undermining our assumptions.

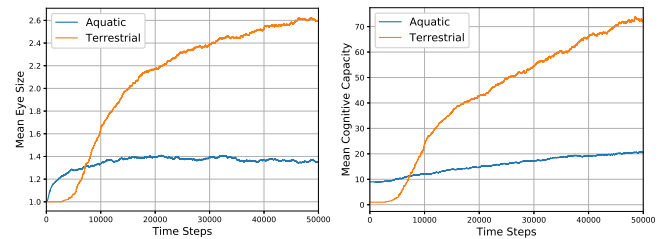


Figure 1: Mean eye size (left) and mean cognition capacity (right) of the terrestrial (mobility > 0) and aquatic (mobility ≤ 0) populations as a function of time.

ther into time as well as space, thus aiding the selection of abilities such as planning, strategic thinking, and complex decision-making. This might be an important factor in explaining the emergence of intelligent life.

References

- Head, B. and Wilensky, U. (2018). *Agent Cognition Through Micro-simulations: Adaptive and Tunable Intelligence with NetLogo LevelSpace: Proceedings of the Ninth International Conference on Complex Systems*, pages 71–81.
- Hjorth, A., Head, B., and Wilensky, U. (2015). LevelSpace NetLogo extension. <http://ccl.northwestern.edu/levelspace/index.html>, Northwestern University, Evanston, IL.
- Kohashi, T. and Oda, Y. (2008). Initiation of mauthner- or non-mauthner-mediated fast escape evoked by different modes of sensory input. *Journal of Neuroscience*, 28(42):10641–10653.
- Long, J. A. and Gordon, M. S. (2004). The greatest step in vertebrate history: A paleobiological review of the fish-tetrapod transition. *Physiological and Biochemical Zoology*, 77(5):700–719. PMID: 15547790.
- MacIver, M. A. (2009). *Neuroethology: From Morphological Computation to Planning*, pages 480–504. Cambridge University Press.
- MacIver, M. A., Schmitz, L., Mugan, U., Murphey, T. D., and Mobley, C. D. (2017). Massive increase in visual range preceded the origin of terrestrial vertebrates. *Proceedings of the National Academy of Sciences*, 114(12):E2375–E2384.
- Miller, K. J., Botvinick, M. M., and Brody, C. D. (2017). Dorsal hippocampus contributes to model-based planning. In *Nature Neuroscience*.
- Mugan, U. and MacIver, M. A. (2018). How sensory ecology affects the utility of planning. In *2018 Conference on Cognitive Computational Neuroscience*.
- Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>, Northwestern University, Evanston, IL.
- Wilensky, U. and Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press.

When is an action caused from within? Quantifying the causal chain leading to actions in simulated agents

Bjørn Erik Juel¹, Renzo Comolatti², Giulio Tononi³ and Larissa Albantakis^{3,*}

¹Department of Molecular Medicine, Brain Signalling Group, University of Oslo, Oslo, Norway

²Institute of Science and Technology, Federal University of Sao Paulo, SP, Brazil

³Department of Psychiatry, Wisconsin Institute for Sleep and Consciousness, University of Wisconsin-Madison, WI, USA

*albantakis@wisc.edu

Abstract

An agent's actions can be influenced by external factors through the inputs it receives from the environment, as well as internal factors, such as memories or intrinsic preferences. The extent to which an agent's actions are "caused from within", as opposed to being externally driven, should depend on its sensor capacity as well as environmental demands for memory and context-dependent behavior. Here, we test this hypothesis using simulated agents ("animats"), equipped with small adaptive Markov Brains (MB) that evolve to solve a perceptual-categorization task under conditions varied with regards to the agents' sensor capacity and task difficulty. Using a novel formalism developed to identify and quantify the actual causes of occurrences ("what caused what?") in complex networks, we evaluate the direct causes of the animats' actions. In addition, we extend this framework to trace the causal chain ("causes of causes") leading to an animat's actions back in time, and compare the obtained spatio-temporal causal history across task conditions. We found that measures quantifying the extent to which an animat's actions are caused by internal factors (as opposed to being driven by the environment through its sensors) varied consistently with defining aspects of the task conditions they evolved to thrive in.

Introduction

By definition, agents are open systems that can dynamically and informationally interact with their environments through sensors and actuators. However, identifying which particular set of events within or outside the agent caused it to act in a certain way is not straightforward, even if its internal structure and dynamics can be assessed in detail. This is demonstrated particularly well by the problem of accountability we currently face with respect to artificial intelligence (Doshi-Velez et al., 2017). While we can, in principle, record all network parameters of a (deep) neural network, such as AlphaGo (Silver et al., 2016), we still lack a principled set of tools to understand *why* the network performed a particular action (Metz, 2016).

Here, we address this issue using artificial agents ("animats") controlled by Markov Brains (MBs) (Hintze et al.,

2017) as a model system of evolved agents, to which we apply a novel formalism for analyzing actual causation (AC) ("what caused what") in complex networks of interacting elements (Albantakis et al., 2019). Although there is no single widely accepted account of (actual) causation (Illari et al., 2011; Halpern, 2016), the AC framework presented by Albantakis et al. (2019) was specifically developed to identify and quantify the strength of the direct causes of any occurrence (subset of network nodes in a particular state at a particular time) within such systems. Notably, this formalism not only considers causes of single-variable occurrences, but also evaluates multivariate causal dependencies.

Given an appropriate model system of behaving agents, the AC framework may serve as a tool for assessing the actual causes of an agent's actions, by characterizing the actual causes of its motor actuators. For this purpose, animats controlled by MBs are particularly suited: MBs are a class of evolvable neural networks that receive sensor inputs and control motor outputs, and can be made small enough to allow for a complete causal and informational analysis, while remaining capable of evolving relatively complex behaviors (Edlund et al., 2011; Albantakis et al., 2014). As MBs may exhibit sparse, recurrent connectivity between their nodes ("neurons"), they resemble biological neural networks more closely than conventional machine-learning systems.

In this study, we demonstrate how the AC analysis can be utilized to evaluate the extent to which an animat's actions are "intrinsic" (caused by internal occurrences) rather than "extrinsic" (caused by sensor inputs, which are driven by the environment). To that end, we evolved animats to solve a perceptual-categorization task (Beer, 2003; Albantakis et al., 2014) under three task conditions with varying demands for memory and context-dependent behavior. As shown by Albantakis et al. (2014), animats evolved in more complex environments relative to their sensor capacity develop MBs with more densely connected nodes, more internal mechanisms, and higher integrated information, indicat-

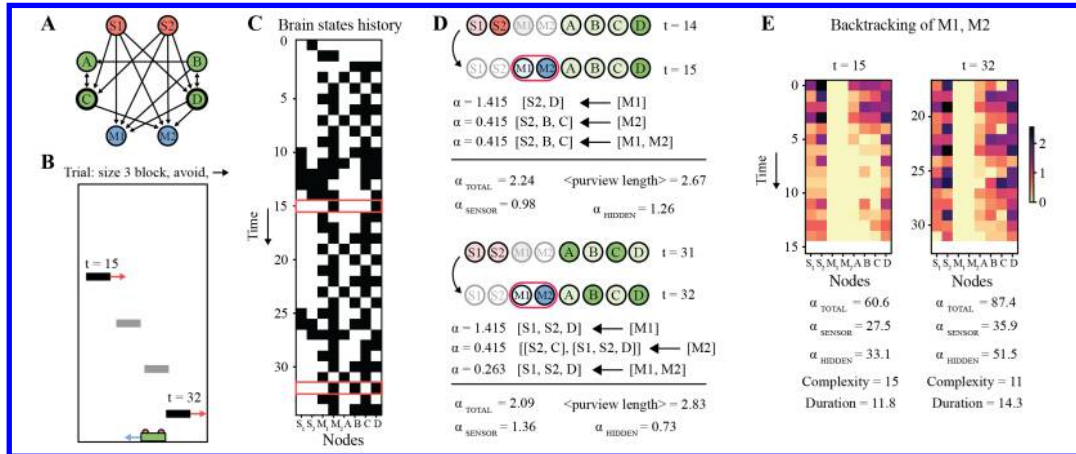


Figure 1: Overview of the task environment, the animat, and analyses used in this study. (A) An animat’s Markov Brain with connections between sensors (red), motors (blue), and hidden nodes (green). Thick borders around nodes C and D indicate self connections. (B) Visualization of the simulated world with the animat at the bottom. (C) Time series of the animat’s brain state during the trial. Black indicates that a node is ‘on’ (‘1’), white means ‘off’ (‘0’). The red frames highlight the two brain states ($t = 15$ and $t = 32$) whose direct causes and causal chains are shown in (D, E). (D) Direct causes of motor states in two example transitions. Dark color shades indicate that a node is ‘on’ (‘1’), lighter shades mean ‘off’ (‘0’). Listed below each transition are the direct actual causes (with corresponding causal strengths α) for the states of M1, M2, and (M1, M2), together with the characteristic quantities used in the statistical analysis. (E) The causal chains obtained by tracing the actual causes of the motor state (M1, M2) = 01 (“move right”) back in time for 15 time steps, starting at time $t = 15$ (left) and $t = 32$ (right). Each row in the pattern shows the summed causal strength (α , indicated by color) that each node contributes to the direct causes of the previous time step. Listed below are the summary measures used to characterize the causal chains.

ing higher intrinsic causal complexity (Oizumi et al., 2014). Here, we exhaustively quantify the direct actual causes of the animats’ actions, and expand the AC framework in order to trace back the causal chain (“causes of causes”) leading up to a given action. In line with Albantakis et al. (2014), we hypothesize that the causal contribution of internal nodes to an animat’s actions and its preceding causal chain will be higher in animats evolved in more complex task conditions. Moreover, we expect that the causal chain will reflect task-specific demands for memory.

Methods

To test our hypotheses, we utilize and extend a formal framework of actual causation to assess and trace back the causes of an agent’s actions and apply it to artificial organisms (“animats”) evolved *in silico* under several task conditions.

The artificial evolution experiments were conducted using the open source software package MABE (Modular Agent Based Evolver) (Bohm et al., 2017; Hintze et al., 2017). Software to identify and quantify direct actual causes is available as part of the PyPhi integrated information toolbox (Mayner et al., 2018). Finally, newly developed scripts that iteratively evaluate the actual causes of previously identified causes are available on GitHub (Comolatti and Juel, 2019).

Data generation: Animat evolution and simulation

Evolution simulations were initialized using MABE’s standard parameter settings with further agent and environment specifications described below. The agent types and task environments investigated in this study were adopted from Albantakis et al. (2014).

Task environment. Animats were evolved in the ‘ComplexiPhi’ world, a 35x16 unit grid with periodic boundary conditions, in which the animat has to move left or right to catch or avoid falling blocks of specific sizes (Figure 1A). Across trials, an animat (3 units wide) is placed at all positions along the bottom of the world, while blocks (one per trial) are positioned in the top left corner, falling to the left or right. The block is ‘caught’ if it overlaps with the animat when it reaches the bottom, otherwise it is ‘avoided’.

Markov Brains (MBs). Animats are equipped with MBs. Each MB consists of up to 8 binary nodes: up to 2 sensors (S1 and S2), 2 motors (M1 and M2), and 4 hidden nodes (A, B, C, and D), whose function and connectivity is specified by hidden markov gates encoded in the animat’s genome, as described in (Hintze et al., 2017). The global update function of the resulting neural network can be described by a state transition probability matrix (TPM). The TPM specifies the probability of an animat’s MB transition-

ing between any two states, thus completely describing its dynamics. Here, we specifically evolved animats with deterministic TPMs. Nevertheless, the causal analysis described below can be applied to probabilistic systems as long as they fulfill the causal Markov condition (Janzing et al., 2013; Albantakis et al., 2019).

During a trial, an animat’s sensor is activated if a block is positioned directly above it at any height, and otherwise remains ‘off’ (‘0’). The two sensors are positioned on each side of the animat, leaving a gap of 1 unit between them. The state of the remaining nodes updates according to the animat’s TPM. However, motors are reset (set to ‘off’ (‘0’)) before each update, effectively excluding any feedback from the motors to the hidden nodes or sensors. The motor state determines the animat’s action at each time step (‘10’: move left, ‘01’: move right, ‘00’ or ‘11’: stand still). This process repeats until the block reaches the bottom of the world, at which point the success of the animat is recorded before the next trial begins.

Genetic algorithm and task fitness. Each evolution simulation is initiated with a population of 100 animats with random circular genomes. At each new generation, this pool of genomes is subject to fitness-based selection and mutation, which allows the animats to adapt to higher fitness across generations. An animat’s fitness, F , is determined by its percentage of successful trials (correctly caught or avoided blocks). After each generation the genetic algorithm draws a new sample of 100 animats (with replacement) based on an exponential measure of F (roulette wheel selection). Before the next generation, the genome of each selected animat mutates using point mutations, deletions, and duplications (Albantakis et al., 2014; Hintze et al., 2017). For each task condition, 50 populations of animats were initialized, and evolved independently for 30,000 generations.

Task conditions. Animats evolve under three conditions varying in difficulty relative to their sensor capacity:

- *Baseline (BL) condition:* catch blocks of size 1, avoid blocks of size 3, using 2 sensors;
- *One sensor (1S) condition:* catch blocks of size 1, avoid blocks of size 3, using only 1 (the left) sensor;
- *Hard task (HT) condition:* catch blocks of size 1 and 4, avoid blocks of size 2 and 3, using 2 sensors.

Thus, animats in the BL and 1S conditions perform the same task, while animats in BL and HT conditions use the same number of sensors. Compared to BL, in conditions 1S and HT additional computations across multiple time steps are necessary to distinguish which blocks have to be caught or avoided. Nevertheless, some internal memory is necessary in all conditions to identify whether the block is moving to the left or right.

Data processing: Causal analysis

The Actual Causation (AC) framework. Here, we briefly describe the relevant concepts of the AC framework by Albantakis et al. (2019). For details and formal definitions of the terminology, we refer to the original publication.

Given a transition $s_{t-1} \prec s_t$ between two subsequent states of a discrete dynamical system of interacting elements S , the AC formalism allows identifying the actual causes of occurrences at time t from the set of occurrences at time $t - 1$ based on a quantitative counterfactual analysis. Here, “occurrence” simply denotes a subset of network nodes in a particular state (for example a motor in state ‘off’: $M1 = 0$). In the AC framework, an occurrence $x_{t-1} \subseteq s_{t-1}$ may only be a cause of another occurrence $y_t \subseteq s_t$, if y_t makes it more likely that x_{t-1} has actually occurred. A “higher-order” occurrence (the joint state of a set of multiple nodes) may specify its own cause x_{t-1} , if it raises the probability of x_{t-1} more than parts of the occurrence do when separated by a partition Ψ (Oizumi et al., 2014; Albantakis et al., 2019). This difference in probabilities indicates the *causal strength* (α) with which y_t determines x_{t-1} . In simplified terms, $\alpha = \min_{\Psi} \left(\log_2 \frac{p(x_{t-1}|y_t)}{\Psi(p(x_{t-1}|y_t))} \right)$, where Ψ partitions $p(x_{t-1} | y_t)$ into $p(x_{1,t-1} | y_{1,t}) \times p(x_{2,t-1} | y_{2,t})$. α can be viewed as the irreducible information that an occurrence specifies about its cause. The *actual* cause of an occurrence y_t is then defined as the subset $x_{t-1}^* \subseteq s_{t-1}$, for which $\alpha(x_{t-1}^*, y_t) = \alpha^{\max}(y_t)$. The set of nodes that constitutes the actual cause ($x_{t-1}^* \subseteq s_{t-1}$) is termed the *cause purview* of the occurrence in question, and the number of nodes in the purview is a measure of how distributed the cause is.

Direct actual causes of motor states. To identify the actual causes of an animat’s action ($M1$ and $M2$ being in a particular state), and to quantify their causal strength, we consider transitions from all inputs to the motors (i.e., the sensors and hidden nodes) at time $t - 1$ to the motors at time t : $(S1, S2, A, B, C, D)_{t-1} \prec (M1, M2)_t$. For a given state of the motors $(M1, M2)_t$, there can be a maximum of three distinct actual causes among all subsets of $(S1, S2, A, B, C, D)_{t-1}$: one for the state of $M1$, one for the state of $M2$, and one for the “higher-order” state $(M1, M2)_t$ (if $(M1, M2)_t$ is irreducible to its partition into $M1$ and $M2$).

As an example, consider the animat transition shown in Figure 1D from $t = 14$ to $t = 15$. Here, $(S2, D)_{14} = (1, 1)$ is the actual cause of $M1_{15} = 0$ with $\alpha = 1.415$ bit. $(S2, B, C)_{14} = (1, 0, 0)$ is the actual cause of $M2_{15} = 1$ with causal strength $\alpha = 0.415$ bit. In addition, $(M1, M2)_{15} = (0, 1)$ also has its own actual cause $(S2, B, C)_{14} = (1, 0, 0)$ with $\alpha = 0.415$ bit. This means that $(M1, M2)_{15} = (0, 1)$ specifies an additional 0.415 bit of information about the state of $(S2, B, C)$ at $t = 14$ compared to $M1_{15} = 0$ and $M2_{15} = 1$ taken individually. Also, note that the cause purviews of the three occurrences contain different sets of

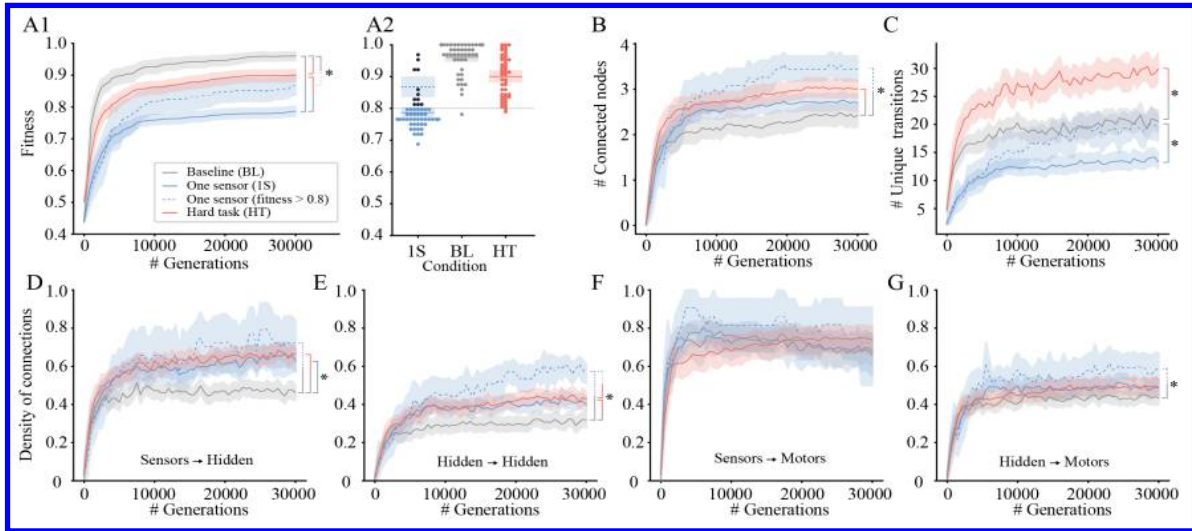


Figure 2: **Fitness, structural, and functional properties of animats.** (A1) Fitness of animats (with final fitness distributions in (A2)), quantified as the ratio of successful trials. (B) #connected nodes measures the number of hidden nodes with incoming and outgoing connections (maximally 4, A-D). (C) #Unique transitions across all trials within a generation estimates the dynamical complexity of the animats. (D)-(G) Density of connections between different node types quantified as the number of connections between the nodes divided by the total possible number of such connections. The lines show the bootstrap resampled means for the population (the shaded area indicates the 95% CI of the mean). Statistically significant difference between conditions are indicated by the colored bars (and stars). The legend in (A) indicates the coloring for all panels.

nodes. For example, hidden node B is part of the actual cause of M2 in this transition, but is not in the cause purview of M1₁₅ = 0. Moreover, there is no need for a node to be ‘on’ (‘1’) to be part of the cause of an occurrence. Finally, S1, for example, is not part of any actual cause even though it has a direct connection to both M1 and M2. In other words, S1₁₄ = 0 does not contribute to “bringing about” the state of the motors at $t = 15$ in this particular transition. However, in the transition from $t = 31$ to $t = 32$, S1₃₁ = 0 does contribute to the causes of (M1, M2)₃₂.

Thus, for every animat, we find the direct actual causes of the motor state in every unique transition and calculate several measures to quantify the degree to which the motor state was caused from within (see legend of Figure 3).

Quantifying the causal chain: the backtracking analysis.

We also perform a ‘backtracking analysis’ of the causes of the motor states, for all transitions in a trial past $t = 15$ (Figure 1E). This analysis amounts to identifying and characterizing the chain of causes leading up to an action.

Having quantified the direct actual causes of a transition $x_{t-1} \prec y_t$ (where x_{t-1} is the state of the sensors and hidden nodes at time $t - 1$ and y_t is the state of the motors at time t), we define a joint purview $z_{t-1} \subseteq x_{t-1}$ as the union of all identified cause purviews. In other words, z_{t-1} is the state at time $t - 1$ of all elements contributing to the actual causes of the motor occurrences, M1 _{t} , M2 _{t} , and (M1, M2) _{t} . In the example shown in Figure 1D (top), this corresponds

to $z_{14} = \{(S2, B, C, D)_{14} = (0, 1, 0, 1)\}$.

Iteratively, we then proceeded to identify and quantify the actual causes for the transitions $x_{t-2} \prec z_{t-1}$, $x_{t-3} \prec z_{t-2}$, etc., thus tracing back the causal chain of the observed motor state at time t . This process is repeated until all cause purviews in the causal account contain only sensors (indicating that the cause is completely “extrinsic”) or upon reaching $t - 15$ (to make results comparable across time steps).

For every animat, we find the causal chain leading to each motor state (after $t = 15$) and calculate several measures aimed to quantify the “intrinsicity” of the backtracking pattern (see legend of Figure 4).

Statistics

Throughout this work, we use bootstrap resampling to estimate means and to calculate 95% confidence intervals between the 2.5th and the 97.5th percentile of the resampled distribution ($CI_{95\%} = [P_{2.5\%}, P_{97.5\%}]$). Although data samples generating overlapping confidence intervals may still differ significantly, we take non-overlapping confidence intervals as an indicator of a statistically significant difference between conditions.

Results

Our simulated evolution experiments under three task conditions reproduce earlier findings reported in (Albantakis et al., 2014). Average fitness is higher in the baseline (BL) condition, than in both the one-sensor (1S) and the hard-task

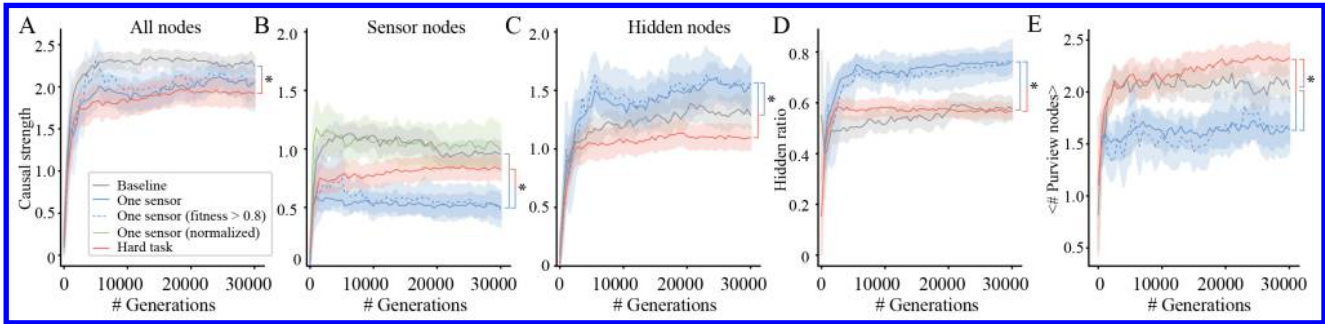


Figure 3: Analysis of direct causes for actions as a function of generation in different task conditions. (A) Total causal strength: summed α across the identified direct causes of M1, M2, and (M1, M2). (B) Sensor causal strength: summed α of the sensor portion in the cause purviews. (C) Hidden causal strength: summed α of the hidden node portion in the cause purviews. (D) Hidden ratio: the ratio of hidden and total causal strength per purview. (B-D) To compute the sensor and hidden portion of the causal strength, we simply multiplied the fraction of sensor and hidden nodes in each actual cause purview by its α value. We did not recompute α values for sensor or hidden node subsets, as these subsets do not correspond to actual causes themselves. (E) Total number of nodes in the cause purviews. All measures are averaged across the unique transitions across times and trials for each animat. Mean, 95% CI, and statistical significance are as in Figure 2.

(HT) conditions (Figure 2A). In addition, the fitness among animats in the HT condition is significantly higher than in the 1S condition, indicating a hierarchy of overall difficulty among the three conditions: BL < HT < 1S. Since fewer animats achieved a final fitness of at least 80% by generation 30,000 for the 1S than the BL and HT conditions (11/50 for 1S vs 49/50 and 48/50 for BL and HT), we use the subset with fitness > 80% as the default population in condition 1S (representing successful evolution), unless otherwise stated.

Structural and dynamic analysis of animats

Generally, measures of the structural and dynamical complexity of the animats increase with fitness throughout evolution (Figure 2; the only exception is a decrease in connections from sensors to motors for 1S and BL in panel F).

In terms of the dynamical complexity exhibited by the animats, those evolved in the HT condition show the largest repertoire of unique state transitions (Figure 2C). Although the number of potential state transitions is smaller in condition 1S due to the reduced number of available nodes (7 vs. 8), the fittest 1S animats still compare to the BL condition.

Structurally, differences from the BL condition are most pronounced in the number of connected nodes (Figure 2B) and the density of connections to the hidden nodes (Figure 2D,E). In particular, the density of connections between hidden nodes reflects the hierarchy of task difficulty. On the other hand, differences in connections to the motors are smaller (Figure 2D,G).

Actual Causation analysis: direct causes of actions

To characterize the causes of the animats' actions across the three task conditions, we first analyzed the direct causes of their motor states for all unique transitions per animat. Here,

only nodes directly connected to the motors may appear in the actual cause purviews of a motor occurrence (see Figure 2D,G). However, whether any particular input node contributes to the cause purviews may vary depending on the transition (see Figure 1D). Nevertheless, as shown in Figure 3, the differences in direct motor causes between conditions do not simply follow the pattern observed for the structural and dynamical analysis (Figure 2).

Animats in the HT condition show a lower total and hidden causal strength (Figure 3A and C), but a higher number of nodes in the cause purviews (Figure 3E) than animats from the BL condition. The 1S condition exhibits lower sensor causal strength, but higher hidden causal strength than conditions BL and HT, and correspondingly, also a significantly higher hidden ratio (Figure 3C-E). Furthermore, the number of purview nodes is significantly lower in the 1S condition than in BL and HT (Figure 3E).

Thus, although there are no clear differences in the density of connections to motors between the conditions, the direct cause analysis seems to distinguish the more difficult conditions from the BL condition by the number of nodes in the purview and the way the causal strength is distributed among different types of nodes, albeit in opposite directions.

Actual Causation analysis: Backtracking analysis.

Next, we investigated whether considering the causal chain ("causes of causes") leading to an animat's action yields additional information about the causal structure of the animat and the causes of its actions. Here, every node with a directed path towards a motor may contribute to the causal chain given sufficient backsteps.

In contrast to the direct cause analysis, in the backtracking analysis both the 1S and HT conditions differ significantly

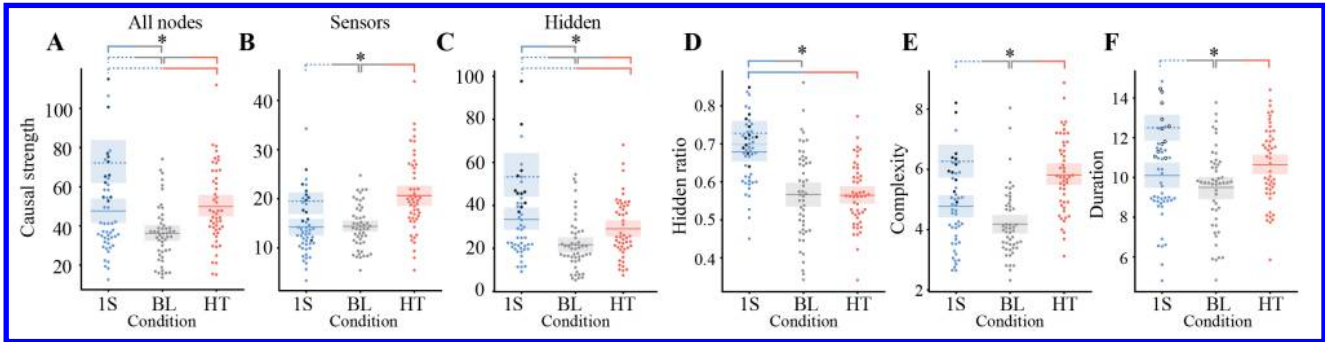


Figure 4: Results of backtracking analysis from animats in the last generation. (A) Total causal strength: summed α across the backtracking pattern. (B) Sensor causal strength: summed α of the sensor portion in the backtracking pattern. (C) Hidden causal strength: summed α of the hidden node portion in the backtracking pattern. (D) Hidden ratio: the ratio of hidden and total causal strength. (E) Complexity of the causal chain: number of unique rows in a backtracking pattern. (F) ‘Duration’ measures the average length of the causal chain by calculating the area of the backtracking pattern and normalizing by the max number of sensor and hidden nodes in the animat (6 for the BL and HT conditions and 5 for the 1S condition). Each dot corresponds to the value for one animat (averaged over times and trials) in the last generation of the evolution. Colors are as in previous figures, with the dark blue dots marking animats from the 1S condition with fitness above 0.8. Shaded patches indicate 95% confidence intervals. Statistically significant differences between populations are indicated by the colored bars (and stars).

from BL in a consistent manner on several measures. First of all, the total, sensor, and hidden causal strengths (Figure 4A-C) of the backtracking patterns are all higher (for the fittest animats) in 1S and HT than in BL. However, only the 1S condition has a higher hidden ratio than BL (Figure 4D), as in the direct cause analysis (Figure 3D). Thus, animats in the HT condition seem to require more involvement from both sensors and hidden nodes than the BL condition to successfully complete their task. On the other hand, animats in the 1S condition seem to compensate their missing sensor by relying more on hidden nodes, resulting in a higher hidden ratio compared to the BL condition. In addition, both the complexity (Figure 4E) and duration (Figure 4F) of the backtracking patterns were higher in the 1S and HT conditions than in the BL condition, reflecting the higher memory requirements in these two conditions.

Finally, the systematic effect of including only the fittest animats from the 1S condition was apparent across all evaluated characteristics of the backtracking patterns, and the hierarchy of task difficulty across the three conditions (BL < HT < 1S) observed in the structural analysis (Figure 2) reemerged for the backtracking analysis.

Discussion

In this paper we applied measures of structural and dynamical complexity, as well as a novel actual causation (AC) framework (Albantakis et al., 2019), to characterize the actions of simulated agents that evolved to solve perceptual-categorization tasks. With the structural and dynamical analysis, we confirmed findings from previous studies indicating that more demanding task conditions lead to the evolution of animats with more interconnected ‘brains’ with a higher

capacity for computations and memory (Albantakis et al., 2014). Using the AC framework, we identified and quantified the direct causes of the animat’s actions as well as their preceding causal chains. To assess the degree to which an animat’s actions were “caused from within” (as opposed to being externally driven through its sensors), we moreover quantified the relative contribution of its hidden and sensor nodes to the cause purviews and backtracking pattern of its motor states.

As discussed in more detail below, we found that the different types of analyses revealed different aspects about the animats’ causal structure and the task conditions under which they evolved. While some measures reflected the hierarchy of task difficulty across the three conditions (BL < HT < 1S), the direct actual causes in particular highlighted differences between 1S and HT.

As hypothesized, the causal chains leading to actions in animats evolved in difficult conditions (hard tasks relative to sensor capacities) with higher demands for internal memory were characterized by higher total causal strength, and reverberated longer within the animat itself than in animats evolved in the simpler baseline condition.

In all, our results suggested that the “intrinsicity” of the direct causes and the causal chain preceding an agent’s actions may serve as a useful indicator of its intrinsic complexity and degree of causal autonomy (see also Marshall et al. (2017); Bertschinger et al. (2008)), while the number of nodes constituting the cause purviews, as well as the complexity and duration of the causal chains may reflect its context-sensitivity.

Differences between conditions

Both the 1S and HT conditions require more internal memory to distinguish which blocks have to be caught or avoided compared to the BL condition. This task property is reflected in the higher number of connected nodes (Figure 2B) and the measures assessed in the backtracking analysis (Figure 4), and also underlies the observed hierarchy in task difficulty (Figure 2A).

However, in the HT condition, more blocks need to be classified than in BL and 1S. Nevertheless, the animats can only make use of the same repertoire of possible actions. Which action is chosen is thus more context-dependent in condition HT than in the other two conditions. The higher context-dependency of condition HT may explain the opposing results between HT and 1S in the direct causes of the animats' actions (Figure 3): the direct cause purviews in the HT condition are larger and more distributed across sensors and hidden nodes, leading to a similar hidden ratio as BL. In contrast, animats in the 1S condition have a higher hidden ratio due to increased memory requirements, but less distributed computation. In sum, our results suggest that the backtracking analysis measures are mostly affected by aspects related to memory requirements, while the direct cause analysis capture context-sensitivity and distributed computation.

Causal analysis

Given that the results of the causal analysis, at least in part, reflect differences in the structural and dynamical properties of the animats, the advantages of a computationally demanding causal analysis may not be immediately clear. For example, one could argue that observed differences between conditions in the causal analysis might be explained by structural properties of the animat populations (such as the longer causal chains with higher hidden causal strength being explained by more, and more densely interconnected, nodes). However, there are at least two reasons for applying the AC analysis in addition to more standard approaches.

First, the AC analysis specifically takes an animat's mechanistic, counterfactual structure into account (see also Shalizi et al. (2005)). Therefore, it may describe aspects of the system that cannot be captured by purely structural, or dynamical, informational, or correlational measures based on observed data only (Marshall et al., 2017). For example, we hardly found significant differences between task conditions regarding the inputs to the motor nodes (Figure 2F,G). Yet, the sensor and hidden causal strength varied significantly across conditions (Figure 3B,C and Figure 4B,C).

Secondly, the AC analysis is applied to each individual transition independently and can identify the causes and intrinsicity of each specific action (motor state), giving a state-dependent description of the animat behavior (see also Lizier et al. (2014) and Beer and Williams (2014)). As can be seen from the example in Figure 1, the same action be-

ing performed by the same animat at two different times may have distinct causes depending on the past states of the rest of the animat. Correlations between an agent's structural/dynamical properties and the results of the causal analysis may thus only become apparent when averaging across many transitions as done here. In future work, it could be investigated how the actual causes of an animat's actions change on a trial-by-trial basis, for specific block sizes, the direction of motion, or whether a block should be caught or avoided. State-independent measures that characterize the animat as a whole cannot assess such questions, but may still serve as useful indicators for a system's capacity for internally caused motor states.

Of course, alternative formalisms for measuring actual causes and causal chains exist (e.g. Datta et al. (2016); Weslake (2015)), which might also be applicable to artificial agents. Nevertheless, the AC framework used here was specifically developed for discrete dynamical systems of interacting elements, such as Markov Brains, which makes it particularly suited for the present study. An interesting question is under which circumstances causal measures effectively exceed dynamical or information-theoretical approaches in elucidating an agent's behavior (e.g., Beer and Williams (2014); Lizier et al. (2014)).

Finally, we did not directly consider issues regarding causal transitivity in this work. The question of whether (and when) the "causes of causes" of an occurrence are themselves causes, is still highly debated. To answer such questions, our proposed approach must be further refined and adjusted accordingly.

Towards a principled definition of agency

On a more philosophical note, the definition of terms used here (such as agent and agency) should be revisited and clarified in future work. For example, throughout this paper we have been using the term agent to refer to the predefined set of nodes that comprise the animats under investigation. And if we define an agent loosely as some system that can sense and interact with its environment, this may not seem problematic. However, if we aim to understand agency more fundamentally, we would also need a way to determine which subset of nodes within a larger set of elements actually constitutes the agent. For this we require a more stringent definition of an agent, as we cannot assume that the borders of the agent itself can always be drawn *a priori*.

One example of such a more stringent definition could be that an agent is (1) an open physical system with stable, self-defined and self-maintained causal borders, with (2) the capacity to perform actions that causally originate within the system itself (Albantakis, 2018) (see also Polani et al. (2016) for an information-based alternative). In this context, it has been shown that the same causal principles on which the actual causation framework is based (Oizumi et al., 2014) can also be used to identify the causal borders of highly inte-

grated subsets of nodes within larger networks, indicating that this type of causal analysis may be used to find autonomous systems that fulfill the first criterion for agency listed above (Marshall et al., 2017) (alternatively, see Friston (2013), Beer (2015), or Kolchinsky and Wolpert (2018)). In summary, we may draw upon and adhere to the theoretical structure and mathematical formalism used in the integrated information theory (IIT) of consciousness (Oizumi et al., 2014) to evaluate both parts of the proposed two-fold definition of agency. Thus, it seems possible that the IIT and AC formalism, taken together, may be used to relate concepts of agency, autonomy, causality, and consciousness within a self-consistent and principled theoretical framework.

Acknowledgements

B.E.J received internationalization support from UiO:Life Science, the University of Oslo, and salary from European Unions Horizon 2020 research and innovation programme under grant agreement 7202070 (Human Brain Project (HBP)). L.A. receives funding from the Templeton World Charities Foundation (Grant #TWCF0196).

References

- Albantakis, L. (2018). A Tale of Two Animats: What Does It Take to Have Goals? pages 5–15. Springer, Cham.
- Albantakis, L., Hintze, A., Koch, C., Adami, C., and Tononi, G. (2014). Evolution of integrated causal structures in animats exposed to environments of increasing complexity. *PLoS Comput. Biol.*, 10(12):e1003966.
- Albantakis, L., Marshall, W., Hoel, E., and Tononi, G. (2019). What caused what? A quantitative account of actual causation using dynamical causal networks. *Entropy*, 21(5):459.
- Beer, R. D. (2003). The Dynamics of Active Categorical Perception in an Evolved Model Agent. *Adaptive Behavior*, 11(4):209–243.
- Beer, R. D. (2015). Characterizing autopoiesis in the game of life. *Artificial Life*, 21(1):1–19. PMID: 25148547.
- Beer, R. D. and Williams, P. L. (2014). Information processing and dynamics in minimally cognitive agents. *Cognitive Science*, 39(1):1–38.
- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: An information theoretic perspective. *Biosystems*, 91(2):331 – 345. Modelling Autonomy.
- Bohm, C., G., N. C., and Hintze, A. (2017). MABE (modular agent based evolver): A framework for digital evolution research. In *Proceedings of the 14th European Conference on Artificial Life ECAL 2017*, volume 14, pages 76–83, Cambridge, MA. MIT Press.
- Comolatti, R. and Juel, B. (2019). actual_agency Toolbox. Available from GitHub[®]. https://github.com/renzocom/actual_agency.
- Datta, A., Garg, D., Kaynar, D., and Sharma, D. (2016). Tracing actual causes. Technical report, Carnegie Mellon University Pittsburgh United States.
- Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S., O’Brien, D., Schieber, S., Waldo, J., Weinberger, D., and Wood, A. (2017). Accountability of AI under the law: The role of explanation. *arXiv*.
- Edlund, J. a., Chaumont, N., Hintze, A., Koch, C., Tononi, G., and Adami, C. (2011). Integrated information increases with fitness in the evolution of animats. *PLoS computational biology*, 7(10):e1002236.
- Friston, K. (2013). Life as we know it. *Journal of The Royal Society Interface*, 10(86):20130475.
- Halpern, J. Y. (2016). *Actual Causality*. MIT Press, Cambridge, MA.
- Hintze, A., Edlund, J. A., Olson, R. S., Knoester, D. B., Schossau, J., Albantakis, L., Tehrani-Saleh, A., Kvam, P., Sheneman, L., Goldsby, H., Bohm, C., and Adami, C. (2017). Markov brains: A technical introduction. *arXiv*.
- Illari, P., Russo, F., and Williamson, J. (2011). *Causality in the Sciences*. Oxford University Press.
- Janzing, D., Balduzzi, D., Grosse-Wentrup, M., and Schölkopf, B. (2013). Quantifying causal influences. *The Annals of Statistics*, 41(5):2324–2358.
- Kolchinsky, A. and Wolpert, D. H. (2018). Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus*, 8(6):20180041.
- Lizier, J. T., Prokopenko, M., and Zomaya, A. Y. (2014). *A Framework for the Local Information Dynamics of Distributed Computation in Complex Systems*, pages 115–158. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Marshall, W., Kim, H., Walker, S. I., Tononi, G., and Albantakis, L. (2017). How causal analysis can reveal autonomy in models of biological systems. *Philos. Trans. A Math. Phys. Eng. Sci.*, 375(2109):20160358.
- Mayner, W. G. P., Marshall, W., Albantakis, L., Findlay, G., Marchman, R., and Tononi, G. (2018). PyPhi: A toolbox for integrated information theory. *PLoS Comput. Biol.*, 14(7):e1006343.
- Metz, C. (2016). How Google’s AI Viewed the Move No Human Could Understand.
- Oizumi, M., Albantakis, L., and Tononi, G. (2014). From the phenomenology to the mechanisms of consciousness: Integrated information theory 3.0. *PLoS Comput. Biol.*, 10(5):e1003588.
- Polani, D., Ikegami, T., and Biehl, M. (2016). Towards information based spatiotemporal patterns as a foundation for agent representation in dynamical systems. *ALIFE/ECAL 2018 Proceedings*, (28):722–729.
- Shalizi, C. R., Haslinger, R., Rouquier, J.-B., Klinkner, K. L., and Moore, C. (2005). Automatic filters for the detection of coherent structure in spatiotemporal systems.
- Silver, D., Huang, A., ..., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- Weslake, B. (2015). A partial theory of actual causation. *Br. J. Philos. Sci.*

Measuring properties of movement in populations of evolved 3D agents

Maciej Komosinski and Konrad Miazga

Institute of Computing Science
Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland
{maciej.komosinski,konrad.miazga}@cs.put.poznan.pl

Abstract

Although determining the similarity of genotypes is often employed in artificial life experiments to measure or control diversity, in practical applications we may often be more interested in similarities of phenotypes. The latter may provide information about the effective diversity in a population, and thus it may be more suitable for diversity estimations and diversity-based search algorithms. A phenotype of a simulated creature can be understood as creature's physiology or its behavior – e.g., body kinematics, movement patterns, or gaits. In this paper, we introduce a set of efficient measures which allow for describing the movement of simulated 3D stick creatures. We use these measures to analyze the results of evolutionary optimization of virtual creatures towards four unique behavioral goals. We show that most solutions obtained for each goal occupy distinct areas of the phenotype space. This suggests that measures defined in this paper create a useful behavioral space for movement-related fitness functions. Finally, we use the introduced measures to visualize how the properties of movement change in populations during the course of evolution.

Introduction

When it comes to the analysis of movement of biological creatures, there exists a large body of research on the importance of locomotion in nature and on the details of its functioning. For humans, human body is obviously the most important, especially in the context of medicine and physical activities (Thorstensson et al., 1984; Sparrow and Newell, 1994; van Ingen Schenau et al., 1994; Courtine and Schieppati, 2004; Thelen and Anderson, 2006; Segers et al., 2007). In other animals, various aspects of locomotion are studied including energetic efficiency, mechanics and control, and these often concern specific orders or species – e.g., cockroaches (Kram et al., 1997), frogs (Ahn et al., 2004), lizards (McElroy et al., 2008), salamanders (Reilly et al., 2006), or cats (Trank et al., 1996; Pearson, 2008).

More general studies concern tetrapod (Biewener, 2006) and hexapod (Full et al., 1991) gaits, and such bodies are often an inspiration for the development of robot body plans (Koditschek et al., 2004). Biological inspirations are also popular in modeling locomotion (Lacquaniti et al.,

2002; Biknevicius and Reilly, 2006; Lee and Harris, 2018) and in experimental and evolutionary robotics (Bongard and Paul, 2000; Krasny and Orin, 2004; Sellers et al., 2004; Ijspeert et al., 2005; Kukillaya and Holmes, 2007; Aydin et al., 2019).

Research works mentioned above are however tailored to specific body plans of existing animals or, in case of biologically-inspired robots, to their artificial counterparts. In this work we introduce and verify more general measures of specific aspects of movement that will work for arbitrary moving 3D stick models. We therefore cannot assume any specific body plan, or even any template that would constrain considered bodies and the movement of their individual parts. Instead, we assume that a body consists of a number of elastic rods and constitutes a three-dimensional graph (Komosinski and Ulatowski, 2009, 2004). We perform simulated evolutionary experiments with different goals, record movements of simulated creatures and then calculate high-level measures in order to investigate how the recorded data differs among different evolutionary goals and individual evolutionary processes.

This research aims to support researchers in automated analysis of movement of both artificial and biological creatures by introducing simple quantitative and objective measures that evaluate individual aspects of body movements. Based on these measures, a number of data analysis techniques can be used, such as clustering, to produce high-level descriptions of large populations of evolved agents without the need for manual human investigation. Various modifications of evolutionary algorithms such as diversity-based search or speciation mechanisms can also be facilitated. Accompanied with statistical and machine learning methods and human expertise, values of the proposed measures of movement may be correlated with existing notions of various movement and gait patterns. They may also be used to quantify how humans perceive moving bodies and biological motion (Castellia et al., 2000; Grill-Spector, 2003; Peelen and Downing, 2007; Pyles and Grossman, 2009).

Measurable properties of movement

In this section we describe what data are acquired when simulated stick creatures move during their lifespan, and then how this data is processed to yield higher level properties of movement. Technical details and extended analyses of these properties are reported in (Gorgolewski et al., 2019).

Low-level raw data

During simulation, each creature provided three series of values C , D_{xy} and D_z defined as follows:

- C – the center of gravity: a series of triplets $c = (x, y, z)$.
- D_{xy} – the dispersion in the xy plane: a series of values d_{xy} .
- D_z – the dispersion in the z dimension: a series of values d_z .

Each location of the center of gravity is described by three coordinates corresponding to the three dimensions. Calculating dispersions required elementary data processing. Since creatures are made of distinguishable parts or vertices (which are connected by joints or edges), we calculate “dispersion” of a creature in a given plane as the weighed standard deviation (Bland and Kerry, 1998) of its body parts. Let us consider c to be the center of gravity and P to be the set containing all body parts. The dispersion in the xy plane will be evaluated as follows (dispersion in the z dimension is calculated analogously):

$$d_{xy} = \sqrt{\frac{\sum_{p \in P} w(p) \cdot ((p_x - c_x)^2 + (p_y - c_y)^2)}{\sum_{p \in P} w(p)}} \quad (1)$$

where $w(p)$ is the importance of part p ; in the experiments reported here it was the mass of p .

High-level description of movement

These three raw data series form the basis for higher-level descriptors of movement. In this paper we employ eight measures describing different qualities of locomotion:

1. Average error of linear regression of position in the xy plane (err_line_xy). A low value indicates movement similar to a straight line.
2. Horizontal oscillation factor (var_dis_xy) – coefficient of variation of dispersion in the xy plane.
3. Vertical oscillation factor (var_dis_z) – coefficient of variation of dispersion in the z dimension.
4. Vertical-to-horizontal oscillation ratio ($sd_dis_z_xy$) – mean ratio of dispersion in z to dispersion in xy .
5. Mean instantaneous speed in xyz ($inst_speed$).

6. Spectral flatness measure (sfm) defined as a geometrical mean of frequency domain of the xyz instantaneous speed divided by its arithmetical mean.
7. The frequency of the highest amplitude of the xyz speed (f_max).
8. The maximal correlation of the xyz speed signal with its offset ($autocorr_max$).

Let us now describe in more detail how these eight properties are calculated. First we define two helpful quantities, ΔC and F . Let $E(p_1, p_2)$ denote the Euclidean distance between points p_1 and p_2 . Then,

$$\Delta C = \{c'_i : c'_i = E(c_i, c_{i+1}); c_i \in C, i = 1..|C| - 1\} \quad (2)$$

is a vector of instantaneous speed of a creature, and

$$F = \text{DFT}(\Delta C) \quad (3)$$

is the discrete Fourier transform of ΔC .

1. The average error of two-dimensional Euclidean regression in the xy plane tells us how well can the path of movement of creature’s center of mass in the horizontal plane be approximated with a straight line. For creatures whose movement cannot be approximated by a straight line, the value of err_line_xy will depend on both the length of the recorded series and the average instantaneous speed of the creature.

To calculate this measure it is necessary to find the Euclidean regression line $y = ax + b$ (Stein, 1983). This is the line for which the sum of the squares of distances from the points of the trajectory is minimal. The average error err_line_xy is calculated as the mean over all ε_i , where

$$\varepsilon_i = \left| \frac{y_i - (ax_i + b)}{\sqrt{a^2 + 1}} \right|. \quad (4)$$

2. The horizontal oscillation factor evaluates the variability of the body dispersion in the xy plane. A creature that has a high horizontal oscillation factor tends to put a lot of effort in the horizontal limb moves like walking, crawling and snake-style swimming.

$$var_dis_xy = \frac{\sigma(D_{xy})}{d_{xy}} \quad (5)$$

3. The vertical oscillation factor evaluates the variability of the body dispersion in the z dimension. Intuitively, a high vertical oscillation factor may indicate a jumping type of movement, while lower values may characterize crawling creatures.

$$var_dis_z = \frac{\sigma(D_z)}{d_z} \quad (6)$$

4. The vertical-to-horizontal oscillation ratio compares the range of movement in vertical dimension to the range of movement in the horizontal plane.

$$sd_dis_z_xy = \frac{\sigma(D_z)}{\sigma(D_{xy})}. \quad (7)$$

5. The mean instantaneous speed in xyz is calculated as

$$inst_speed = \overline{\Delta C}. \quad (8)$$

6. Spectral flatness measure (sfm) describes the uniformity of the Fourier transform of creature's speed. Creatures whose speed in time resembles a sinusoid are assigned a low sfm value, while creatures exhibiting irregular speed will be assigned a higher sfm value. The value of sfm depends on the length of the recorded series of values. For creatures that do not move at all, sfm becomes 0. Spectral flatness measure is calculated as

$$sfm = \frac{e^{\overline{\ln(F)}}}{\overline{F}}. \quad (9)$$

7. The most significant frequency is calculated as the highest non-zero frequency in the spectrum of the creature's speed:

$$f_max = \arg \max(F) \quad (10)$$

8. The maximal correlation of the xyz speed signal with its offset ($autocorr_max$) may be calculated according to the following algorithm:

- 1: $Speed \leftarrow \text{vector}(|C| - 1)$
- 2: **for** $i = 0..|C| - 3$ **do**
- 3: $Speed[i] = E(C[i], C[i + 1])$
- 4: **end for**
- 5: $AutoCorr \leftarrow \text{vector}(|C|/2 - 1)$
- 6: $acLen \leftarrow |AutoCorr|$
- 7: $AutoCorr[0] \leftarrow 1$
- 8: **for** $i = 1..acLen - 1$ **do**
- 9: $AutoCorr[i] \leftarrow \text{corrcoef}(Speed[0 : acLen], Speed[i : acLen+i])$
- 10: **end for**
- 11: $lowerBound \leftarrow 0$
- 12: **for** $i = 1..acLen - 1$ **do**
- 13: **if** $AutoCorr[i] > AutoCorr[i - 1]$ **then**
- 14: $lowerBound \leftarrow i$
- 15: **break**
- 16: **end if**
- 17: **end for**
- 18: $autocorr_max \leftarrow \max(AutoCorr[lowerBound : acLen])$

The `corrcoef` function used in line 9 calculates the correlation coefficient between the original and the delayed

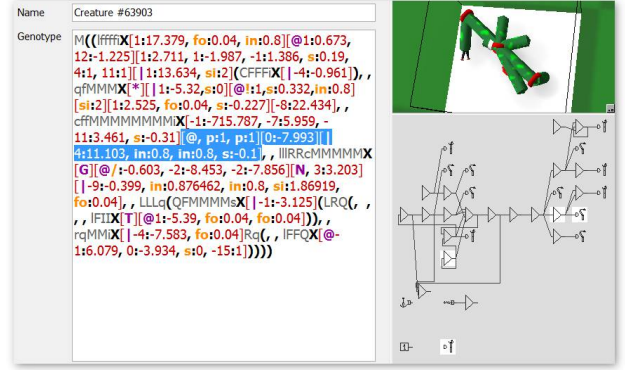


Figure 1: An example of a Framsticks creature encoded with a “f1” genotype. A section of the genotype is highlighted (blue background), and the corresponding neurons are also highlighted (white background under four neurons).

signal (which, for our purpose, is the vector of instantaneous speed). The **for** loop in lines 11-17 determines *lowerBound* – the index of the first local minimum in the *AutoCorr* vector. The offsets lower than this index are then excluded from the search for the highest correlation (line 18).

The eight high-level properties defined above are used in the following experiments as eight descriptors of movement. The movement of each creature during its entire lifespan is described by these eight scalar values.

Simulation experiments

This section describes three analyses of results from two large-scale experiments. In simulations, the Framsticks environment was used with the “f1” genetic encoding (Komosinski and Ulatowski, 2009, 2004, 2019). This direct encoding represents a creature as a string of symbols, and each symbol corresponds to some element of body or brain or to their properties. For example, “X” means a stick (i.e., two connected parts – or, in graph theory, an edge that connects two vertices). Parentheses denote branching, and letters can modify individual properties of sticks (for example, “1” makes a stick shorter, while “L” makes it longer). Neurons are encoded in square brackets as lists of input indexes and their weights, and arbitrary topologies of neural networks can be evolved. An example of a genotype and the corresponding phenotype is shown in Fig. 1. Mutations modify individual aspects of a genotype, and two-point crossing over is used. Mechanical simulation uses a simple finite element method.

In both experiments, there was 64% of mutations, 16% of crossing overs, and 20% of clones. Cloning was used so that fitness could be averaged due to introduced random noise in simulation to make evaluation more robust and avoid over-fitting.

Movement evolved for four fitness functions

In this experiment, the movement of 400 creatures was evaluated. These 400 creatures were evolved in independent evolutionary experiments with four fitness functions (100 independent evolutionary processes per one fitness function). The fitness functions were: maximizing velocity on land, maximizing velocity in water, maximizing vertical position of the center of mass without neural network enabled (“passive”), and maximizing vertical position of the center of mass with neural network enabled (“active”). In all these evolutionary experiments, gene pool size was 1000, and tournament selection was used with 3 genotypes in the tournament. Evolution consisted of two phases: low-pressure phase where the negative selection was random, and high-pressure phase where the negative selection was removing the worst genotype from the gene pool. Each phase ended when 10^6 of non-improving individuals were created via mutation or crossing over and evaluated.

The set of 400 individuals obtained this way was earlier analyzed using phenotypic (body morphology) similarity (Komosinski, 2017) and the behaviors of these creatures are demonstrated at <https://www.youtube.com/watch?v=1o4vL7gOuYk>. Here we will analyze the movement of these creatures; note that the algorithm that evaluates movement is unaware of the fitness functions used to evolve the creatures. Different colors and symbols visible in plots are only used for easier interpretation of results.

Fig. 2 demonstrates values of properties calculated for each creature, divided into four populations, based on the evolutionary goal towards which they were evolved. Values of all properties were normalized to lie within the range of [0; 1]. The properties allow for the distinction between different evolutionary goals, for example *inst_speed* facilitates the distinction between velocity-based evolutionary goals and the two types of vertical position goals. The distinction between the goal of high velocity on land and in water is facilitated by the *var_dis_z* and *sfm* properties of creatures.

Multidimensional scaling (MDS) allowed us to gain more insight into the behaviors exhibited by the examined populations. Although in this experiment equal weights have been assigned to all properties, in general different values of weights can be employed to differentiate the importance of each property. Figs. 3 and 4 show the result of applying, respectively, three- and two-dimensional MDS to the original eight-dimensional space of properties. These two figures confirm that, although the populations are not always perfectly separated, based on the eight properties of movement it is possible to separate in the space of behaviors creatures evolved towards different evolutionary goals. One interesting insight provided by the MDS analysis is the division of the “vertposA” (vertical position, active) population into two distinct subpopulations – two “species”. The first species of “verposA” resides in a close vicinity of the “vertposP” (vertical position, passive) population, which implies that it

contains structures focused on obtaining high vertical position mostly through the evolution of their static structure. The second species can be noticed much closer to “velLand” (velocity on land) population, which suggests that the second species obtains its high vertical position mostly through a jumping movement. Similarly, two distinct species can be observed for the “velWater” population.

Variability of the properties of movement over evolutionary time

In this experiment, 100 independent evolutionary runs were performed to maximize velocity on land. Population size was 50, positive selection was tournament with the tournament size of 3, and evolution consisted of one stage that lasted for 100,000 evaluations of new genotypes.

Fig. 5 presents the course of three selected evolutionary runs; each column shows how the best creature in the population changes its values of a given property over time; time passes downwards in all plots. We distinguish five main patterns of change over time:

Stagnation – occurs when the population converges to a local optimum, or the selective pressure (or genetic drift) guides the population over a part of the landscape where a given property does not change, or its change would lead to the decrease in fitness.

Gradual change – occurs when the fitness landscape facilitates continuous improvement through small changes.

Sudden change – occurs when some new, fit and significantly unique creature appears in the population. This pattern of change is an important feature of the theory of punctuated equilibrium.

Outliers – localized, unusual values of a property may correspond to unusual, nonterminal variations of the most fit creature in the population. These variations die out because of their sub-par fitness or as an effect of a genetic drift.

Noise – may occur when the population traverses the fragment of the fitness landscape where the value of a given property is unstable.

Each property tends to exhibit some patterns more prominently than others. While *var_dis_xy* and *inst_speed* often change gradually, *var_dis_z* tends to stagnate and change only in sudden bursts. This correspond to the fitness landscape resulting from the evolutionary goal of vertical movement – while instantaneous speed and horizontal range of movement are directly linked to this specific task, the range of vertical movement is not, and so evolution does not explore this dimension of behavior.

Interestingly, two of the proposed properties, *err_line_xy* and *autocorr_max*, tend to demonstrate noisy behavior. This

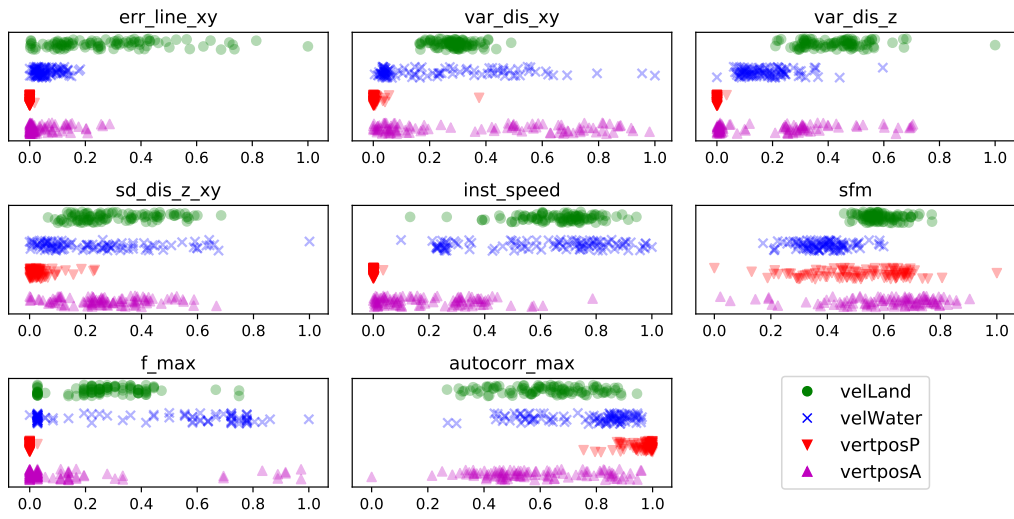


Figure 2: Variability of individual movement properties for populations evolved towards four different goals. A small amount of random vertical jitter is added to reveal individuals that share the same values of movement properties. The vertical axis is only introduced to separate the four populations in order to improve the readability of the plot.

may be a result of a chaotic movement of the simulated creature, as both of these properties are strongly dependent not only on the general characteristics of creature's movement, but also on the specifics of a given simulation run, such as slightly different initial conditions or random noise.

In Fig. 5, f_{max} appears to alternate between a few distinct values; this is probably because there are a few high peaks of a similar amplitude in the frequency spectrum. In such a case, the choice of the value of f_{max} will be highly random.

Fig. 6 depicts 100 overlaid evolutionary runs, similar to

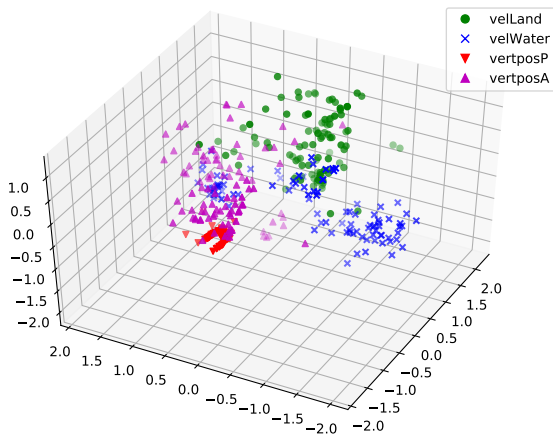


Figure 3: Movement of 400 creatures after multidimensional scaling from the 8D space to 3D. 56.82% of the original variance is preserved.

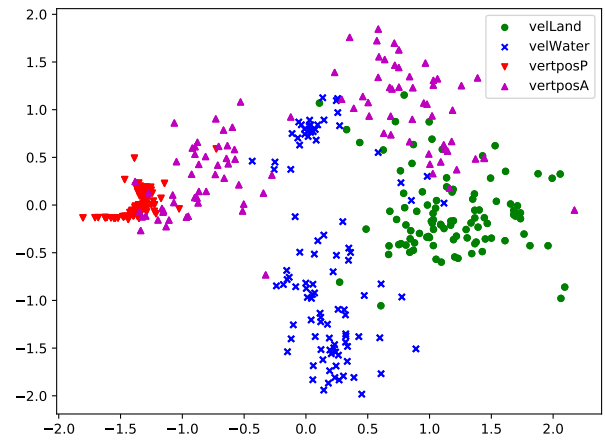
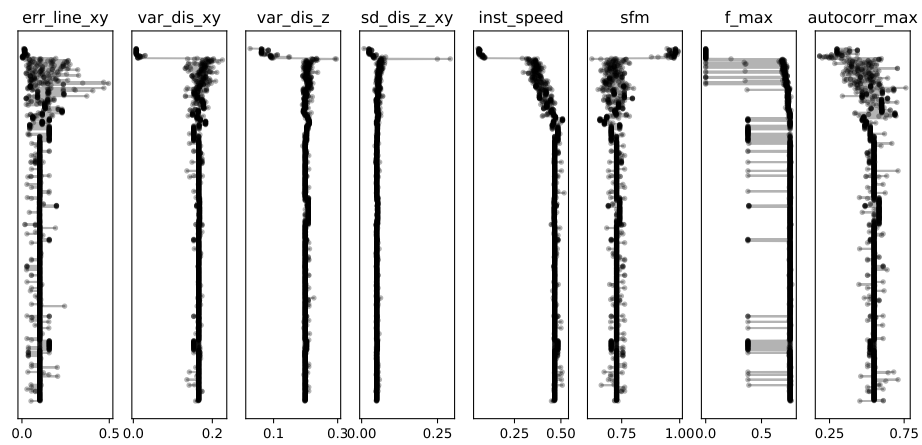
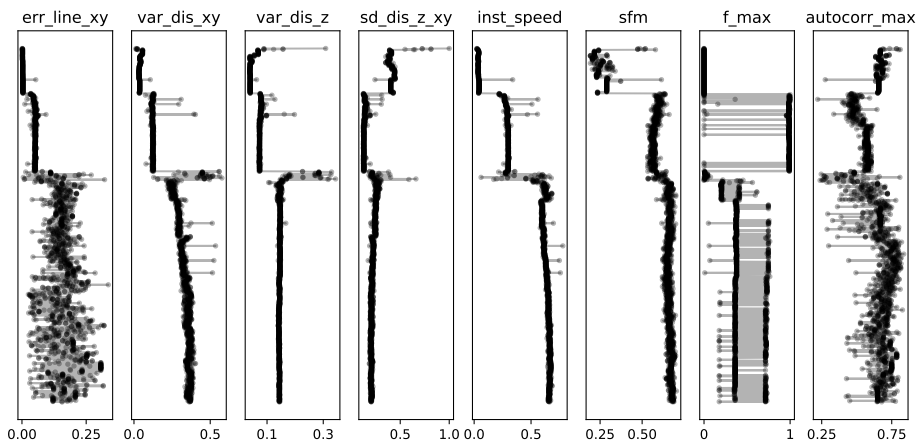


Figure 4: Movement of 400 creatures after multidimensional scaling from the 8D space to 2D. 50.38% of the original variance is preserved.

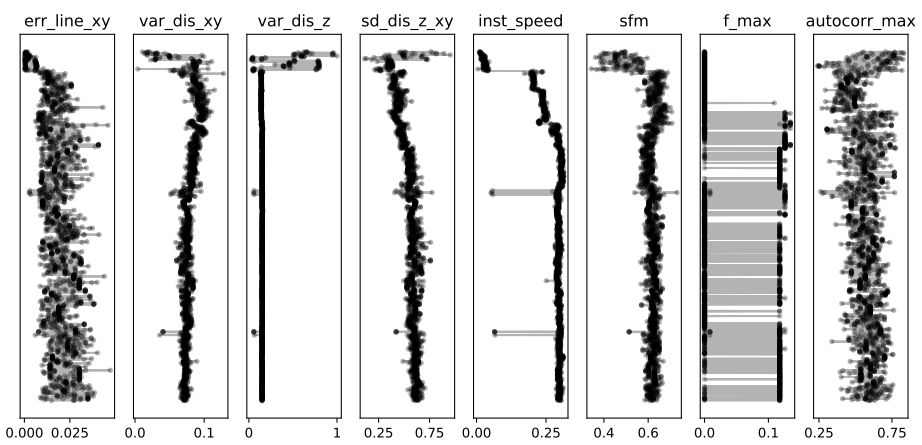
the ones shown in Fig. 5 (where only a single run was shown in each plot). This allows one to analyze the distribution of values of the introduced movement properties over evolutionary time. The probability distributions of the values of properties reveal the existence of bands of more (or less) probable values. Such bands correspond to specific attractors in the behavioral space for the fitness function used here, i.e., maximizing velocity on land. Further analysis of the distributions of behaviors, especially the one that takes into consideration interdependencies between the values of mul-



(a)



(b)



(c)

Figure 5: Eight properties of movement for three selected evolutionary runs. Evolutionary time passes from top to bottom of the plots. Each dot represents a value of a given movement property of a creature that appeared during evolution.

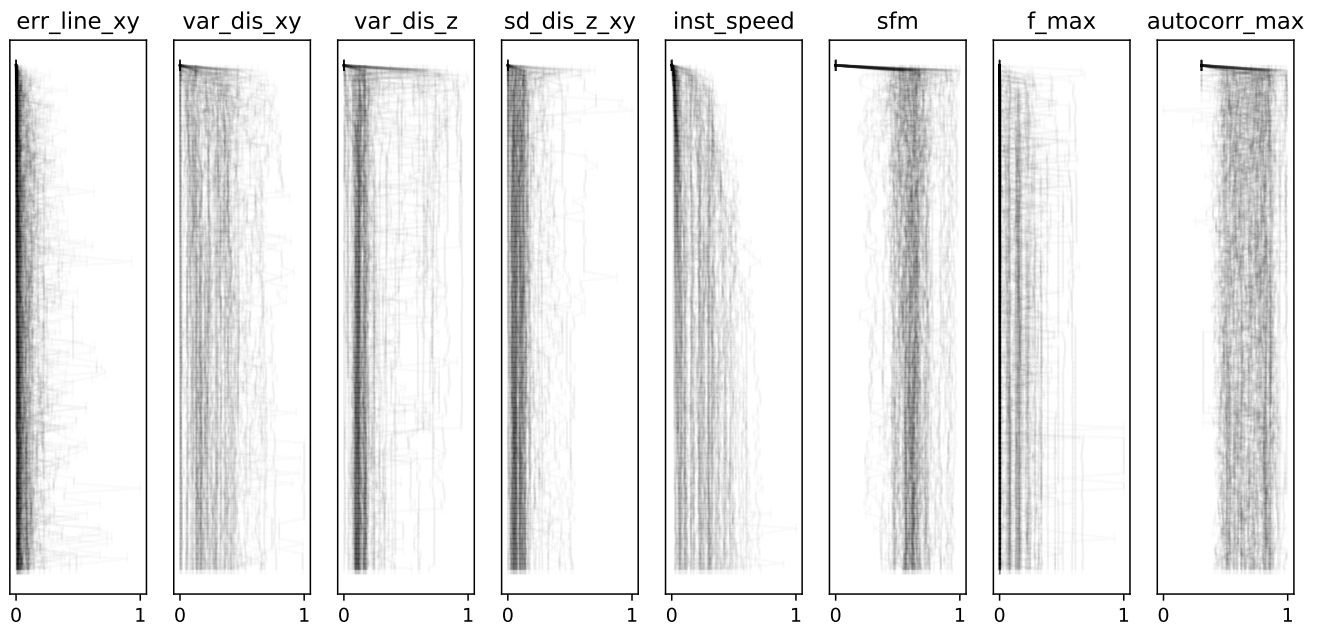


Figure 6: Values of eight movement properties changing in evolutionary time (from top to bottom) for 100 independent evolutionary runs.

tiple properties (e.g., multidimensional scaling) could reveal the presence of sub-species within any specific species, and subsequently allow for the automatic classification thereof.

Summary

In this paper we have introduced a set of eight quantitative properties which describe the movement of simulated three-dimensional creatures. The measures were employed to evaluate the behavior of evolved 3D stick creatures, but they can be easily generalized to other shapes of creatures, including solids.

We have calculated movement properties for four populations, each consisting of 100 creatures evolved using a unique fitness function, and we have shown that these populations are separable in the space of the proposed movement properties. This suggests that the space of behaviors defined by these eight measures can facilitate automatic classification of “species” (i.e., goals towards which the creatures were evolved) or even “subspecies” (specific strategies of achieving these goals). The proposed set of properties can be used not only for automatic classification of evolved or designed behaviors, but also as a space which can be exploited by diversity-based search algorithms (Mouret and Doncieux, 2009), such as the novelty search (Lehman and Stanley, 2011), or for speciation in simulated evolutionary processes.

We have also analyzed how the movement properties of simulated creatures change during their evolution and demonstrated that attractors exist in the space of behaviors,

which make some values of the proposed properties more likely to occur than others.

Some of the properties introduced in this paper could be potentially improved by increasing their stability or identifying and decreasing the amount of dependencies between their values. These issues will be our further work, and we will look for other reliable and informative properties of movement that can be efficiently computed for arbitrary 3D stick agents.

References

- Ahn, A. N., Furrow, E., and Biewener, A. A. (2004). Walking and running in the red-legged running frog, *Kassina maculata*. *Journal of Experimental Biology*, 207(3):399–410.
- Aydin, Y. O., Rieser, J. M., Hubicki, C. M., Savoie, W., and Goldman, D. I. (2019). 6 – physics approaches to natural locomotion: Every robot is an experiment. In Walsh, S. M. and Strano, M. S., editors, *Robotic Systems and Autonomous Platforms*, Woodhead Publishing in Materials, pages 109–127. Woodhead Publishing.
- Biewener, A. A. (2006). Patterns of mechanical energy change in tetrapod gait: pendula, springs and work. *Journal of Experimental Zoology Part A: Comparative Experimental Biology*, 305A(11):899–911.
- Biknevicius, A. R. and Reilly, S. M. (2006). Correlation of symmetrical gaits and whole body mechanics: debunking myths in locomotor biodynamics. *Journal of Experimental Zoology Part A: Comparative Experimental Biology*, 305A(11):923–934.
- Bland, J. M. and Kerry, S. M. (1998). Weighted comparison of means. *British Medical Journal*, 316.

- Bongard, J. C. and Paul, C. (2000). Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In Meyer, J.-A., editor, *Proceedings of the Sixth International Conference on Simulation of Adaptive Behaviour*, pages 420–429. MIT Press.
- Castellia, F., Happé, F., Fritha, U., and Frithc, C. (2000). Movement and mind: A functional imaging study of perception and interpretation of complex intentional movement patterns. *NeuroImage*, 12(3):314–325.
- Courtine, G. and Schieppati, M. (2004). Tuning of a basic coordination pattern constructs straight-ahead and curved walking in humans. *Journal of Neurophysiology*, 91(4):1524–1535.
- Full, R. J., Blickhan, R., and Ting, L. H. (1991). Leg design in hexapedal runners. *Journal of Experimental Biology*, 158(1):369–390.
- Gorgolewski, K., Komosinski, M., Miazga, K., Rosiński, K., and Rychly, P. (2019). Properties of movement of 3D agents. Research report RA-1/2019, Poznan University of Technology, Institute of Computing Science. <http://www.framsticks.com/files/common/PropertiesOfMovementOf3DAgents.pdf>.
- Grill-Spector, K. (2003). The neural basis of object perception. *Current opinion in neurobiology*, 13(2):159–166.
- Ijspeert, A. J., Crespi, A., and Cabelguen, J. M. (2005). Simulation and robotics studies of salamander locomotion. Applying neurobiological principles to the control of locomotion in robots. *Neuroinformatics*, 3(3):171–196.
- Koditschek, D. E., Full, R. J., and Buehler, M. (2004). Mechanical aspects of legged locomotion control. *Arthropod Structure and Development*, 33(3):251–272.
- Komosinski, M. (2017). Applications of a similarity measure in the analysis of populations of 3D agents. *Journal of Computational Science*, 21:407–418.
- Komosinski, M. and Ulatowski, S. (2004). Genetic mappings in artificial genomes. *Theory in Biosciences*, 123(2):125–137.
- Komosinski, M. and Ulatowski, S. (2009). Framsticks: Creating and understanding complexity of life. In Komosinski, M. and Adamatzky, A., editors, *Artificial Life Models in Software*, chapter 5, pages 107–148. Springer, London, 2nd edition.
- Komosinski, M. and Ulatowski, S. (2019). Framsticks web site. <http://www.framsticks.com>.
- Kram, R., Wong, B., and Full, R. J. (1997). Three-dimensional kinematics and limb kinetic energy of running cockroaches. *Journal of Experimental Biology*, 200(13):1919–1929.
- Krasny, D. P. and Orin, D. E. (2004). Generating high-speed dynamic running gaits in a quadruped robot using an evolutionary search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1685–1696.
- Kukillaya, R. P. and Holmes, P. J. (2007). A hexapedal jointed-leg model for insect locomotion in the horizontal plane. *Biological cybernetics*, 97(5–6):379–395.
- Lacquaniti, F., Ivanenko, Y. P., and Zago, M. (2002). Kinematic control of walking. *Archives italiennes de biologie*, 140(4):263–272.
- Lee, D. V. and Harris, S. L. (2018). Linking gait dynamics to mechanical cost of legged locomotion. *Frontiers in Robotics and AI*, 5:111.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- McElroy, E. J., Hickey, K. L., and Reilly, S. M. (2008). The correlated evolution of biomechanics, gait and foraging mode in lizards. *Journal of Experimental Biology*, 211(7):1029–1040.
- Mouret, J.-B. and Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation*. IEEE.
- Pearson, K. G. (2008). Role of sensory feedback in the control of stance duration in walking cats. *Brain research reviews*, 57(1):222–227.
- Peelen, M. V. and Downing, P. E. (2007). The neural basis of visual body perception. *Nature Reviews Neuroscience*, 8:636–648.
- Pyles, J. A. and Grossman, E. D. (2009). Neural adaptation for novel objects during dynamic articulation. *Neuropsychologia*, 47(5):1261–1268.
- Reilly, S. M., McElroy, E. J., Odum, R. A., and Hornyak, V. A. (2006). Tuataras and salamanders show that walking and running mechanics are ancient features of tetrapod locomotion. *Proceedings of the Royal Society B: Biological Sciences*, 273(1593):1563–1568.
- Segers, V., Aerts, P., Lenoir, M., and Clercq, D. D. (2007). Dynamics of the body centre of mass during actual acceleration across transition speed. *Journal of Experimental Biology*, 210(4):578–585.
- Sellers, W. I., Dennis, L. A., W.-J., W., and Crompton, R. H. (2004). Evaluating alternative gait strategies using evolutionary robotics. *Journal of Anatomy*, 204(5):343–351.
- Sparrow, W. A. and Newell, K. M. (1994). The coordination and control of human creeping with increases in speed. *Behavioural brain research*, 63(2):151–158.
- Stein, Y. (1983). Two dimensional Euclidean regression.
- Thelen, D. G. and Anderson, F. C. (2006). Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of biomechanics*, 39(6):1107–1115.
- Thorstensson, A., Nilsson, J., Carlson, H., and Zomlefer, M. R. (1984). Trunk movements in human locomotion. *Acta Physiologica Scandinavica*, 121(1):9–22.
- Trank, T. V., Chen, C., and Smith, J. L. (1996). Forms of forward quadrupedal locomotion. I. A comparison of posture, hindlimb kinematics, and motor patterns for normal and crouched walking. *Journal of Neurophysiology*, 76(4):2316–2326.
- van Ingen Schenau, G. J., de Koning, J. J., and de Groot, G. (1994). Optimisation of sprinting performance in running, cycling and speed skating. *Sports Medicine*, 17(4):259–275.

Edge of Chaos: Artificial Life based interactive art installation

Vasilija Abramovic¹ and Ruairi Glynn²

¹Interactive Architecture Lab, The Bartlett School of Architecture, UCL
Faculty of Architecture, Czech Technical University

² Interactive Architecture Lab, The Bartlett School of Architecture, UCL
va@interactivearchitecture.org

Abstract

Our paper presents an art installation exhibited internationally throughout 2018 at La Gaité Lyrique gallery in Paris, KIKK Festival in Namur, and Cinekid Festival in Amsterdam. The collaborative project between artists and physicists examined the aesthetic possibilities of cellular automata (CA) driven kinetic objects to make theories of emergent life tangible to audiences of children and adults alike. Here we present our approach encompassing: narrative, material, hardware and computation strategies. The “Edge of Chaos” installation, inspired by Christopher Langton’s theory, is an artistic realization of emergent systems at the scale of inhabitable architectural space. The use of on artificial life approaches to behaviour offers distinctly different audience experiences to those in responsive environments that follow master-slave interaction paradigms typically found in Human Computer Interaction fields. Through the use of narrative these emergent behavioural systems and their implications for conceptions of life are articulated in way that is engaging and playful. Through the use of recent metamaterial research the project also provokes discussion on the potential of these material systems to lead to new forms of artificial life.

Introduction

This paper describes an interactive installation that was touring three venues in Europe in 2018, France, Belgium and the Netherlands offering an inhabitable interactive environment based on Christopher Langton’s theory of the “Edge of Chaos”. The Edge of Chaos project is the winning proposal to an international competition calling artists and architects to offer their “perspectives on the technological, dematerialized, complex environments and ubiquitous machines that have become natural elements in our lives” (Competition Brief, 2017). The project is situated at the crossroad between art, design and physics, and this paper’s primary focus is on the interaction and artistic aspects of the work that reflect the use of reconfigurable metamaterials, distributed control, and bottom-up strategies for designing responsive environments. The resulting built installation reflects on our Interactive Architecture Lab’s critical position on so called “Smart Architecture” and corporate led “Internet of Things” which while espousing notions of ecology, remain dominantly driven by top-down strategies to controlling the behaviour of the places we live, work and play.

We have been examining ways of creating intelligent environments not through classical Artificial Intelligence techniques but rather through Artificial Life, embracing the aesthetic emergent possibilities that can spontaneously arise from this approach. The competition brief provoked a response by our team to the statement, “We are entering an era of hybrid ecology: AI as landscape, networks as biotopes, data as organisms and media as humus. Where do we go from here?”. We were drawn to the theory of the Edge of Chaos that examines the balancing point where “the components of a system never quite lock into place, and yet never quite dissolve into turbulence either... the edge of chaos is where life has enough stability to sustain itself and enough creativity to deserve the name of life” (Waldrop 1993). Our art installation is built upon a custom CA algorithm that embodied this notion and creates a space for contemplation about the boundary relationships between chaos and order. In the context of the competition we found a means to make these ideas of emergent life more tangible in four integrated parts, namely, narrative, material, hardware and computation.



Figure 1. Edge of Chaos, La Gaité Lyrique gallery Paris

Methods

Narratively, there are three features within the installation (Figure 1). At its centre, a kinetic tree (animated by servo motors and RGB LEDs) that represents “Life” (the peak of the Edge of Chaos). It is surrounded by an inert “Cloud” representing the vast unorganised matter of an entropic universe (Chaos), and between them an interactive surface that represents the boundary of the “Edge of Chaos”. The

movement and proximity of inhabitants of the installation measured by proximity sensors activates the “Edge of Chaos” triggering the surface to light up and physically transform. Local interactions by inhabitants stimulate our custom CA triggering chain reactions throughout the surface, that depending on the level of interaction and current state of the surface will produce more ordered patterns, or more chaotic sequences. When the “Edge of Chaos” surface becomes highly activated the tree “comes to life” blossoming into full colour and performing its most dramatic kinetic movements.

Meta-materials are materials which get their properties from their structure, rather than their chemistry (Overvelde et al. 2017). Translating this into spatial design we are able to apply this geometric strategy into designing flexible and transformable spaces with tuneable functionality.



Figure 2. Metamaterial Structures of “Tree”

The initial choice of cellular automata algorithm was Conway’s classic “Game of Life” (1970) but the patterns were not as desirable as hoped. The manner in which kinetic elements would remain on or fall into oscillating rhythms actually detracted from the pleasure audiences got stimulating the wall from an inactive state. This improved legibility of human agency in the space. To create stable patterns, we implemented an additional state leading to a total of three states for each cell: listening (0), active (1), and sleeping (2) (Hoekstra et al. 2010). Different behaviour was achieved by implementing a stochastic response in the behaviour of the CA (Hoekstra et al 2010). During each time step, each listening cell checks the states of the neighbours. If the total sum of the neighbours being active is equal or higher than n , the cell becomes active only with some probability p . Once active, the cell remains active for s_a steps, after which it goes to sleep. Finally, after sleeping for s_s steps, the cell starts listening again. Given a network of cells with connections to their neighbours, changing these four parameters tunes the behaviour of the total system.

As the installation was intended to run for months at a time, interactive motion and lighting needed to be limited to periods where there is stimulation in the environment by visitors to the gallery space. When there are no inhabitants, the installation must in effect “die out”. The results of this enquiry demonstrate alternative strategies for maximising aesthetic use of CA rules while minimising the live time of kinetic elements. This approach also suggests modes of

environmental activation and deactivation that are robust as well as playful and suggests approaches to intelligent architecture based on Alife principles.

Conclusion

The Edge of Chaos installation, demonstrates simple but compelling appeal that responsive kinetic objects elicit. Observing the emerging patterns of our installation we can recognise Langton’s mathematic and theoretical principles of “Phase Transition” where structures are able to grow, split and recombine, in spontaneous and surprising ways. This area represents neither Order nor Chaos but lives in between thus never becoming predictable nor too unpredictable to lose the interest of observers.

The open ended and semi-unpredictable behaviour of the Edge of Chaos installation encourages continued play as inhabitants search for understanding in a field of interactions. While controlled by simple rules it displays complexity by virtue of emergent properties and complex and changing environmental stimuli. Our implementation of custom CA code is effective on aesthetic grounds but also in minimising wear and tear of multiple long running exhibitions which contributes to possible applications in longer term architectural applications. Our narrative and spatial approach proves a novel and tangible approach to communicating abstract mathematical theories of complexity. It is also a celebration of the creative possibilities of collaborating across science and architecture.

References

- Beesley, P. (2014). *Near-living architecture: Work in progress from the Hylozoic Ground Collaboration 2011–2013*. Toronto: Riverside Architectural Press.
- Conway, J. (1970). *The game of life*. Scientific American
- Frazer, J. (1995). *An evolutionary architecture*. Architectural Association London
- Haque, U. (2006). *Open Burble*, London
- Ilachinski, A. (2001). *Cellular automata: a discrete universe*. World Scientific Publishing Inc.
- Langton, C. (1997). *Artificial life: An overview*. Addison-Wesley
- Levy, S. (1993). *Artificial life: A report from the frontier where computers meet biology*. Random House Inc..
- Overvelde, J., Hoberman, C., Bertoldi, K., Weave, J. (2017). *Rational design of reconfigurable prismatic architected materials*. Springer Nature
- Waldrop, M.M. (1993). *Complexity: The emerging science at the edge of order and chaos*. Simon & Schuster, Inc.

Acknowledgements

Edge of Chaos installation is a collaboration between researchers and designers Vasilija Abramovic and Ruairi Glynn (Interactive Architecture Lab) and scientist Bas Overvelde (AMOLF).

Autopoiesis in digital learning design: Theoretical implications in education

Claudio Aguayo¹

¹Auckland University of Technology
claudio.aguayo@aut.ac.nz

Abstract

Today's mobile and smart technologies have a key role to play in the transformative potential of educational practice. However, technology-enhanced learning processes are embedded within an inherent and unpredictable complexity, not only in the design and development of educational experiences, but also within the socio-cultural and technological contexts where users and learners reside. This represents a limitation with current mainstream digital educational practice, as digital experiences tend to be designed and developed as 'one solution fits all' products, and/or as 'one-off' events, failing to address ongoing socio-technological complexity, therefore tending to decay in meaningfulness and effectiveness over time. One ambitious solution is to confer the processes associated with the design and development of digital learning experiences with similar autopoietic properties found within living systems, in particular adaptability and self-organisation. The underpinning rationale is that, by conferring such properties to digital learning experiences, intelligent digital interventions responding to unpredictable and ever-changing socio-cultural conditions can be created, promoting meaningful learning over-time. Such an epistemological view of digital learning aims to ultimately promote a more efficient type of design and development of digital learning experiences in education.

Introduction: Autopoiesis in digital learning

Digital technology has proven to enhance learning outcomes across educational sectors and contexts, providing great potential for achieving societal transformation (Cook & Santos, 2016; Pachler, Benchair & Cook, 2010). Current smart mobile technology allows learning to occur practically anywhere in collaboration with anyone (Cochrane, 2011); promote innovative (Parsons, 2013), inclusive (Traxler, 2010) and transformative (Lindsay, 2015) types of learning, thereby challenging traditional pedagogical approaches (Merchant, 2012). Mobile learning can fit individual characteristics and needs (Aguayo, 2016) through self-driven learning (Hase and Kenyon, 2013) while addressing local societal challenges (Aguayo and Eames, 2017). Yet critical challenges remain, notably: minimising the decay of digital interventions over time; and achieving widespread learning outcomes in diverse and multicultural audiences (Dunn and Marinetti, 2008; Hennessy, 2019).

One ambitious solution may lie in the theoretical concept of autopoiesis coming from systems biology. Autopoiesis, literally meaning *self-making*, defines living organisms as

self-organising units, capable of adapting to unpredictable changes in their environments while maintaining internal coherence over time (Maturana and Varela, 1980). The focus here is on organisational processes rather than structural components. Since its introduction more than four decades ago, the fundamental ideas of autopoiesis have been transferred and explored across disciplines, including psychology, creative arts, economy, and sociology (Hallowell, 2009; Razeto-Barry, 2014), yet they still remain to be fully explored and applied in the field of technology-enhanced learning.

The Santiago School of Cognition, founded on the concept of autopoiesis, considers the adaptive capacity of living systems towards their environment as an intelligent cognitive process (Maturana and Varela, 1980). But most importantly, it establishes that human experience and cognition are unique to every individual and context (Thompson, 2007). This can have profound epistemological consequences when designing digital technology in education, as the dominant 'one solution fits all' paradigm becomes invalid. On the contrary, digital technology and their associated educational processes on learners ought to provide as many intelligent solutions as individuals and contexts there are over-time, via adaptability and self-organisation (Aguayo, 2018). How can this be achieved? The solution may lie in replicating autopoietical principles and processes during the design and development of digital learning experiences (Aguayo, Veloz and Razeto-Barry, 2019). Yet the focus here is on the paradigm shift when it comes to designing digital learning experiences for a complex audience residing in an ever-changing complex environment.

Theoretical implications in education

From the perspective of the Santiago school, embedding digital learning systems with properties found in living systems could, in theory, create 'intelligent' educational interventions capable of promoting and facilitating the emergence of learning while responding to ongoing socio-cultural changes. This could reduce the amount of time and resources required to maintain digital learning interventions over time, as the current trend is to develop one-off events, many times without long-term considerations or planning. In theory, developing digital learning experiences containing autopoietic properties would contribute to the efficient design and use of digital technology resources and products in

education, which in turn can contribute to overcoming current societal challenges.

Another theoretical premise derived from the Santiago school is that all types of experiences are unique and belonging to the individual and to the moment and context. Therefore, as users interact with intelligent and adaptable digital learning systems (through their ‘user interface’ UI), the coming together of such interaction between two learning actors (Sumara and Davis, 1997), or ‘structural coupling’ (Jorg, 2000; Maturana and Varela, 1980), produces the emergence of a unique ‘user experience’ (UX as known in digital design) bodily lived and experienced by the user; and from where the digital system could cognate to re-adapt and self-organise its structure and functioning to ongoingly facilitate such interaction over-time. The key point to make here is that, in consequence, such a view of UX in digital learning design implies that only user interfaces can be designed and created, with user experiences occurring as an unpredictable emergence of the interaction between users and UI. Yet the current dominating practice in digital learning design is that both UI/UX can be targeted and designed for, with the ‘one solution fits all’ paradigm still dominating. Such paradigm is invalid from the perspective of the Santiago school of cognition, as user experiences are unpredictable, and unique to the individual, the learning situation, and the shared coming together between the digital UI and the user, and therefore cannot be designed for, but facilitated and nurtured. This has profound implications to the current practice of digital learning design in education.

The underlying theoretical and conceptual rationale for the inclusion of autopoietic principles in digital learning design, development and implementation can ultimately contribute to current societal challenges through promoting and facilitating meaningful educational outcomes and experiences over-time. But more importantly, this epistemological view of technology-facilitated user experience invites us to reconsider and reconceptualise the role of digital learning systems and tools in educational practice.

References

- Aguayo, C. (2018). Exploring autopoietic principles in technology-enhanced learning. In *paper presented at the Inaugural Scholarship of Technology-Enhance Learning 2018 Symposium*, 15-16 February, AUT South Campus, Auckland, New Zealand: SoTEL. Accessible at: https://www.researchgate.net/publication/324329272_Exploring_autopoietic_principles_in_technology-enhanced_learning
- Aguayo, C. (2016). Activity theory and community education for sustainability: When systems meet reality. In D. Gederá & J. Williams (Eds.), *Activity Theory in Education: Research and practice* (pp. 139–151). Rotterdam: Sense.
- Aguayo, C., Veloz, T., & Razeto-Barry, P. (2019). Autopoietic principles and processes in the development of digital affordances in education. *In preparation*. Preprint accessible at: https://www.researchgate.net/publication/331192949_Autopoietic_principles_and_processes_in_the_development_of_digital_affordances_in_education
- Cochrane, T. (2011). mLearning: Why? What? Where? How. In *Proceedings of the 28th ASCILITE Conference, ASCILITE*(pp. 250-262).
- Cook, J., & Santos, P. (2016). Three phases of mobile learning state of the art and case of mobile help seeking tool for the health care sector. In D. Churchill, J. Lu, T. K. F. Chiu, & B. Fox (Eds.), *Mobile learning design* (pp. 315-333). Singapore: Springer.
- Dunn, P., & Marinetti, A. (2008). Beyond localization: Effective learning strategies for cross-cultural e-learning. In H. Rahman (Ed.), *Developing successful ICT strategies: competitive advantages in a global knowledge-driven society* (pp. 155-164). London: Information Science Reference.
- Hallowell, R. (2009). Humberto Maturana and Francisco Varela’s Contribution to Media Ecology: Autopoiesis, The Santiago School of Cognition, and Enactive Cognitive Science. In P. A. Soukup (Ed.), *Proceedings of the Media Ecology Association* (pp. 143–158). Saint Louis, MO: Media Ecology Association. Retrieved from: <http://www.mediaecology.org/2009/09/24/hallowell-r-humberto-maturana-and-francisco-varela-s-contribution-to-media-ecology-autopoiesis-the-santiago-school-of-cognition-and-enactive-cognitive-science/>
- Hennessy, S., Mavrikis, M., Girvan, C., Price, S., & Winters, N. (2019). BJET Editorial for the 50th Anniversary Volume in 2019: Looking back, reaching forward. *British Journal of Educational Technology*, 50(1), 5-11. <https://doi.org/10.1111/bjet.12731>
- Jorg, T. (2000). About the unexpected: complexity of learning based on reciprocity and human agency. *Chaos and complexity theory: Special interest newsletter*. Retrieved 8 June, 2009, from <http://www.udel.edu/aeracc/library/Fall00.htm>
- Lindsay, Lucie. (2015). Transformation of teacher practice using mobile technology with one-to-one classes: M-learning pedagogical approaches. *British Journal of Educational Technology*, n/a-n/a. doi: 10.1111/bjet.12265
- Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and cognition. The realization of the living*. Dordrecht, Boston, London: D. Reidel Publishing Company.
- Merchant, G. (2012). Mobile practices in everyday life: Popular digital technologies and schooling revisited. *British Journal of Educational Technology*, 43(5), 770–782. doi:10.1111/j.1467-8535.2012.01352.x
- Pachler, N., Bachmair, B., & Cook, J. (2010). *Mobile Learning: Structures, agency, practices*. Boston, MA: Springer US. Retrieved from <http://link.springer.com/10.1007/978-1-4419-0585-7>
- Parsons, David (Ed.). (2013). *Innovations in mobile educational technologies and applications*. Hershey, PA: IGI Global.
- Razeto-Barry, P. (2012). Autopoiesis 40 years Later. A Review and a Reformulation. *Origins of Life and Evolution of Biospheres*, 42(6), 543–567. doi:10.1007/s11084-012-9297-y
- Sumara, D., & Davis, B. (1997). Enactivist theory and community learning: Toward a complexified understanding of action research. *Educational Action Research*, 5(3), 403-422.
- Thompson, E. (2007). *Mind in life: Biology, phenomenology, and the sciences of mind*. Cambridge, MA: Belknap/Harvard Univ. Press.
- Traxler, John. (2010). Will student devices deliver innovation, inclusion, and transformation? *Journal of the Research Center for Educational Technology (RCET)*, 6(1), 3-15.

How to reduce a genome?

ALife as a tool to teach the scientific method to school pupils

Quentin Carde¹, Marco Foley¹, Carole Knibbe^{1,2}, David P. Parsons¹,
Jonathan Rouzaud-Cornabas^{1,3} and Guillaume Beslon^{1,3}

¹Inria Beagle Team, F-69603, France

²Univ. Lyon, CarMeN Lab., INSERM, INRA, INSA-Lyon, Univ. Claude Bernard Lyon 1, F-69621, Villeurbanne, France

³Univ. Lyon, LIRIS Lab., INSA-Lyon, CNRS, UMR5205, F-69621, France
guillaume.beslon@inria.fr

Abstract

When Artificial Life approaches are used with school pupils, it is generally to help them learn about the dynamics of living systems and/or their evolution. Here, we propose to use it to teach the scientific and experimental method, rather than biology. We experimented this alternative pedagogical usage during the 5 days internship of a young schoolboy – Quentin – with astonishing results. Indeed, not only Quentin easily grasped the principles of science and experiments but meanwhile he also collected very interesting results that shed a new light on the evolution of genome size and, more precisely, on genome streamlining. This article summarizes this success story and analyzes its results on both educational and scientific perspectives.

Introduction

In France, the school program for teenagers aged 14 or 15 includes a 5-day internship in a professional environment. The goals of this internship are (i.) to discover the economic and professional world, (ii.) to have the pupils face the concrete realities of employment and (iii.) to help them build their professional project. Pupils are often welcomed in a relative's company. Children of researchers are no exception and they are often welcomed in a laboratory of their parents' university. During these internships in laboratories, it is generally considered that the actual scientific work is out of reach for the pupils, either because they don't have the necessary background or because the internship is not long enough. Hence the pupils generally visit various teams, discuss with the researchers and with the technical and administrative staff without discovering the reality of research.

In November 2018, one of us was contacted by the father of a young pupil, Quentin, who wished to discover the job of researcher while his family had no background and no contact in this domain. We agreed to welcome Quentin in the team and proposed that it would be a real professional internship, *i.e.*, that Quentin would conduct his own, real, research project during his stay. We proposed to Quentin to discover the reality of the research work, from the statement of a scientific questioning to the analysis of experimental results. On our side, the idea was that artificial life could make

it possible to carry out a research project, even in a the very limited time-frame of five days, with little prior knowledge and no previous experimental practice.

Quentin finally completed his internship in the Beagle team from Monday, January 28th to Friday, February 1st, 2019 under the direct supervision of GB. This article presents the results of this internship with a double objective. First, it shows how artificial life can be used to train young students with method and scientific rigor. Second, it presents Quentin's results, which are very real and worth sharing with the community. The article is structured chronologically, each section corresponding to a day of internship and to a stage of the research project. These five chronological sections are followed by two separate discussions. The first one deals with teaching the scientific process by means of artificial life; the second discusses the scientific results obtained on the causes of genome streamlining as it is observed in several species of bacteria. Finally, a Material and Methods section presents the tools used during the internship. All along this article we will make an extensive use of footnotes, to discuss technical points that either have not been taught to Quentin (because we considered they were too difficult) but that are important for the reader, or experimental results that have been recomputed after the internship to improve confidence¹.

Monday: Science always starts with a question

The scientific method is known to start with a question, generally raised by a striking observation. Hence, experiencing the scientific method requires an observation, simple enough to be understandable by a naive person but also open enough to raise an interesting question. In the context of Quentin's internship, we chose to address the question of genome streamlining and to begin with the diver-

¹All the simulations and statistical analyses were conducted anew by GB, MF, JRC and CK after the internship. In particular, we used a new Wild-Type – see methods – because the one used by Quentin was evolved in two steps (10⁷ generations in a population of 1024 individuals followed by 10⁶ generations in a population of 100 individuals), possibly biasing the results. We wanted to exclude this possibility before publication of the results.

sity of genome sizes and structures in the bacterial kingdom.

Question: Difference in genome size across bacterial species. More than ten years ago, Giovannoni et al. (2005) published a graph comparing the sizes and structures of a large variety of bacterial genomes. Figure 1 shows a similar graph, as we explained it to Quentin.

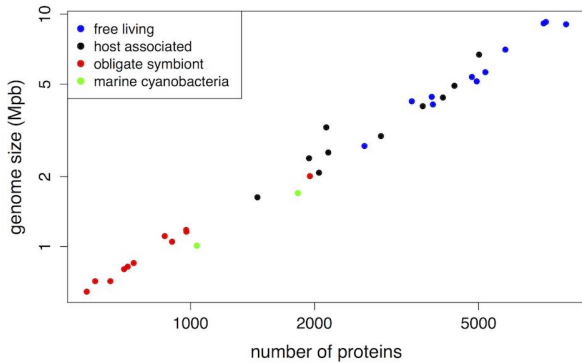


Figure 1: Genome size vs. number of protein coding genes for free living bacteria (blue), host-associated bacteria (black), obligate symbionts (red) and marine cyanobacteria living in very large populations (green). Data from NCBI.

The red dots on Figure 1 represent obligate symbionts. These bacteria have experienced a severe genome streamlining following their obligate association with a host (Wernegreen, 2002), raising the question of the causes of this streamlining. The genomes of free-living marine cyanobacteria show a similar pattern (Figure 1, green dots). Taking into consideration the profound difference in lifestyle, this similarity is quite striking.

A very popular theory to explain the large variety of genome sizes and structures has been proposed by Lynch and Conery (2003). It states that one of the main determinants of genome size is the effective population size N_e . As it is well known in population genetics, N_e drives selection efficiency. Hence, in very large populations, genomes are under a strong selective pressure, preventing them from accumulating slightly deleterious sequences. On the opposite, small populations cannot avoid the proliferation of such elements, hence their large genome size. One of the strengths of this theory is its very elegant statement: by linking genome architecture to a single parameters (N_e) it predicts a continuum of genome size and content very similar to what is observed on Figure 1. However, obligate symbionts do not fit easily with this theory. Indeed, because of their obligate status, they necessarily live in small subpopulations (each within a specific host individual). Starting from the observation that genome streamlining has occurred both in small and large population sizes (Batut et al., 2014), we proposed

that Quentin could address the following question during his internship:

Starting from a wild-type genome, can different processes lead to genome streamlining and if so, is it possible to distinguish between them by observing the resulting genomes?

Hypothesis: Both large population size and high mutation rates can streamline genomes

Once the question has been identified, science proceeds through experiments. However, experiments cannot be directly inferred from the question; we first have to propose *hypotheses* and then design the experiments to test these hypotheses. Many different hypotheses have been proposed to explain the striking genome streamlining in bacteria (reviewed in (Batut et al., 2014)). Here we will focus on two mechanisms that have been suggested to lead to genome streamlining: population size and mutation rate. Indeed, both have been independently suggested to impact genome size (Lynch and Conery, 2003; Lynch, 2006; Knibbe et al., 2007) but their respective effects have never been assessed experimentally. Moreover, both mechanisms have been proposed to impact the genetic structure differently: while population size has been proposed to act on non-coding sequences (because non-coding sequences are supposed to have slightly deleterious effects (Lynch and Conery, 2003)), mutation rates have been proposed to act on the whole genome, including coding and non-coding sequences (Knibbe et al., 2007). We thus proposed that Quentin test the following hypothesis:

Genome streamlining can be caused by changes in population sizes and/or by changes in mutation rates. These two mechanisms are likely to have different effects on coding and non-coding sequences.

Experimental design Being for experimental reasons or due to the limited time of the internship, it was not possible to perform *in vivo* experiments to test the aforementioned hypothesis. Now, provided that the experiments are well designed, it is possible to turn to artificial life and propose designs that enable (i.) to really perform scientific experiments (though *in silico*) (ii.) to get sound results after only a few hours of computation. We hence used the Aevol simulation platform (Knibbe et al., 2007; Batut et al., 2013; Liard et al., 2018) which has been specifically designed to study the evolution of genome architecture and complexity (see Methods).

Since artificial life enables to strictly follow the scientific method while minimizing the experimental and technical issues (e.g. here, how to modify the mutation rate of an organism?), we were able to teach Quentin the basis of the experimental method:

- Modify only one factor at a time,
- Make replicates to get statistical accuracy,

- Compare the results with a control condition in which no factor has been changed,
- Record everything in your lab notebook².

Importantly, we also discussed the issue of experimental costs which, while often neglected in teaching, strongly constrains the experiments in practice. Here, the experimental costs were exemplified by the available computational power Quentin had at his disposal during his internship. The experimental design phase was thus the occasion to present and discuss the actual scientific process and the importance of its different phases. In particular, we insisted on the fact that the experimental results must be gathered early enough such that enough time will remain to analyze them.

With all these elements in mind, Quentin designed two experiments, one to test the effect of mutation rate and one to test the effect of population size. Following the “*in silico* experimental evolution” strategy proposed by Batut et al. (2013), evolutionary runs started from a pre-evolved clone (the “Wild-Type”). Here, the same Wild-Type genome was used to seed all evolutionary runs. To test the effect of mutation rate, three series of evolutionary runs were performed: a series of `Control` runs were performed with the same parameters as those used to produce the Wild-Type, a series of `Mu+` runs were the mutation rate was increased, and a series of `Mu-` runs were the mutation rate was increased. For the population size experiment, the same control runs were used as for the mutation rate experiment, and a series of `N+` (resp. `N-`) runs were performed with increased (resp. decreased) population size. All simulations lasted 100,000 generations. Table 1 summarizes the five tested conditions. While designing the experiments, we discussed resource allocation. Here the problem was to estimate computation time to establish the number of repeats we were able to compute in a reasonable time. We initially chose to compute five repeats for each condition³. Finally, since in *Aevol* the computation time mainly depends on the population size, it was decided to allow more computational resources to the `N+` condition.

Tuesday: Preliminary results

The second day of the internship was almost entirely devoted to technical issues regarding *Aevol* output files, their location on the disk, how to collect them and the different tools available to analyze them; including the reconstruction of lineages (`aevol_misc_lineage`), the computation of lineages statistics (`aevol_misc_ancestors_stats`) and the visualization tools (`aevol_misc_view`). For plotting and data analysis, we decided to use `gnuplot` and `LibreOffice/Calc` as they are user-friendly.

²This was actually the first point we explained to Quentin at the beginning of the internship: we gave him a “notebook” and urged him to write down everything during his internship, including observations, hypotheses, experiments, results or simply ideas.

³All data presented in this paper have been computed with 10 repeats.

Exp. name	Mutation rates	Rearrangement rates	Pop. size	Nb cores
<code>Control</code>	1×10^{-7}	1×10^{-6}	100	1
<code>N+</code>	1×10^{-7}	1×10^{-6}	400	4
<code>N-</code>	1×10^{-7}	1×10^{-6}	25	1
<code>Mu+</code>	4×10^{-7}	4×10^{-6}	100	1
<code>Mu-</code>	2.5×10^{-8}	2.5×10^{-7}	100	1

Table 1: Experimental design. Mutation and rearrangement rates are given in $\text{events.bp}^{-1}.\text{generation}^{-1}$. Column “Nb Cores” corresponds to the degree of parallelism used to compute each condition.

From the current state of the simulations, we were able to estimate the total computation time of the experiments and to reevaluate the number of repeats we could do during the internship. We hence decided to add two more repeats in order to increase the statistical accuracy of the results⁴

Wednesday: Analyzing experimental results

At the beginning of the third day of the internship, all computations were finished. We thus entered into a new phase of the scientific method: results analysis. Since Quentin was then autonomous enough with the experiments, we asked him to collect the characteristics of the best organism of each population at generation 100,000 for the five experimental conditions and to compute their mean values. Table 2 shows the corresponding results.

Exp name	Fitness (mean)	Genome length (mean)	Coding length (mean)	Non-coding length (mean)
Wild-Type	0.00632	44,419 bp	12,235 bp	32,184 bp
<code>Control</code>	0.00643	44,044.4 bp	12,229.2 bp	31,815.2 bp
<code>N+</code>	0.00766	37,116.9 bp	12,216.4 bp	24,900.5 bp
<code>N-</code>	0.00292	46,067.6 bp	11,885.3 bp	34,182.3 bp
<code>Mu+</code>	0.00468	33,693.8 bp	12,003.8 bp	21,690.0 bp
<code>Mu-</code>	0.00641	45,226.7 bp	12,166.4 bp	33,060.3 bp

Table 2: Mean characteristics of the best individuals in the populations at generation 100,000 for the five experimental conditions. These values are to be compared with those of the Wild-Type at generation 0 (first row).

Having computed the mean values for these characteristics, we entered a decisive step by asking Quentin the following question: Can the mean values, as shown in Table 2, be used to draw conclusions about the respective effects of population sizes and mutation rates? In Table 2, all mean values are different – actually means are always different – but part of the difference is due to randomness and sampling

⁴This decision was not based on the p-values obtained so far since we did not compute them at this stage. Had we decided to add more runs until the p-values became significant, it would have been a form of p-hacking (Head et al., 2015). Note that the experiments presented here are replications of Quentin’s ones and have been performed with a predefined experimental design (with ten repeats per condition).

fluctuations. To clarify this point, we plotted the evolution of genome size along the line of descent of the best final organism for the 100,000 generations of the experiment. We used this temporal data to explain to Quentin that, especially when dealing with stochastic processes, mean values must be used with care as they don't account for an important element: dispersion. Now, when looking at these graphs, one could have the *impression* that *i*) an increased population size leads to genome streamlining (Figure 2) while a reduced population size tends to cause a slight increase in genome size (Figure 3), and that *ii*) an increased mutation rate leads to genome streamlining while a reduced mutation rate has no effect (figures not shown). Now the decisive question is "is this *true*?", opening a discussion about what does *being true* mean in experimental sciences?

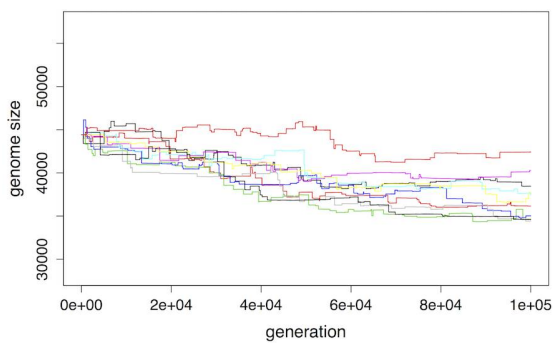


Figure 2: Variation of genome size in the lineage of the clones evolving within an increased population size (N+ clones). Colors indicates the repeats.

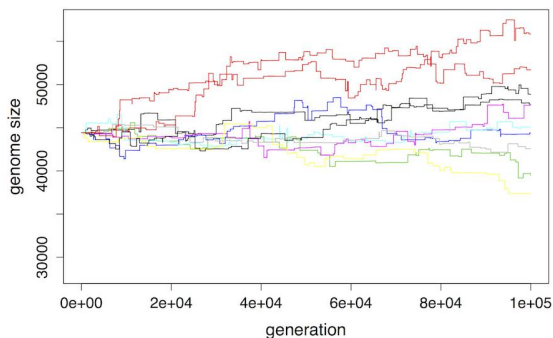


Figure 3: Variation of genome size in the lineage of the clones facing a reduced population size (N- clones).

Thursday: Statistical analysis

Having conducted our experiments, it remained to be tested whether the data *statistically* supported our initial hypotheses, namely that both an increase in population size and an increase in mutation rates are likely to cause a genome reduction.

Given that we had *sampled* a few evolutionary runs among the infinite number of possible runs, we had only *estimates* of the mean evolved genome size in each condition. We explained to Quentin that if we were to replicate the experiment, we would sample different runs and hence obtain different mean estimates for each condition. Perhaps this time the observed mean genome size in the N+ condition would not be smaller than the one in the Control condition. In other words, perhaps the smaller genomes we obtained in the N+ condition was only due to sampling chance! But we observed a change of mean of more than 15%, is sampling chance alone able to produce that? Actually, yes it is, and not necessarily with a low probability. Thus, we need to quantify the change in mean estimate that is expected by sampling chance only. Regarding the question of scientific truth, there is no such thing as experimental truth, only chances of being wrong when drawing conclusions from an experiment...

Given Quentin's age, it was not possible to enter a detailed discussion about random variables, normal distributions, statistical inference, parametric or non-parametric tests, etc. We instead decided to go for a semi-statistical, semi-graphical approach, in three steps:

Step 1: the Central-Limit Theorem We first explained to Quentin the basis of the Central Limit Theorem (CLT⁵). The CLT tells us that if we replicate many times the procedure of sampling n runs and computing the observed average genome size across the n runs of the sample, and if we draw the histogram of the observed sample means, then we will get a bell shape⁶. The width of the bell tells us how much the sample mean is likely to change by sampling chance alone, from sample to sample.

Step 2: Confidence Intervals Statistical theory gives us a formula to estimate the width of the bell and to compute a so-called Confidence Interval (CI) for the mean. This formula depends both on the observed sample dispersion and on the sample size. In our case, with $n = 10$ (i.e. 9 degrees of freedom), the 95% confidence interval is $CI_{95\%} = [\bar{x} \pm 2.262\sqrt{(s^2/n)}]$, with $s^2 = \frac{1}{n-1} \sum (x_i - \bar{x})^2$ and 2.262 coming from Student's t table for 9 degrees of freedom. A 95% CI captures the true mean for 95% of the samples. We helped Quentin build a spreadsheet to compute s^2 and the $CI_{95\%}$ for each of the conditions.

⁵It would have been highly valuable to test it experimentally by e.g. computing more repeats in the Control condition. However, this was impossible in the limited duration of the internship.

⁶Actually, the random variable $\frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$ has a standard normal distribution (i.e. normal with expected value 0 and variance 1), and the random variable $\frac{\bar{X} - \mu}{s/\sqrt{n}}$, where s is the Bessel-corrected sample variance, has a Student's t-distribution with $n - 1$ degrees of freedom.

Step 3: Do confidence intervals overlap? From confidence intervals, we asked Quentin to find a way to identify interesting effects, i.e. those that are most likely not due to sampling chance alone. He decided to check whether confidence intervals overlap or not⁷. Based on this criterion, he concluded that “significant” effects were the following:

1. The genome of Mu+ and N+ are smaller than the genome of the Control (Figure 4).
2. The genomes of the Mu+ and N- contain less coding sequences than the genome of the Control and the genome of the N+ (Figure 5, top panel).
3. The non-coding length of Mu+ and N+ are lower than the non-coding length of the Control (Figure 5, bottom panel).

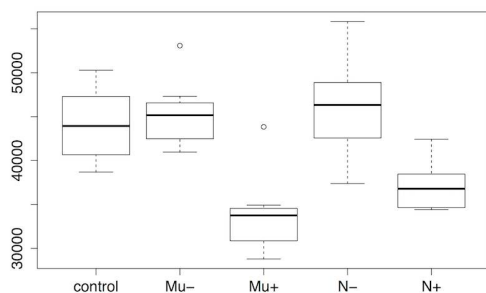


Figure 4: Genome size at generation 100,000 for the 5 conditions.

Friday: Put the results into perspective

One might think that the previous results, obtained after four days, close the scientific process. We chose to show Quentin that this is not the case, on the contrary! Friday was entirely devoted to discussion between Quentin, his supervisor and the rest of the team. He also presented his results to members of the team who had not followed his work. Our goal was to show Quentin that a scientific result must be put in perspective and confronted with the current state of knowledge. We also wanted to show him that communicating results and conclusions is an important part of a scientific work: a scientist must be able to present his results to the community, discuss them and possibly argue against opponents.

Last but not least for a youngster attracted by a scientific career, we discussed the qualities that are necessary to become a researcher, from the obvious (curiosity, rigor, intellectual honesty...) to qualities less often put forward but

⁷Quentin came up with a criterion that is actually used quite often by e.g. biologists, although this approach is not equivalent to performing a statistical test (Krzywinski and Altman, 2013). Here, we performed Kruskal-Wallis tests followed by post-hoc Dunnett tests to compare each condition to the control. We then applied a Bonferroni correction. The p-values of the tests are presented in appendix.

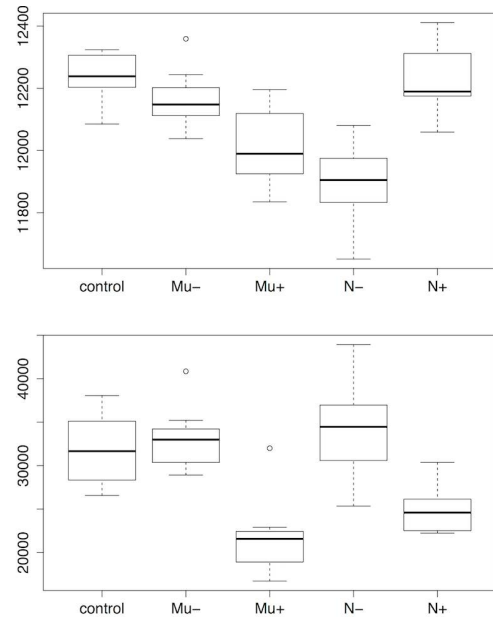


Figure 5: Size of the two main genomic compartments at generation 100,000 for the 5 conditions. Top: coding compartment. Bottom: non-coding compartment.

just as important: passion, pertinacity, scientific (and non-scientific) culture, or – fundamental for a young French boy – the level of English!

Discussion

During this internship – and in this article – we showed two things. First, that ALife could serve as a powerful pedagogical tool to teach the scientific method, including to young and untrained students. Second, that life-traits can strongly influence the length and structure of genomes. Below we separately discuss these two points.

Using ALife to teach the scientific method

Quentin’s internship in the Beagle team illustrated the strength of ALife as a teaching tool. However, contrary to what is generally proposed, here ALife has not been used to teach biology or evolution but rather to teach the scientific method, its main tools and its main issues. We argue that this pedagogical usage of ALife is actually more straightforward than the former usage. Indeed, using simulation to teach biology requires that the student have preliminary understanding of difficult, abstract and actually fuzzy concepts linked to modelling of biological systems (models being, by essence, different from the system they model). This is even more so for artificial life that intends to model life “as it could be” (i.e. in its whole generality) rather than as it is. It also implicitly implies to change the *destination* of the models (from mere research to teaching) and its *user*. Now,

following *e.g.* Minsky (1965) definition of a model (“*To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A*”), it is clear that both the observer *B* and the destination (*i.e.* the “questions that interest him”) of a model are central in the complex relationship that links the model object *A** to the original object *A*. In a word, changing the destination and the user of the model results in such a deep alteration of the *A/A** relationship that it generally implies changing... the model! This is indeed the process the Avida team engaged through the development of Avida-ED (Speth et al., 2009).

These difficulties vanish when ALife is used to teach the basis of the scientific method, be it during an internship or a labwork. Indeed, in this case, the student is engaged in a scientific process: he/she must answer a question about the model behaviour, exactly as the original user of the model would. Hence there is no change in the model destination and the user change is only minor since the student actually plays the role of a scientist. As a matter of fact, during Quentin’s internship, we did not encounter any conceptual issues regarding the differences between the model and the “real” system. This is simply due to the studied object being Aevol and not what it models. The relationship between the model and the real world was indeed discussed during the internship but that was at the very end (on Friday) and it did not need to be accepted *a priori*. When Quentin was asked to summarize what he had learned during his internship, his answer was: “*During my five day work experience with Inria, I have been able to observe and to learn the reality of research and what are the different tasks of this job: to put forward hypotheses, to experiment, to analyze and to publish results. I have learned what are the main qualities of a researcher and what are the studies leading to this job. In only five days, thanks to simulation, I obtained results, I was able to analyze them and then to present them. I noticed that, in teamwork, its very important to have good relations between team members*”. Of course, he has also learned a lot about evolution, genomics and genetics. But this appears to be less important that the insights into the scientific process itself...

Of course, we – Quentin and us – encountered difficulties during the internship. But most of them were technical, not directly related to the use of Artificial Life. Importantly, none of them proved to be crucial and none compromised the learning process. In fact, the main difficulty encountered was the visualization of the raw data and the visualization of the results of the statistical analyzes. Indeed, all the figures presented in this article were made using R but this software was clearly unusable in the context of such a short educational process. Even though we were able to work around the problem, it would clearly have been desirable to have a simple tool allowing Quentin – a naive user on that matter – to manipulate and visualize the data autonomously.

When teaching the scientific method, ALife proved to have valuable advantages. Here, we will focus on the two main ones. First ALife relies on *in silico* experiments, which is twice an advantage as it allows for fast experiments that can also be easily replicated (the replication effort being supported by the computer rather than by the student). In the case of Quentin’s internship, a single computer – though a relatively powerful one – was used, enabling him to conduct 5 series of 7 runs, in little more than 24h. Second, using ALife, one can propose internships or labworks focusing on open scientific questions, for which neither the students nor the mentors have a definitive answer. This creates a strong initial motivation and allows to maintain it all along the process as both the students and the mentors are likely to be surprised – possibly negatively – by the results. In the case of Quentin’s internship, the results shed an interesting light on the process of genome streamlining that deserves to be discussed on its own.

How to reduce a genome?

Genome reduction is common in Nature but its causes are still elusive as biological data suggest that genome reduction could be either neutral or adaptive (Wolf and Koonin, 2013). Our results suggest that at least two distinct mechanisms can lead to genome reduction and that they are distinguishable by their effect on coding sequences. We also show that these mechanisms are triggered by two different causes: increased mutation rates or increased population size.

Interestingly, the effects we observed here fit remarkably well with what is observed in streamlined bacteria. Indeed, obligate symbionts have an elevated mutation rate (Itoh et al., 2002) while marine cyanobacteria live in very large populations (Batut et al., 2014). Both have undergone genome streamlining but the reduction is more pronounced in obligate symbionts (Figure 1), as in our simulations. Moreover, in these two families, the reduction seemed to have impacted differently the different genomic compartments. Marine cyanobacteria have mainly lost non-coding or duplicated elements, as exemplified by *Pelagibacter ubique*, one of the smallest genome of free living bacteria. Its genome is characterized by a very small fraction of non-coding DNA (less than 5%) and the quasi-absence of redundancy in coding sequences while all metabolic pathways are still present (Giovannoni et al., 2005). By contrast, the genome of *Buchnera aphidicola*, an aphid endosymbiont has lost 90% of its genome (compared to *E. coli*, one of its close relatives), lost several metabolic pathways but, strikingly, has conserved 15% of non-coding sequences, a proportion similar to what is observed in *E. coli* (Batut et al., 2013).

The results presented here don’t allow to identify the causal link between mutation rates, population size and genomes size. However, taking advantage of the model characteristics – and of our previous results with Aevol, we can

exclude some mechanisms and put others forward. Typically, many authors suggest that genome size may be governed by mutational biases, selection for optimized physiological traits (cell size, replication time, energetic costs...) or by transposable elements activities. These are all excluded by our simulation parameters or by Aevol itself. Moreover, in Aevol, there is no cost associated to non-coding sequences. Hence, the influence of population size on genome length cannot be linked to the deleterious effect of non-coding sequences as argued by (Lynch and Conery, 2003).

In the absence of direct selective effects or mutational biases, we hypothesize that genome size is driven by indirect selective constraints, namely by selection for robustness (Wilke et al., 2001). Indeed, as already shown by Knibbe et al. (2007), selection for robustness links genome size to mutation rates, the higher the latter, the smaller the former. We hypothesize that the same phenomenon also explains the influence of population size: in large populations, the selection strength is higher, increasing the pressure for robustness, thereby favouring smaller genomes. The striking question is then to explain why both phenomena act similarly on non-coding sequences but differently on coding sequences (for which only an elevated mutation rate induces a reduction). We propose that, exactly like selection for fitness, selection for robustness may act *positively* (selecting more robust clones) or *negatively* (eliminating clones that are not robust enough – aka *purifying* selection). Now, in the case of an increased mutation rate, the error threshold (Eigen and Schuster, 1977) moves down and some individuals may find themselves over this crucial threshold. In this case, selection will purify the population from these individuals, retaining only those that reduced their genome, whatever the elements they have lost (including coding and non-coding sequences). On the opposite, in case of an elevated population size, the error threshold does not move down and individuals are still robust enough to maintain their fitness. However, providing adaptive mutations are rare (which is the case here), this results in a positive selection for robustness. In this case, individuals must retain their fitness (hence their coding sequences) and the only way to increase their robustness is to get rid of non-coding elements. To the best of our knowledge, these two contrasting effects of selection for robustness had never been identified before. Not content with illustrating the interest of using artificial life to teach the scientific method, our results also show that interesting scientific insights can be gathered meanwhile and open the exciting perspectives of characterizing these two effects in our experiments (by *e.g.* measuring robustness levels along the evolutionary path), in different conditions and in different systems.

Material and methods

Simulation platform

All simulations were run using the regular Aevol model, version 5.0, as available on the platform website (www.aevol.fr). Since Aevol has been extensively described elsewhere (Knibbe et al., 2007; Batut et al., 2013) we will not detail it here and focus only on its core principles and on the elements that are specifically of interest for this paper. Figure 6 shows the main components of the model. Aevol simulates a population engaged in a generational process (Fig. 6.A). Each individual is described by a circular double-strand genomic sequence whose structure closely models a bacterial genome (including non-coding sequences, transcription and translation initiation sequences, open-reading frames...) making Aevol an ideally suited platform to study the evolution of genome length and structure. This genome is decoded into a $[0 : 1] \rightarrow [0 : 1]$ mathematical function which proximity with a target function gives the fitness of the individual (Fig. 6.B). Individuals replicate locally (Fig. 6.C) and, importantly for the present study, Aevol implements a large variety of variation operators (Fig. 6.D) including mutations (base switches, small insertions and small deletions) and chromosomal rearrangements (duplications, deletions, translocations and inversions). All variation operators can be tuned independently, making the platform an ideally suited tool to study indirect selection for mutational robustness.

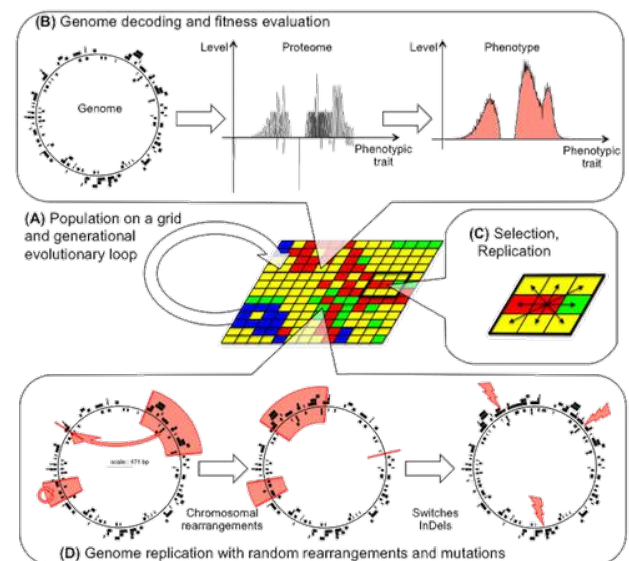


Figure 6: The Aevol model (figure from (Liard et al., 2018)). (A) Population on a grid and evolutionary loop. (B) Overview of the genotype-to-phenotype map. (C) Local selection process with a Moore neighborhood. (D) Variation operators include chromosomal rearrangements and local mutations.

Evolution of the Wild-Type strain

To study genome streamlining, we had to give Quentin an initial organism (the “Wild-Type”) that was evolved prior to the internship. In the experiments presented here the wild-type evolved with a population of 100 individuals, a mutation rate of 10^{-7} events.bp⁻¹.generation⁻¹ for each of the three kinds of mutations and a rearrangement rate of 10^{-6} events.bp⁻¹.generation⁻¹ for each of the four kinds of rearrangements. We used an unusually small population size in order to allow for fast experiments.

In order to ensure that the genome size and structure of the wild-type have reached a steady state, we let it evolve for 10 million (10^7) generations in constant conditions as preliminary results have shown that the genome size did not stabilize before 5.10^6 generations (data not shown). Note that the evolution of genome size and genome structure in the Control conditions (see Table 2) confirmed that the genome was stabilized. We then extracted the genome of the best individual at the last generation. It contains 44,419 bp, 32,184 of which are non-coding and 12,235 coding. It encodes 146 genes transcribed on 108 coding mRNA, approximately half of which being polycistronic. Its fitness is 0.00632. Note that the proportion of non-coding sequences is rather high in this organism, probably because of the small population size.

Experimental design

Starting from the wild-type genome, we used the `aevo.create` tool to initialize a clonal population of wild-types with specific parameters (population size, mutation and rearrangement rates, see Table 1). This procedure allowed us to avoid sampling issues when changing the population size.

All simulations were then performed on an Intel Xeon CPU with 32-cores at 2 Ghz with 32 Go RAM that Quentin had at his entire disposal for the whole duration of the internship. With this configuration, all the computations required approximately 48h.

Appendix: results of the statistical analyses

Results of the Kruskal-Wallis (KW) and post-hoc Dunnett tests for effects of mutation rates and population size on genome size, coding length and non-coding length. Dunnett p-values under 0.017 are in bold face (accounting for $p < 0.05$ with a Bonferroni correction for three responses).

Genome size:

KW on mutation rate: $\chi^2 = 16.694$, $df = 2$, p -value = 0.0002371
Dunnett: Mu- vs. Control: 0.7205; **Mu+ vs. Control: 4.4e-06**
KW on Population size: $\chi^2 = 15.36$, $df = 2$, p -value = 0.000462
Dunnett: N- vs. Control: 0.4661; **N+ vs. Control: 0.0020**

Coding length:

KW on mutation rate: $\chi^2 = 13.388$, $df = 2$, p -value = 0.001238
Dunnett: Mu- vs. Control: 0.2833; **Mu+ vs. Control: 4.9e-05**
KW on population size: $\chi^2 = 19.311$, $df = 2$, p -value = 6.407e-05
Dunnett: **N- vs. Control: 3.5e-07**; N+ vs. Control: 0.9523

Non-coding length:

KW on mutation rate: $\chi^2 = 16.498$, $df = 2$, p -value = 0.0002615
Dunnett: Mu- vs. Control: 0.6955; **Mu+ vs. Control: 6e-06**
KW on population size: $\chi^2 = 15.801$, $df = 2$, p -value = 0.0003705
Dunnett: N- vs. Control: 0.3650; **N+ vs. Control: 0.0022**

References

- Batut, B., Knibbe, C., Marais, G., and Daubin, V. (2014). Reductive genome evolution at both ends of the bacterial population size spectrum. *Nature Reviews Microbiology*, 12(12):841.
- Batut, B., Parsons, D. P., Fischer, S., Beslon, G., and Knibbe, C. (2013). In silico experimental evolution: a tool to test evolutionary scenarios. *BMC bioinformatics*, 14(15):S11.
- Eigen, M. and Schuster, P. (1977). A principle of natural self-organization. *Naturwissenschaften*, 64(11):541–565.
- Giovannoni, S. J., Tripp, H. J., Givan, S., Podar, M., Vergin, K. L., Baptista, D., Bibbs, L., Eads, J., Richardson, T. H., Noordewier, M., et al. (2005). Genome streamlining in a cosmopolitan oceanic bacterium. *science*, 309(5738):1242–1245.
- Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., and Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLOS Biology*, 13(3):1–15.
- Itoh, T., Martin, W., and Nei, M. (2002). Acceleration of genomic evolution caused by enhanced mutation rate in endocellular symbionts. *Proceedings of the National Academy of Sciences*, 99(20):12944–12948.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding dna. *Mol. Biol. Evol.*, 24(10):2344–2353.
- Krzywinski, M. and Altman, N. (2013). Points of significance: error bars. *Nature methods*, 10(10):921.
- Liard, V., Parsons, D., Rouzaud-Cornabas, J., and Beslon, G. (2018). The complexity ratchet: Stronger than selection, weaker than robustness. In *Artificial Life Conference Proceedings*, pages 250–257. MIT Press.
- Lynch, M. (2006). Streamlining and simplification of microbial genome architecture. *Annu. Rev. Microbiol.*, 60:327–349.
- Lynch, M. and Conery, J. S. (2003). The origins of genome complexity. *Science*, 302(5649):1401–1404.
- Minsky, M. (1965). Matter, mind and models. In *Proceedings of IFIP Congress, Spartan Books, Wash. DC*, pages 45–49.
- Speth, E. B., Long, T. M., Pennock, R. T., and Ebert-May, D. (2009). Using avida-ed for teaching and learning about evolution in undergraduate introductory biology courses. *Evolution: Education and Outreach*, 2(3):415.
- Wernegreen, J. J. (2002). Genome evolution in bacterial endosymbionts of insects. *Nature Reviews Genetics*, 3(11):850.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331.
- Wolf, Y. I. and Koonin, E. V. (2013). Genome reduction as the dominant mode of evolution. *Bioessays*, 35(9):829–837.

Gathering of the Hive:

Investigating the clustering behaviour of honeybees through art and swarm robotics

Haakon Haraldsen Roen[§], Vako Vartkes Varankian[§], Stefano Nichele[%] and Kristin Bergaust[§]

[§]OsloMet: Oslo Metropolitan University, Department of Art, Design and Drama

[§]OsloMet: Oslo Metropolitan University, Department of Mechanical, Electronics and Chemical Eng.

[%]OsloMet: Oslo Metropolitan University, Department of Computer Science
s236590@oslomet.no, s305138@oslomet.no, stenic@oslomet.no, krike@oslomet.no

Abstract

In the *Gathering of the Hive* project, the societal and ecological implications, as well as technological possibilities of swarm robotics are explored through artistic methodology applied to Artificial Life. These matters are examined through an algorithm inspired by the clustering behaviour of honeybees applied to a swarm of Thymio robots interacting in a physical, changing environment. This work is a part of the ongoing FELT¹ project (Futures of Living Technologies), which explores artificial life systems through art and technology.

Introduction

A self-organizing clustering behaviour is most commonly found in social bees, and occurs when weather conditions are sub-optimal (Crailsheim, et al. 1999). These clusters are formed through mutual, collaborative, interactive behaviour within groups of bees. A combination of factors play into the movement of the individual bee, but gives a collective result in tightly formed clusters around areas with ideal conditions for the group as a whole, a living behavioural system.

Working through artistic practice gives a flexibility and open-endedness to the collaborative process, opening up to new ideas emerging from unexpected places. With the theoretical entrance point of System Aesthetics, developing art in regards to the way things are done (Burnham, 2015) – be that by human, robot or bee – artistic and technological strategies are used to examine this clustering and reflect around the implications of its technological application. Using this behavioural pattern towards programming a swarm of robots, algorithmically set to exhibit traits of this natural living system, we explore the artistic potential and concerns central to that of Artificial Life (Penny, 2017). By looking at not only the behaviour of bees in-amongst themselves, but towards and in interaction with their environment, ecological perspectives, also come into play. In the condition of the Anthropocene (Wark, 2016), where natural and human forces intertwine – man-made technologies increasingly possess properties of life and the planet is irreversibly altered by the actions of humans – what role can these technological systems

drawing on natural mechanisms play? This is explored through a combination of artificial life, artistic production and research. This work is a part of the ongoing FELT¹ project (Futures of Living Technologies), carried out at the Living Technology Lab (Berg, et. al. 2019; Hansen, et al. 2018) at the Oslo Metropolitan University (Oslo, Norway).

Algorithm

The algorithm taken as an entrance point in this project comes from Schmickl and Hamann (Schmickl and Hamann, 2011) studies of the collective behaviour of honeybees, where they developed an algorithm for clustering behaviour designed for implementation into a robot swarm. Essentially the algorithm, named BeeClust, performs the following steps:

- The honeybees roam around randomly until encountering a wall, or another honeybee.
- When they encounter a wall, they turn around, and continue roaming.
- If they meet another honeybee, they read the sensor measurement and stay idle for a time proportional to the sensor value.

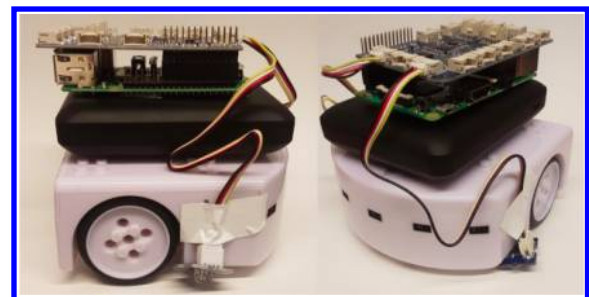


Figure 1: Technological set-up of the thymio robots

This will cause the honeybees to cluster around the focused area, where their sensor values stay high (Schmickl and Hamann, 2011). Being an algorithm designed with swarm robotics in mind – optimal result depends on the collaborative effort of multiple actors – the amount of robots present would become an important point of investigation (Navarro and Matia, 2013).

[1] <https://sites.google.com/view/feltproject>

Implementation

The BeeClust algorithm was developed in Python, stored on a Raspberry Pi 3 Model B. The Raspberry Pi is connected to a Thymio II² robot, and used as a controller for the robot's actions, and to read the sensor values. A power bank is used to supply the robot and the Raspberry Pi with power.

The Thymio II has sensors in the front, five sensors facing forwards and two facing downwards. These infrared sensors can detect nearby objects, and colour range ranging from black to white. These sensors are not precise enough to detect small environmental changes, thus external sensors were implemented, to read the light intensity of the projection. The GrovePi³ is a shield to Raspberry Pi that can easily be connected to sensors provided by the same company. The used sensors can measure light changes in different environmental conditions precisely. The GrovePi reads sensory values and converts the signal to analogue, providing Raspberry Pi the ability to read analogue signals, and enables us to work with intensity, rather than binary on/off values. The robot setup is depicted in Fig. 1.

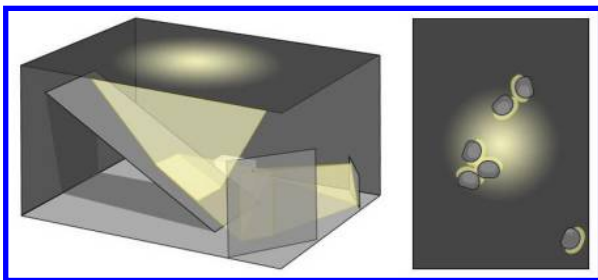


Figure 2: Model of rear-lit set-up (left), and model of bees clustering in relation to light-gradient (right)



Figure 3: Thymios clustering on the projected gradient.

Designing the Environment

When designing the environment for the Thymio robots to roam, the goal was to make a dynamic environment, where the source of optimal 'heat' would be in a changing condition, thus affecting the clustering performance of the bees. The way bees use the sun to triangulate the location of food sources in relation to their hive, was used as inspiration for the design (Evangelista, et al. 2014). Instead of a food source being the site of aggregation, this would be a simulated 'Sun' in the form of a pulsating white-to-black gradient projected onto a semi-transparent surface the robots were to move on. Aided by mirrors built into a box, this rear-projection would make

possible a system for the robots to move, cluster, and re-cluster as the 'Sun' goes its way across the surface. Thus making practical use of the behavioural pattern in the form of a robot artistic performance in a visually perceivable way, also for the spectator. A setup is shown in Fig. 2, and an example of the swarm robotics art performance is shown in Fig. 3. This follows as a continuation of previous work on the project called *Pheromone Performance with Swarm Robotics*. A short introductory video to this can be seen through the following link: http://y2u.be/hA_YsC6mLP0

Conclusion and Further Work

Our work on the implementation and application of the BeeClust algorithm has resulted in a system capable of making use of, and highlight the distinctive pattern of honeybee clustering. Using clearly defined principles, the ongoing systematic process the robots conduct in this artistic performance carry its own possibilities and limitations. By using technology in unexpected ways, such as a light-led robot performance, artistic practice can provide new perspectives towards technological development. One could, for example, imagine similar principles used by robot swarms to cluster around lost people at sea through the lead of body temperature.

The arts opens up to an expanded use of the strategies and mechanisms set forth in this project and in artificial life in general, giving room to imagine robots and artificial living systems 'programmed' towards other parts of society. In future work this will be further explored through art exhibitions and workshops where an audience is invited to learn about and explore principles and possible execution of artificial life – Further reflecting over its biological and ecological roots and technological potential, in the current state where these seem to become increasingly entangled.

References

- Berg, J., Berggren, N., Borgeteien, S., Jahren, C., Sajid, A., Nichele, S. (2019). Evolved art with transparent, overlapping, and geometric shapes. *arXiv preprint arXiv:1904.06110*
- Burnham, J. (2015). Systems Aesthetics. In E. A. Shanken (Ed.), *SYSTEMS*. Cambridge, Massachusetts: MIT Press
- Crailsheim, K., Eggenreich, U., Ressi, R., & Szolderits, M. (1999). Temperature preference of honeybee drones (Hymenoptera: Apidae). *Entomologia Generalis*, 24(1), 37-47.
- Evangelista, C., Kraft, P., Dacke, M., Labhart, T., & Srinivasan, M. V. (2014). Honeybee navigation: critically examining the role of the polarization compass. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 369(1636).
- Hansen, E., Nichele, S., Yazidi, A., Haugerud, H., Mofrad, A., Alcocer, A. (2018). Achieving Connectivity Between Wide Areas Through Self-Organising Robot Swarm Using Embodied Evolution. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018.
- Navarro I. & Matía F. (2013) An Introduction to Swarm Robotics. In *ISRN Robotics*, vol. 2013.
- Penny, S. (2017): *Making Sense: Cognition, Computing Art, and Embodiment*. Cambridge, Massachusetts: MIT Press
- Schmickl, T. & Hamann, H. (2011). BEECLUST: A Swarm Algorithm derived from Honeybees. *Bio-inspired computing and communication networks (2011)*: 95-137.
- Wark, M. (2016). *Molecular Red: Theory of the Anthropocene*. London: Verso

[2] <https://www.thymio.org/en:thymio>

[3] <https://www.dexterindustries.com/grovepi/>

Data Standards for Artificial Life Software

Alexander Lalejini^{1,2,3,*}, Emily Dolson^{1,2,3,*}, Clifford Bohm^{1,2,4,*}, Austin J. Ferguson^{1,2,3},
David P. Parsons⁵, Penelope Faulkner Rainford⁶, Paul Richmond⁷, Charles Ofria^{1,2,3,*}

*Organizing author

¹BEACON Center for the Study of Evolution in Action

²Ecology, Evolutionary Biology, and Behavior Program, Michigan State University

³Department of Computer Science, Michigan State University

⁴Department of Integrative Biology, Michigan State University

⁵Inria Beagle Team, F-69603, France

⁶Department of Mathematical Sciences, Durham University

⁷Department of Computer Science, University of Sheffield

lalejini@msu.edu

Abstract

As the field of Artificial Life advances and grows, we find ourselves in the midst of an increasingly complex ecosystem of software systems. Each system is developed to address particular research objectives, all unified under the common goal of understanding life. Such an ambitious endeavor begets a variety of algorithmic challenges. Many projects have solved some of these problems for individual systems, but these solutions are rarely portable and often must be re-engineered across systems. Here, we propose a community-driven process of developing standards for representing commonly used types of data across our field. These standards will improve software re-use across research groups and allow for easier comparisons of results generated with different artificial life systems. We began the process of developing data standards with two discussion-driven workshops (one at the 2018 Conference for Artificial Life and the other at the 2018 Congress for the BEACON Center for the Study of Evolution in Action). At each of these workshops, we discussed the vision for Artificial Life data standards, proposed and refined a standard for phylogeny (ancestry tree) data, and solicited feedback from attendees. In addition to proposing a general vision and framework for Artificial Life data standards, we release and discuss version 1.0.0 of the standards. This release includes the phylogeny data standard developed at these workshops and several software resources under development to support our proposed phylogeny standards framework.

Introduction

Artificial Life (ALife) research is becoming more complex as the field advances and as computational power increases. Further, more recent initiatives have broadened the scope of the field to intersect topics such as society and education, attracting new and interesting perspectives to the community. We find ourselves in the midst of an increasingly complex ecosystem of ALife software systems (Taylor et al., 2016), including research platforms, metrics, data visualizations, *et cetera*. Each system is developed to address particular research objectives, all unified under the common goal of

understanding life; such a monumental goal begets a number of algorithmic challenges (*e.g.*, tracking a single gene through a genetic lineage, measuring the open-endedness of a system, identifying and characterizing complex interactions among individuals in a population). Many projects have solved some of these problems in individual systems, but these solutions are rarely portable and often must be re-engineered across systems.

Many other communities have developed and leveraged data standards to dramatically improve their software ecosystems. Data standards are specifications for organizing, annotating, and recording commonly-collected information. That is, what specific values should we keep, what descriptors (properties) should we use to specify them, and in what format should they be stored?

We propose a community-driven process of developing such standards for Artificial Life in an effort to improve software re-use and allow for easier comparisons of data generated with different artificial life systems. Standards allow tools to be developed that can immediately be applied to data produced by unrelated systems, eliminating the need for these tools to be re-written by each research group. In addition to saving time, expanding the user base for tools increases their reliability by making it harder for bugs to go undetected. Moreover, creating a collaborative software ecosystem will facilitate communication and cooperation among research groups by making it easier to compare results across different systems using the same analysis tools. Further, standards increase the incentive to develop tools that solve elusive community-wide challenges, as you will be able to immediately apply them to a broad cross-section of available systems and data; likewise, many fellow researchers will be able to make easy use of your tools.

We began the process of developing data standards with two discussion-driven workshops at the 2018 Conference for Artificial Life and the 2018 Congress for the BEACON Center for the Study of Evolution in Action. At both work-

shops, we discussed the vision for ALife data standards, proposed and refined a standard for phylogenies (that is, a standard for describing parent-offspring relationships over time), and solicited feedback from attendees (Lalejini and Dolson, 2019); in conjunction, we developed software tools to leverage these proposed standards. This paper is a continuation of these efforts. Here, we provide examples of how data standardization has benefited other scientific communities. We summarize our vision for artificial life data standards, and propose a framework for ALife data standards. By way of example, we present the phylogeny standard discussed in both 2018 workshops; additionally, we identify several existing software resources under development to support our proposed phylogeny standards framework: developer utilities, data converters, and end-user tools. We conclude with a discussion of future directions, including possible concepts for future standardization.

The Benefits of Data Standardization: Examples From Other Communities

As scientific communities extend their reach, data standards provide a mechanism to unify software development and provide a better user experience. Data standards afford developers a reduced barrier to entry, the ability to more easily communicate across disciplines, and a broader impact from their software efforts. As such, users experience a more unified software ecosystem where they can use the same analysis and visualization tools across research platforms. Our vision for Artificial Life data standards is inspired by these other successful efforts.

Data standards have been successfully adopted throughout history. For example, the metric system revolutionized how weights and other measures are used throughout science, and failures to keep to this standard have proved catastrophic (Board, 1999). The ASCII standard shaped how modern computers manage text, allowing developers to write versatile tools to manipulate human-readable files. In modern biology, both computational neuroscience (Gleeson et al., 2010; Richmond et al., 2014) and systems biology (Hucka et al., 2003) have adopted successful data standards. In both cases, these standards were driven by an open community approach, resulting in improved model exchange and design as well as the development of compliant simulators. Furthermore, digital access to global data is integral to biodiversity research. The Darwin Core project data standards (Wieczorek et al., 2012) define relevant properties for a range of scientific entities (*e.g.*, taxa, occurrences, fossil specimens); this standardization has eased communication of and collaboration using biodiversity data, allowing the community to homogenize biodiversity record structure across multiple repositories (Parr et al., 2012). The bioinformatics community is moving toward widely-adopted data standards (Zhang et al., 2011), ushering the development of broadly used (Wren, 2016) databases and software tools.

Highly relevant to the field of Artificial Life, the Robot Operating System (ROS) is a popular, open-source software development framework that defines communication and data standards. The ROS standards have facilitated massive community software development, sharing, and reuse (Quigley et al., 2009). By defining common standards for software, ROS unifies disparate sub-communities (ranging from academic researchers to industrial engineers to hobbyists), creating the opportunity for robotics collaborations among people who would have never even communicated otherwise. In 2012, the ROS community began organizing annual ROScon events, an international conference where ROS software developers meet and present recent software applications, ideas, and tools. All of this community buy-in and support for software developers eases onboarding for new researchers and lowers the barrier to making meaningful software contributions for the community.

Our Vision for Artificial Life Data Standards

Data standards specify how data are described (ontologies) and recorded (formats). Because the types of data used in Artificial Life research are many and varied (from experiment-to-experiment and system-to-system), any useful set of standards for our community will need: 1) a minimalist and inclusive core shared by all, 2) a flexible mechanism for extensions to encompass the idiosyncrasies of individual systems, and 3) enough descriptive power to allow for tools that will be broadly useful.

Because ALife systems and experiments are diverse, we must ensure that standards for *describing* different types of data are flexible. We envision that each data standard should minimize the number of properties required to specify the concept of interest; we should avoid incorporating extraneous or restrictive assumptions into the standards, ensuring the core of each standard remains inclusive. For example, to meaningfully describe a lineage, we require, at minimum, information about parent-offspring relationships. While broadly applicable, such a data standard would sacrifice utility if it disallowed extra information of potential interest such as organism characteristics or mutational changes that occurred along the lineage. Thus, in addition to a set of mandatory properties, each data type will also standardize a set of optional (or “conventional”) properties. Standard-compliant data are not required to report optional information; however, *if* standardized optional properties are included, they must use the specified labels and formatting. Each standard-compliant software tool must document which optional properties it accepts and/or requires, and be able to ignore those that it does not use.

The diversity of systems and experimental settings not only requires flexibility on *what* data may be recorded but also on *how* it should be recorded (*i.e.*, the underlying file formats). Depending on the data itself and on what one wants to do with it, the choice of file format could range

from a verbose format such as JSON or XML to a compressed binary format. Our vision is for the standards to support multiple, interchangeable file formats (*e.g.*, XML, JSON, CSV, binary, *etc.*) to ensure that data recording is maximally flexible. Any of these formats could be used to store data and provide input to standard-compliant software tools; this flexibility, however, demands that we provide detailed guidelines for conversions between storage formats.

Proposed Standards Framework

Our proposed standards framework specifies: 1) the terminology for referring to specific concepts (ontology), 2) a structure for describing data, 3) rules for formatting data, and 4) the process for creating and modifying the standards. Note that our proposed framework does not constrain methods for analyzing or working with data. However, agreeing on standardized ways of formatting and describing data will ease the development of analysis tools and visualizations.

Ontology

A critical component of any data standard is a set of agreed-upon terminology for describing data, with clear rules about how information fits together. In information science, such a framework is called an *ontology* (Smith and Welty, 2001). For consistency, we will use the same terms that are used in other ontology development research.

Each individual standard in our framework specifies a way of describing and storing a particular instance of a *concept* (*e.g.*, the “phylogeny standard” describes how to store the concept of a phylogeny). Each instance of a concept can be described with a single, arbitrarily large data table where rows are *entities* (*e.g.*, individual taxa in a phylogeny) and columns specify *properties* (*e.g.*, ancestor IDs or trait values) of each entity; further, an individual data table may contain entities of only a single type or category.

Describing Data

Each standardized data type (concept) has three categories of properties (*i.e.*, data fields or attributes): *required* properties, *conventional* (or optional) properties, and *extra* (or additional) properties. To qualify as standard-compliant, a data file must abide by *all* required properties. Required properties are what the community determines to be the minimal set of properties needed to meaningfully specify the concept of interest. Properties should only be given required status if they are fundamental to the concept being represented, such as parent-offspring relationships in a phylogeny.

In addition to required properties, each concept will standardize a set of conventional properties. Conventional properties are used to describe pieces of data that are often important, but are not fundamental to the concept being recorded. As such, standard-compliant systems are not required to output conventional properties (and, indeed, these properties may not even be meaningful in all systems or setups). If you

do choose to output conventional properties and label them according to the standard, these properties may be leveraged by standard-compliant tools. For example, mutation counts are not required to record a phylogeny, but if they are included, analysis tools can produce more informative visualizations indicating amount of change over time.

Extra properties include any data not otherwise specified; these are system-specific or experiment-specific properties that further describe the concept. Allowing for arbitrary extra properties ensures the standard is inclusive and easy to use. Software tools should document any extra properties they can make use of and their meaning. In the event that an extra property is used by multiple software systems, it may be appropriate to formalize it as a conventional property. Additionally, many software tools may be able to use arbitrary properties by name. For example, a user may be able to choose an any property they want to color-code a phylogeny.

Property names must be consistent across file formats. As of version 1.0.0, all property names are in snake case: fully lowercase (*e.g.*, ‘id’ instead of ‘ID’ or ‘Id’) and underscore-separated as appropriate (*e.g.*, ‘ancestor_list’). When deciding on required or conventional property names, we will err on the side of being descriptive to ensure that files remain intuitive. We encourage extra properties to be similarly descriptive to simplify data sharing, limit name collisions, and facilitate future conversion to conventional property status.

Naming Modifiers Workshop participants suggested we specify conventions for naming common *types* of properties (*e.g.*, lists, averages, variances, *etc.*). For example, the property for identifying the set of offspring produced by a particular organism might be called ‘offspring_list’ instead of ‘offspring’ to indicate that the property refers to a list. We envision the set of these conventions (*i.e.*, naming modifiers) to grow as new tools are developed and as the standards grow to encompass more concepts. Table 1 provides the set of naming modifiers in version 1.0.0 of the standard. Naming modifiers should be applied to the end of the property name, connected via an underscore (*e.g.*, ‘offspring_list’). These conventions allow tools to be more flexible when loading and processing standardized data by inferring data types for properties and identifying property relationships from a common prefix. For example, if a tool sees both fitness_ave and fitness_std, it may reasonably assume that these refer to the average and standard deviation of the same distribution.

Reserved and Default Values For certain properties, it is valuable to reserve values to have special meanings. For example, what value should ancestor_list use to indicate that an organism in a phylogeny was created randomly and therefore has no ancestors? This parameter could be left empty, but how would we be able to differentiate this organism from one that migrated from another population or from an or-

Table 1: Proposed conventional property name suffixes.

Suffix	Description of Property
<code>_id</code>	A unique identifier, often numerical.
<code>_name</code>	A string label identifier.
<code>_count</code>	A whole number count.
<code>_total</code>	A cumulative result of counts over time.
<code>_list</code>	A list of values.
<code>_sum</code>	The summed total of a list of values.
<code>_time</code>	A numerical measure of time.
<code>_rate</code> <code>_prob</code>	A rate or probability.
<code>_ave</code> <code>_med</code> <code>_min</code> <code>_max</code> <code>_var</code> <code>_std</code> <code>_skew</code> <code>_kurt</code>	Measurements of an observed or calculated distribution (average, median, minimum, maximum, variance, standard deviation, skew, and kurtosis).

ganism that was loaded from a previous experiment? The standard can specify a set of conventional reserved values, giving standard-compliant tools a way to recognize and differentiate these special cases.

Just as it is useful to reserve data values, it may also be useful for the standard to specify reasonable default values for certain conventional (non-required) properties. Accepted and well-documented default values allow tools to make predictable assumptions in cases where conventional properties are missing.

Formatting Data

Our proposed standards framework supports multiple formats for recording and storing data. Every standard-compliant tool will be required to support at least one standard-compliant file types as input; we will curate tools capable of converting data between supported file formats. As of version 1.0.0, the standards support JavaScript Object Notation (JSON) and Comma Separated Values (CSV) formats. As demand builds for additional file formats, the community can develop rules for representing standard data in these new formats (along with converters to and from already-supported formats).

Process for Amending Standards

In the long run, we will model the ALife data standards on other open source projects. As the number of standard-compliant tools increases, so too will the number of people who are invested in their maintenance and improvement. Thus, it is important to establish a process for amending the standards. Our initial set of standards are housed in a repository on GitHub (Lalejini et al., 2019). Anyone can suggest an update to the standards by submitting a pull request or issue. Proposed changes will be reviewed and discussed by interested community members to ensure that 1) they are backwards-compatible (unless there is a compelling reason for a breaking change), 2) they do not replicate or clash with existing standards, and 3) they are well specified, inclusive, and flexible. Following this discussion and a positive consensus, the changes will be merged in. This process fol-

lows the successful precedent set by other open source standards (Darwin Core task group, 2014). We anticipate that additions to the standards will be driven by tool developers adopting conventions for how to name specific types of data.

Changes to the standards will be tracked using semantic versioning, a system of assigning version numbers that conveys information about how similar successive versions are. Versions are identified with a sequence of three numbers (*e.g.*, as of this paper, we are on version 1.0.0), which indicate the “major version”, “minor version”, and “patch” respectively. Typically, a change in the major version denotes a break in backwards compatibility, a change in the minor version denotes the addition of new backwards-compatible features, and a change in the patch denotes minor bug fixes.

Proposed Phylogeny Standard

The problem of how to represent a phylogeny in a standardized way has long been important to biology, even without the wealth of data we have access to in ALife. Biologists have developed a few standard representations for phylogenetic trees (Cranston et al., 2014). Some popular formats include Newick (Cardona et al., 2008), Nexus (Maddison et al., 1997), NHX (Zmasek and Eddy, 2001), PhyloXML (Han and Zmasek, 2009) and RecPhyloXML (Duchemin et al., 2018). Newick format is a simple nested format for representing the hierarchy of a tree (a tree with three nodes might be represented as ((A,B),C), for example). Nexus and NHX formats build upon Newick with additional information (*e.g.*, the inclusion of a genetic sequence alignment among the represented taxa). PhyloXML and RecPhyloXML support the inclusion of additional supplemental data, but still use a nested format.

Why not use one of these existing standards for phylogenies in artificial life? These biological phylogeny standards were designed to work with species phylogenies inferred from extant taxa (and fossil data). As such, these standards are not designed to represent phylogenies with finer taxonomic scale. In artificial life, it is often reasonable to examine the complete phylogeny of every individual that ever existed. Attempting to do so presents two problems for phylogeny standards used in biology: 1) representing any phylogeny where taxa have multiple parents is impossible in these standards, and 2) complete phylogenies can be so big that it is necessary to split them across multiple files – nested formats (as used in biology) do not support such splitting.

During the workshops at ALife 2018 and the 2018 BEACON Congress (Lalejini and Dolson, 2019), we created a proposed standard for representing phylogenies. After continued discussion, we believe that it is now ready to be adopted.

Phylogenies depict parent-offspring relationships over the course of evolution. Phylogenies can be constructed for any taxonomic unit of organization (*e.g.*, individuals, genotypes, species, *etc.*); thus, we use the term “taxon” to refer gener-

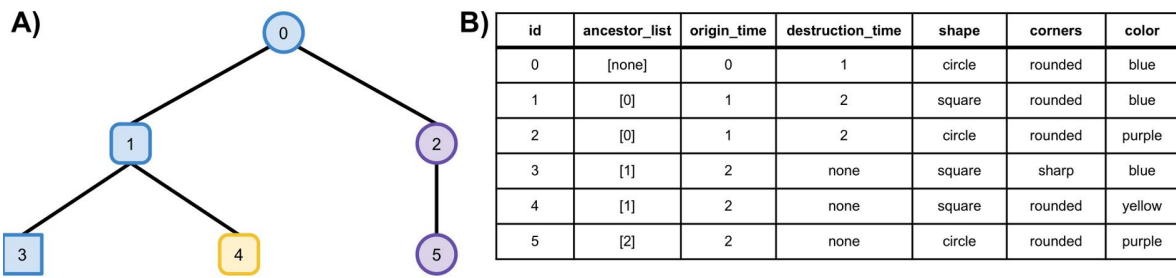


Figure 1: (A) A simple phylogenetic tree where each entity’s id is given inside of its ‘colored-shape’ phenotype. (B) A corresponding standard-compliant data table. The data includes required properties (id and ancestor_list), two optional properties (origin_time and destruction_time), and three extra properties that were used by the visualization (shape, corners, and color).

ally to an entity in a phylogeny. Each taxon in the file must have existed at some point, and each relation from one entity to another defines an ancestor-descendant relationship between the two entities (taxa).

The phylogeny standard has two mandatory properties: **id** and **ancestor_list**. The **id** property provides a unique identifier corresponding to that taxon. The **ancestor_list** property contains a list of ids corresponding to ancestors of the taxon. These are not required to be the direct parents of the taxon, but they will usually be treated as the closest ancestors in the phylogeny. All ids in the **ancestor_list** must correspond to taxa in the file. In cases where a taxon has no ancestors in the file, non-numeric string values can be used to specify that taxon’s origin.

Version 1.0 of the phylogeny standard has two optional properties: **origin_time** and **destruction_time**, which specify the time that the taxon came into and out of existence, respectively. Setting these properties to strings also allows for special values, such as a keyword for **destruction_time** to indicate that a taxon is still alive. Figure 1 gives an example of a phylogeny and its associated standard-compliant description.

Current Software Support

In addition to housing the Artificial Life data standards specifications in a GitHub repository, we plan to maintain a community-curated list of standard-compliant software tools on GitHub (<https://github.com/alife-data-standards/alife-data-tools>; Bohm et al. 2019). This list of software resources is modeled after other efforts to use GitHub as a platform for selecting, evaluating, and organizing public resources for preservation and future use (Wu et al., 2017). Anyone can suggest an update to the list of software resources by submitting an issue or a pull request; for example, a student who has developed a useful visualization tool and written an explanatory blog post would be able to submit an issue to have a link to their visualization and blog post added to the list of software resources.

While concentrating developer effort on a single set of tools will make those tools more reliable, it also increases

the harm that any individual bug in the software can do. To minimize this risk, we advocate the use of software development best practices. All repositories maintained by the Artificial Life Data Standards Organization use continuous integration to ensure that all code is automatically tested when a change is made. Test coverage is measured to facilitate these test suites in becoming comprehensive. Lastly, static analysis is automatically performed to identify error-prone code. Software on the resource list will be classified by reliability based on how well it follows these best practices.

Thus far, we have developed (and are continuing to develop) a variety of software resources to support our proposed phylogeny standard. These resources fall under three broad categories (that are not necessarily mutually exclusive): developer utilities, data converters (to and from the standard), and end-user tools. In addition to being useful on their own, we intend for these resources to serve as templates for developing new standard-compliant software tools. As we develop more software support, we will document them on our list of standard-compliant software tools on GitHub (Bohm et al., 2019). The rest of this section discusses each of the software tools already developed that work with the ALife data standards.

Developer Utilities

Developer utilities include software packages and libraries that can be incorporated into new tools. Thus far, we have begun development on a Python package for working with standardized phylogeny data. We plan to produce a similar set of tools in C++. As additional ALife standards are released, tools to assist developers using that standard will encourage broader adoption.

ALife Standards Development Python Package

The ALife data standards Python package includes functions for loading a standard-formatted phylogeny file as a NetworkX (Hagberg et al., 2008) directed graph object. Python is a popular language for many tasks including data manipulation and analysis; further, many efforts have been made to build interfaces from Python to other languages (e.g., Allaire et al. 2018; Guelton et al. 2015). These benefits, in

addition to its ease of use, make Python an ideal language for developing this initial package of software utilities.

NetworkX is a popular Python package for creating and manipulating graphs. By representing phylogenies as NetworkX graph objects, we can apply existing graph algorithms and visualizations to our phylogenies. Additionally, our Python package contains utilities for saving, manipulating, and analyzing phylogenies and lineages. See <https://github.com/alife-data-standards/alife-std-dev-python> for a more detailed description of this package's current functionality.

Data Converters

Data conversion tools translate data files between formats. Data converters may allow one to use standard-compliant analysis or visualization tools on data produced by non-standard-compliant systems and vice-versa. Data converters may also be developed to translate between different encodings of ALife standard data (*e.g.*, from JSON to CSV, each of which having their pros and cons). We envision data converters serving as the bridge between otherwise incompatible software tools and systems. Thus far, we have developed three data conversion utilities:

Avida to Standard Phylogeny In the Avida Digital Evolution Platform (Ofria and Wilke, 2004), self-replicating computer programs compete, mutate, and evolve. Avida has been used to study a wide range of evolutionary dynamics (*e.g.*, Goldsby et al. 2012; Zaman et al. 2014; Dolson et al. 2016). By default, Avida outputs population files at regular intervals during an experiment. Each population file contains information about the genotypes present in the current population as well as the full ancestral lineages for each extant genotype. Our Avida to standard phylogeny converter takes a single Avida population file as input and converts it into the standard phylogeny format (either as CSV or JSON). This converter and more detailed usage information can be found on GitHub at <https://github.com/alife-data-standards/converters-avida>.

MABE to Standard Phylogeny The Modular Agent-based Evolver (MABE) is a software framework developed to support research in digital evolution and artificial life (Bohm and Hintze, 2017). MABE allows researchers to construct experiments by combining different types *modules*: genomes, brains, environments, and selection methods. These modules can be drawn from an ever-growing collection or be developed by the user as necessary.

MABE outputs ancestry information in a series of population snapshot files (either full snapshots or pruned snapshots without reproductively unsuccessful individuals). The MABE to standard phylogeny converter takes these MABE snapshot files and optionally a list of column names that should be included in the standard phylogeny output. This converter and more detailed usage informa-

tion can be found on GitHub at <https://github.com/alife-data-standards/converters-mabe>.

Standard Phylogeny to VINE The Visual Inspector for NeuroEvolution (VINE) (Wang et al., 2018) is under active development by UBER Labs in conjunction with their Deep Neuroevolution project (Such et al., 2017). VINE allows users to visualize how an evolving population moves through trait space over time. The standard phylogeny to VINE converter allows users to identify which properties of their phylogeny data should be translated into the VINE input format and creates the VINE-compliant input files (in the appropriate directory structure). For more information about VINE, see (Wang et al., 2018). More detailed usage information for our standard phylogeny to VINE converter can be found on GitHub at <https://github.com/alife-data-standards/converters-vine>.

End-user Tools

In support of the phylogeny standard and for our own research purposes, we have developed data-processing scripts and visualizations. These tools require users to provide standard-compliant input files (via a command line or graphical interface), processing the input as part of a data processing pipeline or producing a visualization of the given data.

Phylogeny Web Visualization Visualization is a critical part of data analysis, helping us build intuitions and communicate our results. We are actively developing a web-based phylogeny visualization tool that takes standard-compliant asexual phylogeny data as input and generates a phylogenetic tree, color-coded based on a user-specified numeric property (see Figure 2 for example output). Refer to our GitHub repository for more detailed usage information for this tool (https://github.com/emilydolson/lineage_viz_tool).

Time to Coalescence Command Line Tool The time to coalescence command line tool takes a standard phylogeny as input calculates how far back in time we need to look to find the most recent common ancestor of all extant taxa. The tool searches the given standard phylogeny file for the organisms with the greatest *origin_time*. It then traces the ancestors recursively until it finds the most recent common ancestor (MRCA), at which point the tool returns the time to coalescence and the id value of the MRCA. See our GitHub repository for more detailed usage information for this tool (<https://github.com/alife-data-standards/tools-pack-phylogeny>).

Conclusions and Future Directions

In this work, we discussed the motivation and vision for a set of ALife data standards. Additionally, we proposed a standard for describing and storing phylogeny data and presented several software tools that have already been devel-

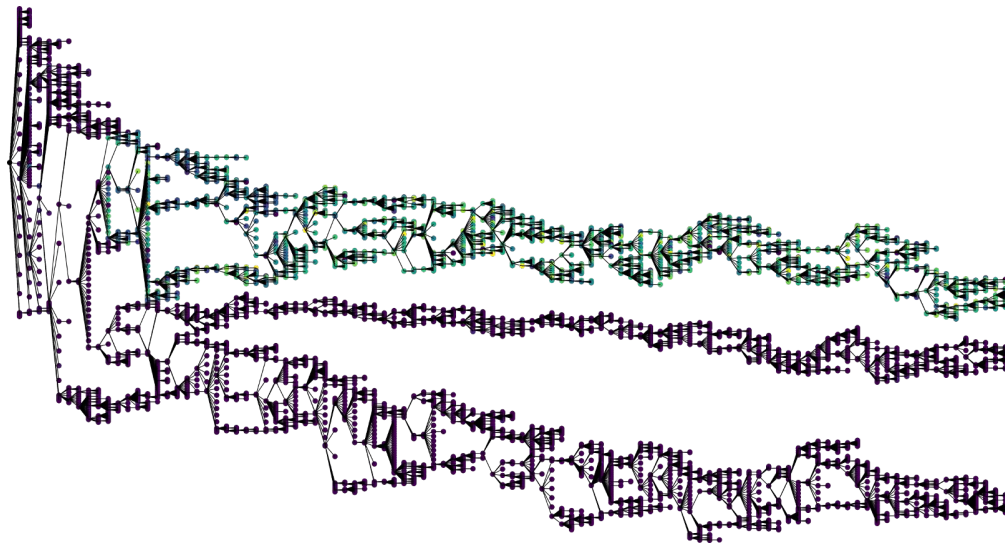


Figure 2: Example output generated using the phylogeny web visualizer. Generations proceed from left to right with the extant taxa shown on the far right. Nodes can be color-coded; here green and yellow indicate relatively high numeric values for the specified property.

oped to support the standard. We see this work as a continuation of the conversation started at the discussion-driven workshops at the 2018 Conference for Artificial Life and the 2018 Congress for the BEACON Center for the Study of Evolution in Action. By reaching out to the wider ALife community, we hope to broaden the scope of our standards, continue developing software tools that work with standardized data, and build community support for adopting and improving data standards.

Ultimately, the success our proposed data standards will depend on the level of community buy-in and adoption. The utility of these standards will grow as more of the community adopts and contributes to our ecosystem of data standards. To ensure an inclusive environment for standards development and discussion, we have adopted a Contributor Covenant code of conduct (Covenant, 2014).

As we only represent a subset of the ALife community, we do not know the full set of data standards that would be valuable to the community; for this, we turn outward: what types of data should we develop standards for? In workshop discussions, we identified the following targets for future data standards: genomes, interaction networks, fitness landscapes, and meta-data. While genomes can be broadly defined as heritable and mutable material, developing a genome standard has proven elusive because of the enormous variety of genetic representations used across different systems. Any adopted genome standard should be flexible enough to support the varied types of both artificial and natural genomes; this would allow us to make direct comparisons between digital and biological systems and make the tools we develop useful to biologists. Interaction networks describe the relationships between interacting entities (*e.g.*, objects, individual organisms, chemicals, *etc.*);

for example, a food web is a type of interaction network, describing the predator-prey relationships among species represented in the network. A fitness landscape characterizes the mapping between the space of possible genotypes (or a set of phenotypic traits) and fitness for a given environment. For example, given a genome and an environment, the fitnesses of all possible one-step mutants describes the local fitness landscape adjacent to the given genome. A fitness landscape standard would allow researchers to more effectively compare fitness landscapes across multiple environments and better study how populations move through a fitness landscape over the course of evolution. Meta-data provide context for other data; for example, meta-data might identify the system or the parameters used to generate a data set (*e.g.*, a phylogeny). A standard for meta-data would facilitate improved data annotation and documentation, allowing researchers to more easily replicate experiments from other research groups.

Acknowledgements

We thank workshop participants at ALife 2018 and the 2018 Congress for the BEACON Center for the Study of Evolution in Action for thoughtful discussion and feedback on developing software standards for the Artificial Life community. This work was supported by the National Science Foundation (NSF) through the BEACON Center (Cooperative Agreement DBI-0939454), Graduate Research Fellowships to ED and AL (Grant No. DGE-1424871), and NSF Grant No. DEB-1655715 to CO.

References

Allaire, J., Ushey, K., and Tang, Y. (2018). *reticulate: Interface to 'Python'*. R package version 1.7.

- Board, M. I. (1999). Mars climate orbiter mishap investigation board phase i report november 10, 1999.
- Bohm, C. and Hintze, A. (2017). Mabe (modular agent based evolver): A framework for digital evolution research. In *ALife Conference Proceedings 14*, pages 76–83. MIT Press.
- Bohm, C., Lalejini, A., Dolson, E., and Ferguson, A. (2019). Software Resources for the Artificial Life Data Standards. DOI: 10.5281/zenodo.2595536 URL: <https://github.com/alife-data-standards/alife-data-tools>.
- Cardona, G., Rosselló, F., and Valiente, G. (2008). Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics*, 9(1):532.
- Covenant, C. (2014). A code of conduct for open source projects. *Coraline Ada Ehmke*. Available at: <https://www.contributor-covenant.org/>.
- Cranston, K., Harmon, L. J., O’Leary, M. A., and Lisle, C. (2014). Best Practices for Data Sharing in Phylogenetic Research. *PLOS Currents Tree of Life*.
- Darwin Core task group, B. I. S. T. (2014). Darwin Core: 2014-11-08. DOI: 10.5281/zenodo.12694.
- Dolson, E., Wiser, M. J., and Ofria, C. A. (2016). The Effects of Evolution and Spatial Structure on Diversity in Biological Reserves. In *Artificial Life XV: Proceedings of the Fifteenth International Conference on Artificial Life*, pages 434–440, Cancun, Mexico. MIT Press.
- Duchemin, W., Gence, G., Chifolleau, A.-M. A., Arvestad, L., Bansal, M. S., Berry, V., Boussau, B., Chevenet, F., Comte, N., Davin, A. A., Dessimoz, C., Dylus, D., Hasic, D., Mallo, D., Planel, R., Posada, D., Scornavacca, C., Szöllösi, G., Zhang, L., Tannier, É., and Daubin, V. (2018). RecPhyloXML: a format for reconciled gene trees. *Bioinformatics*.
- Gleeson, P., Crook, S., Cannon, R. C., Hines, M. L., Billings, G. O., Farinella, M., Morse, T. M., Davison, A. P., Ray, S., Bhalla, U. S., et al. (2010). Neuroml: a language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS computational biology*, 6(6):e1000815.
- Goldsby, H. J., Dornhaus, A., Kerr, B., and Ofria, C. (2012). Task-switching costs promote the evolution of division of labor and shifts in individuality. *Proceedings of the National Academy of Sciences*, 109(34):13686–13691.
- Guelton, S., Brunet, P., Amini, M., Merlini, A., Corbillon, X., and Raynaud, A. (2015). Pythran: Enabling static optimization of scientific python programs. *Computational Science & Discovery*, 8(1):014001.
- Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.
- Han, M. V. and Zmasek, C. M. (2009). phyloXML: XML for evolutionary biology and comparative genomics. *BMC Bioinformatics*, 10(1):356.
- Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., et al. (2003). The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531.
- Lalejini, A. and Dolson, E. (2019). 2018 standards workshop description and notes. DOI: 10.5281/zenodo.2592691 URL: <https://github.com/alife-data-standards/ALIFE2018-Standards-Workshop>.
- Lalejini, A., Dolson, E., Ferguson, A., Bohm, C., and Rainford, P. F. (2019). Alife data standards. Version 1.0-alpha. DOI: 10.5281/zenodo.2577410. URL: <https://github.com/alife-data-standards/alife-data-standards>.
- Maddison, D. R., Swofford, D. L., and Maddison, W. P. (1997). Nexus: An Extensible File Format for Systematic Information. *Systematic Biology*, 46(4):590–621.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- Parr, C. S., Guralnick, R., Cellinese, N., and Page, R. D. (2012). Evolutionary informatics: unifying knowledge about the diversity of life. *Trends in Ecology & Evolution*, 27(2):94–103.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.
- Richmond, P., Cope, A., Gurney, K., and Allerton, D. J. (2014). From model specification to simulation of biologically constrained networks of spiking neurons. *Neuroinformatics*, 12(2):307–323.
- Smith, B. and Welty, C. (2001). FOIS introduction. In *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, volume 2001, pages .3–9, New York, New York, USA. ACM Press.
- Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. (2017). Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv:1712.06567 [cs]*. arXiv: 1712.06567.
- Taylor, T., Auerbach, J. E., Bongard, J., Clune, J., Hickinbotham, S., Ofria, C., Oka, M., Risi, S., Stanley, K. O., and Yosinski, J. (2016). WebAL Comes of Age: A Review of the First 21 Years of Artificial Life on the Web. *Artificial Life*, 22(3):364–407.
- Wang, R., Clune, J., and Stanley, K. O. (2018). Vine: an open source interactive data visualization tool for neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1562–1564. ACM.
- Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., Dring, M., Giovanni, R., Robertson, T., and Vieglais, D. (2012). Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. *PLoS ONE*, 7(1):e29715.
- Wren, J. D. (2016). Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. *Bioinformatics*, 32(17):2686–2691.
- Wu, Y., Wang, N., Kropczynski, J., and Carroll, J. M. (2017). The appropriation of GitHub for curation. *PeerJ Computer Science*, 3:e134.
- Zaman, L., Meyer, J. R., Devangam, S., Bryson, D. M., Lenski, R. E., and Ofria, C. (2014). Coevolution drives the emergence of complex traits and promotes evolvability. *PLoS Biol*, 12(12):e1002023.
- Zhang, Z., Bajic, V. B., Yu, J., Cheung, K.-H., and Townsend, J. P. (2011). Data integration in bioinformatics: Current efforts and challenges. In Mahdavi, M. A., editor, *Bioinformatics*, chapter 2. IntechOpen, Rijeka.
- Zmasek, C. M. and Eddy, S. R. (2001). ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, 17(4):383–384.

Organic Selection and Social Heredity: The *Original* Baldwin Effect Revisited

Nam Le

Natural Computing Research & Applications Group
University College Dublin, Dublin, Ireland, A94 XF34
namlehai90@gmail.com

Abstract

The so-called “Baldwin Effect” has been studied for years in the fields of Artificial Life, Cognitive Science, and Evolutionary Theory across disciplines. This idea is often conflated with genetic assimilation, and has raised controversy in trans-disciplinary scientific discourse due to the many interpretations it has. This paper revisits the “Baldwin Effect” in Baldwin’s original spirit from a joint historical, theoretical and experimental approach. Social Heredity – the inheritance of cultural knowledge via non-genetic means in Baldwin’s term – is also taken into account. I shall argue that the Baldwin Effect can occur via social heredity without necessity for genetic assimilation, instead the Baldwin Effect can promote more plasticity to facilitate future intelligence when the fidelity of social heredity is high. Computational experiments are then carried out to support the hypothesis of interest. The role of mind and intelligence in evolution and its implications in an extended synthesis of evolution are briefly discussed.

Introduction

Studying the relationship between evolution and learning is a very important topic in understanding adaptive behaviour demonstrated by both natural and artificial agents. There exists an intriguing idea called the **Baldwin Effect** by Simpson (1953), named after James Mark Baldwin, as an interpretation of *Organic Selection* proposed by Baldwin (1896). Since Simpson, this idea has often been interpreted as how an adaptive behaviour first acquired during lifetime can later be replaced by fixed innate traits due to the cost of individual learning. This interpretation has often conflated the Baldwin Effect with *genetic assimilation* by Waddington (1953). The Baldwin Effect studied in ALife and complex adaptive systems often used the interpretation of Simpson (Hinton and Nowlan (1987), Harvey (1996), Mayley (1997)).

This interpretation, intentionally or unintentionally, has made the Baldwin Effect more restrictive than what Baldwin originally proposed through organic selection. Most studies in this line of thought, including those in ALife, often neglected the presence and importance of **social heredity** – what Baldwin (1896) originally meant by a parallel heredity of social knowledge via non-genetic means. Baldwin’s ideas of social heredity and its influence on evolution bear some

sort of similarity to what we now call *gene-culture coevolution* or *dual-inheritance theory* (Peter J. Richerson (2005)).

When social heredity comes in, the story would be more interesting as to how the Baldwin Effect occurs. Some of the interesting questions could be asked as: if social heredity is permitted, then if adaptive information can be gained and transmitted easily through social transmission, what would genetic assimilation look like? More curiously, is genetic assimilation necessary to claim the presence of the Baldwin Effect as studied previously? Plausibly, it seems to us that if adaptive behavioural information is encoded into culture, and this information can be handed down easily from generation to generation by some form of social learning, *ontogenetic* learning still plays a role in directing evolution but the assimilation step seems not to be required. After ‘the new factor’ in Baldwin (1896), Baldwin stressed the importance of social heredity more in his later books (Baldwin (1902), Baldwin (1909)) that I think it is worth a further investigation to understand what he really meant by his *effect*.

The main aim of this paper is to re-discuss the Baldwin Effect in Baldwin’s original spirit to clarify what the *effect* would possibly be. My contribution in this paper can be divided into two parts. First, I present and discuss the history of understanding to show why and how Baldwin’s original idea, and ideas, may differ from the rich literature studying the Baldwin Effect. I will prove that Baldwin did not restrict his new factor in evolution to the idea of genetic assimilation, instead he believed that social heredity can provide another way to affect evolution, which may promote plasticity to boost the intelligence of an evolving system. Second, a simple computer simulation, combining evolution, learning, and cultural inheritance, is carried out in order to see how this combination affects the underlying evolutionary process, and whether this would-be effect requires a strict requirement of genetic assimilation. The last section briefly presents some future implications of the Baldwin Effect in various avenues, including the present-day interest in extending and expanding Darwinian account of evolution into a new synthesis (Laland et al. (2015)).

The Baldwin Effect

For clarity, I shall use the term *Baldwin-Baldwin Effect* to refer to the Baldwin's original effect, and Baldwin-Simpson Effect as a reference to Simpson's re-interpretation.

I. A Brief History of Understanding

A. The Baldwin-Baldwin Effect:

At the turn of the 20th century, the idea that learning as part of the ontogenetic adaptation can influence, and somehow *direct* an evolutionary process without resorting to Lamarckian inheritance, was proposed independently by at least three independent thinkers: Baldwin (1896) (published in *The American Naturalist*), Osborn (1896), and Morgan (1896) (both published in *Science Magazine*). Baldwin (1896) re-discussed and joined his two previous ideas, published in *Mental Development in the Child and the Race* (Baldwin (1895)), on *Organic Selection* (chap. vii) and *Social Heredity* (chap. xii), and called this "A new factor in evolution".

When first appeared, the idea by Baldwin (also Morgan and Osborn) set a new movement in understanding how evolution works, more specifically when it comes to explaining the inheritance of acquired characteristics. Before Baldwin and the like, the French Naturalist Lamarck proposed that characters acquired during the lifetime of the parent are directly passed down onto the offspring. The English Philosopher Herbert Spencer seemed to agree with Lamarckian inheritance when he said "intelligence would allow an animal to acquire complex habits that would later solidify into instincts. But such transformation required Lamarckian inheritance" (Richards (1989)). Darwin himself believed that Lamarckian evolution might play a small role in life, but most Darwinians rejected Lamarckism (Huxley (1942)) based on Weismann et al. (1893).

Baldwin came to light and explained evolution without resorting to Lamarckian style, in which acquired characters are somehow *indirectly* inherited. A new factor in evolution by Baldwin is organic selection, which includes any form of individual adaptation during the lifetime (through Physico-genetic, Neuro-genetic Psycho-genetic) that directs the evolutionary pathway of an evolving species. He stressed the role of psycho-genetic, by which he meant conscious intelligence, that includes any form of ontogenetic learning, such as imitation, pleasure and pain, reasoning. For Baldwin, it is organic selection that can explain how a behaviour that has learned might be becoming *innate*, or *partially innate*, in future generations. If a group of animals migrates into a new environment for which they initially lack congenital adaptations, those **plastic** enough to accommodate themselves through conscious learning will tend to survive, blocking the strong hand of natural selection. This will allow natural selection opportunity to accumulate chance variations that follow the path laid down by the acquired behaviours. Acquired characteristics are immediately heritable implied a loss of phenotypic flexibility. Such inheritance would tend

so to bind up the child's nervous substance in fixed form that he [or she] would have less or possibly no plastic substance left to learn with.

Interestingly, Baldwin did insist the importance of what he termed **Social Heredity** – a means of extra-organic transmission from generation to generation through copying, imitation, teaching, or any form of social learning. Baldwin (1896) considered it heredity because of the following reasons: 1) it is a handing down of physical functions; while it is not biological (physical) heredity; 2) it directly influences physical heredity in the way mentioned, i.e., it keeps alive variations, thus sets the direction of ontogenetic adaptation, thereby influencing the direction of the available congenital variations of the next generation. Of course, social heredity is a form of organic selection or ontogenetic adaptation, but it deserves a special name because of its special way of operation and its farther value. It keeps alive a series of functions which either are **not yet, or never do become, congenital at all**.

Fixity or Plasticity

Baldwin (1896) said: "The two ways of securing development in determinate directions – the purely extra-organic way of Social Heredity, and the way by which Organic Selection in general (both by social and by other ontogenetic adaptations) secures the fixing of phylogenetic variations, as described above – seem to run parallel". And more importantly he concluded that in more complex living animals like humans, "*social transmission is an important factor, and the congenital equipment of instincts is actually broken up in order to allow the plasticity which the human being's social learning requires him to have*".

Later in *Development and Evolution* Baldwin (1902) said "organic selection opens a great sphere for the application of the principle of natural selection among organisms, i.e. selection on the basis of what they do rather than what they are; of the new use they make of their functions rather than of the mere possession of certain congenital characteristics. A premium is set on plasticity and adaptability of function rather than on congenital fixity of structure; and this adaptability reaches its highest levels in the intelligence" (p. 117).

By looking further into his work, it can be found in *Darwin and The Humanities* in which Baldwin (1909) presented that "in cases where the intelligent or other adjustive factor is on the whole of greater utility, variations towards the disintegration of the instinctive congenital part, would be selected. The growth of intelligent action superseding instinctive" (p. 21), and that "once admit that the intelligence, even in its simplest forms, as seen in imitation, play and the resulting accommodative actions, can be applied to the learning of anything, and that variations in plasticity are selected to allow of its development this once admitted, we have the possibility of a continuous handing down from generation to generation, a Social Heredity, which is no longer subject to the limitations set upon physical heredity" (p. 28).

Here it seems to us that for Baldwin, with social heredity, there is **no need of fixing phylogenetic variation for previously acquired behaviour** if organisms can easily acquire those behaviour through imitation, teaching, or just copying.

B. The Baldwin-Simpson Effect:

There have been quite a few reasons why the Baldwin's idea was not common in the literature of both psychology and biology. I do not want to go too far here, yet one of those was *the Baltimore scandal* in which the head of psychology department of Johns Hopkins University (Baldwin) was caught by the police, which then made him mostly disappear from any scientific community (Horley (2001)).

Baldwin's original idea of organic selection seemed to come back to scientific discourse through Simpson (1953), appeared in *Evolution* 1953, in which the idea was first called the *Baldwin Effect*. Interestingly, Simpson's interpretation of the Baldwin Effect seemed to be stimulated by the idea of *genetic assimilation* by Waddington (1953) in the same issue. We shall call this the Baldwin-Simpson Effect since it has some differences from the original version.

Through Simpson's interpretation, the Baldwin Effect (or the BS Effect) occurs in two phases: Phase 1, individuals that through lifetime learning acquire an adaptive behaviour needed for the survival in its current environment occupy the population; and Phase 2, then the evolutionary path finds the innate trait that can replace the learned trait because of the *cost of individual learning*. Phase 2 was conflated with the idea of genetic assimilation of acquired characters by Waddington (1953) in his experiments to study epigenetics with drosophila. Interestingly and ironically, Simpson (1953) gave birth to the catchy name of the effect just for the intention of deflating the interest in the Baldwin Effect. Simpson was skeptical of the Baldwin effect as he posited that if learned behaviors do become genetically underwritten, a population will favour long-term fixed adaptation at the cost of short term and more plastic [learned behaviors], thus corrupting the point of the Baldwin effect. By the early sixties, a deeper skepticism came from a famous figure in evolutionary theory, Mayr (1963), which was then followed by Dobzhansky (1970). These authors all disagreed with Phase 2 of the Baldwin Effect (or BS effect) as evolution should favour plastic phenotype, rather than collapsing *norms of reaction* for fixity.

C. The Baldwin-Simpson Effect in Computation

The Baldwin Effect gradually gained more attention since the classic and elegant computational model by Hinton and Nowlan (1987) (henceforth H&N). H&N used the same metaphor as Simpson and attempted to demonstrate that the Baldwin Effect (or the BS effect) can occur. Figure 1 describes the detail of a replication of H&N's model which results in the same conclusion. The result from Hinton and Nowlan (1987) did stimulate the doyen of British biologists

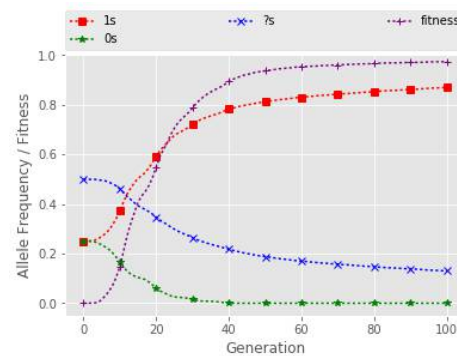


Figure 1: Replication of H&N's experiment. The task is to find the all-ones target string 111...1 (20 bits). There is only one correct solution, the target string, which has the fitness of 20. All other configurations are wrong and have the same fitness of 1. This forms a Needle-in-a-haystack landscape whereby an evolutionary search alone cannot find the solution. H&N used a different encoding. A genotype now is initialised with 3 alleles: 25% 0, 25% 1, and 50% ?. The plastic allele ? allows for lifetime learning (or plasticity), over 300 rounds (since the H&N's original 1000 was often criticised as too big by many). On each round, an individual agent is allowed to perform individual learning by changing its allele ? to either 0 or 1 as the expressed value. After learning, the fitness of an individual is calculated as: $1 + 19(300-n)/300$ (n is the learning trials performed to find the solution). The population consists of 1000 individuals, crossover is only the genetic operator employed, and selection is based on fitness-proportionate as in Hinton and Nowlan (1987). We run the simulation through 100 generations, and over 30 independent runs. The frequency of the allele is plotted against the average fitness normalised in $[0, 1]$. There is small difference in detail perhaps due to different programming environments, yet the overall trend is the same with the original model. The Baldwin-like Effect is claimed as the frequency of 0 disappears, the frequency of correct allele 1 is increased (also the average fitness), and the frequency of plastic allele ? decreases as an instance of genetic assimilation due to the cost of individual learning.

Maynard Smith (1987) to feature "when learning guides evolution" in *Nature Magazine*. Dennett (1991) adopted the same idea to explain consciousness.

The model developed by Hinton and Nowlan, though simple, is interesting, as it opens up the trend followed by a number of studies investigating the Baldwin Effect, or how learning affects evolution, in the computer, including Mayley (1996), Harvey (1996), Mayley (1997), Suzuki and Arita (2007)). These studies interpret the Baldwin Effect in two phases, and stress the importance of the assimilation phase. Mayley (1997) and Mayley (1996) studied quite thoroughly how the cost-benefit trade-off of individual learning that could trigger genetic assimilation. Interestingly, the H&N's model has been criticised that it could not reach the state when the whole adaptive behaviour (all-ones) is assimilated, leaving no plasticity (Harvey (1996), Santos et al. (2015)). The so-called *effect* has also been employed in artificial intelligence, yet the goal is to to borrow phenomena of evolution and learning (even social learning) to create more intelligent agents to solve a problem of interest, rather than understanding the Baldwin Effect (Le (2019), Le et al. (2019)). All of these studies, for or against the effect, rely on the re-interpretation of Simpson, or the BS Effect.

D. The Recovery in Modern-day Interest

More than a century later, the ideas set out by Baldwin have also been recovered in other fields such as Evo-Devo (West-

Eberhard (2003)), *Cognitive Science* (Dennett (1991)). Especially, in an edited book by Weber and Depew (2003), present-day discussions about the Baldwin effect from different points of view, including epigenetics, language evolution, niche construction theory (Odling-Smee et al. (2003)) are presented. The Baldwin's 1986 paper was also cited in the recent movement in Evolutionary Biology, called the *Extended Evolutionary Synthesis* (EES) (Laland et al. (2015), Pigliucci (2007)), which tries to incorporate many factors, including epigenetics and developmental processes (West-Eberhard (2003)), in evolution that have been neglected for years in the mainstream evolutionary biology. I shall not be going too far at this moment yet it can be seen that the Baldwin Effect, which emphasises the active role of intelligence or phenotypic plasticity in evolution, can fit into, and even somehow boost the active status of the EES framework.

However, many of them are still not so clear whether the Baldwin effect requires the need for acquired characters to be assimilated. West-Eberhard (2003) says that "Baldwin conceived of it (organic selection) as a mechanism that could, in principle, lead to the reduction of plasticity as the trait in question comes under increasingly powerful genetic influence. Yet this stands at odds with the remarkable flexibility exhibited by observed organisms". The whole book dedicated for the reconsideration of the Baldwin effect by Weber and Depew (2003) also presents the controversy within the selected authors in that edition on the issue of genetic assimilation, which has led to an even stronger skepticism of what the Baldwin Effect really is, as reviewed by Sterelny (2004) and Shettleworth (2004). Shettleworth even concluded her review by referring to Depew, saying that there is really no such thing as the Baldwin Effect.

Paradoxically, what is missing from the majority of the available bibliography is the original viewpoint from which Baldwin actually formed his theory of organic selection and social heredity. Most of the contemporary discussions on the Baldwin effect seem to rely on the Simpson's interpretation. As we have argued so far, Baldwin's original factor in evolution can argue that organic selection can drive greater plasticity, escaping from genetic assimilation.

E. Concluding Remarks

Now we can feel at ease to conclude that originally Baldwin stressed on the importance of intelligence, which includes ontogenetic learning as a form of phenotypic plasticity, in directing evolution. He was right to say that the future evolution will follow the path laid by what adaptive behaviour has been acquired before. Indeed, social heredity should not be neglected when studying the "effect" on evolution.

We can offer another important point here. It was the re-interpretation of Simpson that conflated the Baldwin Effect with the idea of genetic assimilation that has raised a strong skepticism of the effect. This interpretation has had a relatively strong influence on the study of the Baldwin Effect

in many disciplines, including ALife. This, indeed, restricts the original idea of the Baldwin-Baldwin Effect. Moreover, it is the lack of social heredity in the Baldwin-Simpson Effect that made the skepticism even stronger. What has been shown informs us that there exists a scenario, with the presence of social heredity, in which the Baldwin Effect occurs differently from the genetic assimilation process as often believed previously, promoting more plasticity to facilitate future intelligent acquisitions by learning.

In the next section I briefly present what Baldwin thought of social heredity and its relationship to the contemporary research on social learning and cultural evolution. I then describe the experiment to study the Baldwin-Baldwin Effect through the prism of social heredity.

II. The Baldwin Effect through Social Heredity

A. Social Heredity

Baldwin proposed social heredity as an important inheritance mechanism in which cultural knowledge and values can be transmitted both within and between generations. Baldwin (1909) said that "when we come to ask for a full account of the propagation of mental acquisitions from generation to generation, we find it necessary to recognise another form of handing down or real transmission" (p. 28). In *Mental Development*, Baldwin described social heredity as largely independent of physical heredity. However, Baldwin (1896), Baldwin (1902), and Baldwin (1909) later acknowledged that the two modes of inheritance can interact and have influence on each other. Baldwin (1902) wrote that "social heredity keeps certain variations alive, thus sets the direction of ontogenetic accommodation thereby influences the direction of the available congenital variations of the next generation, and so determines phylogenetic evolution" (p. 103).

Interestingly, what Baldwin once proposed more than 100 years ago bears a flavour similar to the so-called *gene-culture coevolution*, or *dual-inheritance theory*, currently promoted by cultural evolution researchers, such as Peter J. Richerson (2005), Lumsden and Wilson (2005). Gene and culture are said to co-evolve to further adaptivity of social or cultural species. Learning, both asocial (individual) and social, are media to trigger the establishment and transmission of cultural adaptations. More interestingly, the cost-benefit relationship between social learning (SL) and individual learning (IL) can produce variable evolutionary dynamics (Laland (2018), Peter J. Richerson (2005)). A combination of both trial-and-error and imitation learning is often said to produce more adaptivity, especially in human cultural evolution (Peter J. Richerson (2005)). Importantly, culture has been said to emerge only when the fidelity of cultural transmission is high (Laland (2018)). We shall incorporate fidelity of cultural transmission in our experiments in the next section.

B. Experiments and Results

In this section I present a simple computer simulation as an extension of H&N’s replication, combining evolution, individual learning, and cultural inheritance. Cultural inheritance here is understood as the transmission of behaviour from parents to their offspring, vertically via social, or imitation, learning. Some limitation on this computational model should be noted. First, as previously shown in Harvey (1996), Mayley (1996), the H&N’s landscape is extreme. Individual learning is quite random. Importantly, it was mostly criticised as it cannot lead to the absolute assimilation of the correct behaviour (all-ones), thus it is not the Baldwin Effect (Santos et al. (2015)). However, as I have shown in the theory part, the Baldwin’s original effect does not necessarily mean the assimilation of acquired characters is required. Indeed, we shall be seeing the reverse.

For that reason, we can feel at ease to replicate the elegant H&N’s model. For now, it is the transparent simplicity of H&N’s original work which is critical to its impact; such simplicity is our preference while adding new mechanisms to study the effect of interest by two experimental setups.

B1. Setup I: Evolution with Social Learning alone

I propose the social learning procedure via imitation as described in Algorithm 1 below. The imitative process works as follows: For each question mark position, the observer will decide whether to copy exactly the *trait* or a mutated version of that *trait* from the demonstrator based on the parameter *fidelity* which represents the fidelity of the social transmission. By default, the *fidelity* is set to 1, that means imitative process will copy exactly the values from the demonstrator to the observer.

Algorithm 1 IMITATION

```

1: function IMITATION(observer, demon, fidelity = 1)
2:   questions = [] comment: question mark array
3:   for position i in observer.pheno do
4:     if i = ? then
5:       questions.add(i)
6:       observer.learning_attempt += 1
7:     end if
8:   end for
9:   for i in questions do
10:    if rand() < fidelity then
11:      observer.pheno(i) = demon.pheno(i)
12:    else
13:      observer.pheno(i) = 1 - demon.pheno(i)
14:    end if
15:   end for
16: end function

```

Algorithm 2 presents the process in which evolution is combined with only social learning in place of asocial learning as in H&N’s model (denoted by EVO+SL). The demonstrator is set to be the better parent of an individual. This represents a vertical cultural inheritance process, as described above. After social learning, the population operates an evolutionary process as in H&N’s model described in Figure 1.

Algorithm 2 EVO + SL

```

1: function EVO+SL(pop, fidelity = 1) comment: Do life-time learning
2:   for ind in pop do
3:     demon = ind.max_parent() comment: extract the better parent
4:     Imitation(ind, demon, fidelity) comment: do imitation
5:   end for
6:   comment: Evolve the population
7:   Do selection, reproduction, replacement
8: end function

```

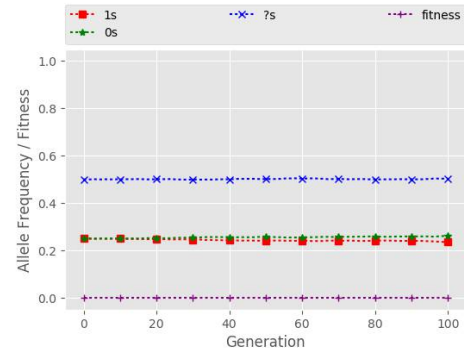


Figure 2: EVO+SL alone. Fitness is normalised in [0,1]

Look at the result in figure 2, without individual learning, social learning fails to guide evolution in the H&N’s landscape. The Baldwin effect does not show up in this case. Figure 2 shows that frequency of all three alleles keeps relatively constant. No individual can find the solution, as shown in the lowest average fitness.

It is not hard to explain this. SL is *information-parasitism* – can only learn from information, or solution, produced by others. The H&N’s landscape is quite special in this case. Without individual learning, there is no gradient for evolution to seek for the solution. In other words, without the presence of individual learning, no solution will be found in the evolving population. All evolving individuals are wrong. Social learners that copy from their wrong parents become wrong. Simply speaking, social learning cannot learn anything that has not been learned.

There is no influence of organic selection on evolution in this case, hence no Baldwin-Baldwin Effect.

B2. Setup 2.2: Evolution + IL + SL

Based on the analysis above, we design Algorithm 3 combining evolution with both social and asocial learning, or evolution with a *learning strategy*. A strategy is set as at each generation, an agent performs social learning based on Algorithm 1 only when its demonstrator is correct, otherwise the agent seeks for the solution individually. The demonstrator of an agent is again the better individual amongst its parents. The demonstrator is said to be correct when its fitness value is greater than 1. This is because 1 is the lowest fitness in our landscape, and an agent has its

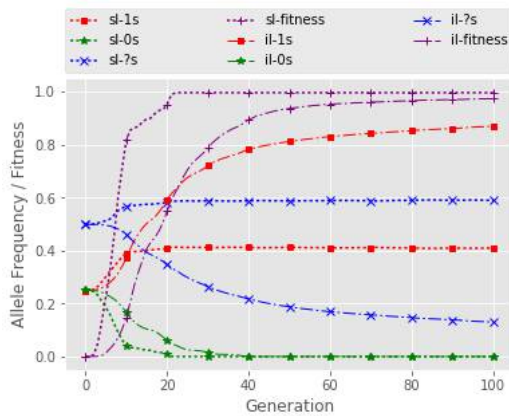


Figure 3: EVO+IL+SL vs EVO+IL. sl = EVO+IL+SL, il=EVO+IL.

fitness greater than 1 only when it successfully found the solution. After this lifetime learning process, the population goes through selection and reproduction.

Algorithm 3 EVO+IL+SL

```

1: function EVO+SL(pop, fidelity = 1) comment: Do life-time learning
2:   for ind ∈ pop do
3:     demon = ind.max_parent() comment: extract the better parent
4:     if demon.fitness > 1 then
5:       Imitation(ind, demon, fidelity)
6:   else
7:     ind.individual_learning()
8:   end if
9: end for
comment: Evolve the population
10: Do selection, reproduction, replacement
11: end function

```

In Figure 3, we plot our EVO+IL+SL against the H&N’s setup (EVO+IL) to see the difference between the two “effects”. It is shown that social learning in combination with asocial learning can also *direct* the underlying evolutionary process. More specifically, we see that the frequency of wrong allele (0s) drops to zero quicker in EVO+IL+SL (at around generation 20). Contrary to the effect found in EVO+IL, EVO+IL+SL maintains a higher proportion of plasticity than the correct allele (1s). After generation 20, all the alleles in EVO+IL+SL relatively keep constant. This means there is no pressure to replace the plasticity with the fixation of 1s. Also, the average fitness of EVO+IL+SL reaches the higher point and sooner than that of EVO+IL.

How the Baldwin Effect can be interpreted here? We observe that the behaviour of EVO+IL+SL can be divided into two phases: In the first phase, which includes 20 first generations, through individual learning some agent can find the solution. That successful agent should have no 0s in its genetic composition at first, and will be favoured by selection, leaving more offspring, promoting its allele configuration (with 1s and ?s) in later generations. Moreover, the offspring of successful agents (without 0s) tends to have its

genotype consisting of no 0s. Since its parent now is successful, via **social heredity** that offspring can **copy** the successful behaviour from its parent, and becomes successful too. Its genetic makeup will also be promoted, without 0s. Thus the proportion of 0 will quickly diminish.

In the second phase, we observe that there is relatively no change in frequency of 1s and ?s, and the average fitness reaches its highest point. The explanation for the observation here is that once the frequency of 0s is zero, every individual in the population will have only 1s and ?s in its genotype. Each individual agent now has a chance to be successful via individual or social learning. We call it *potential agent* from now. Moreover, once the correct solution is found (in previous generations), the cultural inheritance as a vertical transmission will transmit the correct behaviour down to generations very quickly since the potential learner can copy **exactly** the solution yet with little learning attempt (the nature of our imitation algorithm). The fitness function as depicted in Figure 1 says that a lower learning cost results in a higher fitness for the learner. Therefore, the average fitness of the population in our Evo+IL+SL is higher than that in EVO+IL. That also indicates that having more plastic alleles, specifying the ability to learn socially, is more adaptive in the future, hence the dominance of ‘?s’.

Information Fidelity

One notable factor in the explanation above is the ability to transmit **exactly** the solution down to later generations. I argue that the default *fidelity* = 1 makes it much easier for the child to copy the correct solution with the much less cost. This indicates that the information fidelity could have an influential role on the effect of social heredity on evolution. This argument should be validated by running EVO+IL+SL with different levels of fidelity. For example, here we choose 0.8 and 0.5. One interesting thing is that when *fidelity* = 0.5 the imitation process as shown in Algorithm 1 performs pretty much the same as a random guessing. This is because a plasticity ‘?’ now, on average, has 50 percent of being correct as ‘1’, or incorrect as ‘0’. Thus, it is highly expected that the behavior of social learning when *fidelity* = 0.5 is quite similar to that of individual learning alone as in H&N’s simulation.

In Figure 4 and 5, it can be observed that the higher the fidelity, the higher the plastic allele, the less the amount of ‘1’, the higher the average fitness, and vice versa.

Particularly, when *fidelity* = 0.5, there is little difference in performance between EVO+IL+SL and EVO+IL in all criteria. The results obtained here are as expected and also consistent with what we have argued so far.

An explanation for this can be through the cost-benefit of social learning. When the fidelity is high, a potential agent by imitation tends to spend less learning effort than it does by trial-and-error. This leads to the fact that an agent having more plastic alleles has a higher average fitness. The selection process will favor that kind of plastic allele over others.

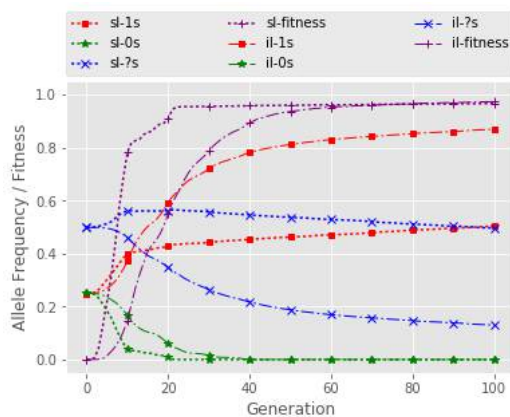


Figure 4: EVO+IL+SL vs EVO+IL. Fidelity = 0.8.

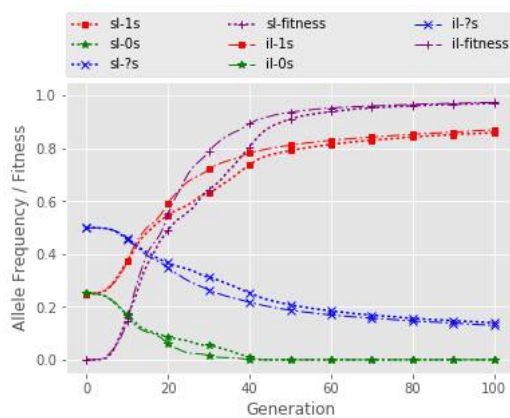


Figure 5: EVO+IL+SL vs EVO+IL. Fidelity = 0.5.

When the fidelity decreases, an observer has more chances of not copying correct values from the demonstrator. This means some plasticity ‘?’ results in higher chance of being incorrect (having the value of 0). Now having more plasticity ‘?’ means having more possibility of being incorrect. This also means that each plastic value in this case requires more learning effort to find the correct value of 1. Thus, having fewer number of ‘?’ reduces the learning cost. Again, the selection process will favor a correct individual with less learning cost, the allele ‘?’ will be less favored when the fidelity is lower.

From all of the observation and analyses above, we can conclude that information fidelity plays an important role in how social heredity directs evolution.

Conclusion and Further Discussion

In this paper, I have reconsidered the Baldwin Effect in both theoretical and empirical (computational) points of view. By briefly discussing the literature of interest, I have shown that Baldwin did not restrict the effect to genetic assimilation – which has mostly been used to understand the Baldwin Ef-

fect for many years in trans-disciplinary discourse, including in computational studies. What is implied here is that the Baldwin Effect should not be conflated with the idea of genetic assimilation, instead genetic assimilation may just be one of the ways through which the Baldwin Effect may occur. Social heredity has also been shown to play an important role in directing evolution.

Experimental results support what has been theorised. Through a specific landscape and parameter settings, it has been empirically shown that without individual learning, social heredity shows no “effect” at all. This shows that the adaptive behaviour should exist first, before social heredity takes place. When coupled with individual learning, social heredity via social learning can direct evolution in different ways depending on the fidelity of the cultural transmission. When the fidelity is high, plasticity is promoted more than the assimilation of acquired characters; yet when fidelity goes down, more assimilation emerges.

Here and now I would like to pose a question that why we should be, and keep being, interested in the Baldwin Effect. It seems that this question should have been mentioned earlier. Yet I think that only after we have presented and explained the effect in Baldwin’s original spirit and how it differs from what has often been understood, it is less uncertainty to talk about what the original Baldwin Effect, or the Baldwin Effect, would imply. One plausible reason, to me, is that the effect, if happens, helps explain why and how evolution can be directed by intelligent faculties which are also the products of evolution. This stresses the role and importance of intelligence, mind, behaviour, or any form of ontogenetic development in evolution. This also means there are circumstances in which the phenotype is not just the passive product of the gene and environment, but plays an active role in directing the evolutionary pathway of the species. The Baldwin Effect, I think, implies a **reciprocal causation** in evolution that phylogeny and ontogeny should be considered both causes and consequences. This line of thought can change the way we understand and explain evolution in biological, cultural, and even *artificial* worlds.

In the modern-day discussion of evolution there has been a call for an extension and expansion of Darwinian account of evolution via the modern synthesis (Pigliucci (2007)). Proponents of the extended evolutionary synthesis also stress the constructive role of the organism, or what we call niche construction (Odling-Smee et al. (2003)), and its reciprocal causation in evolution. This research programme has raised serious questions about the *reductionist* approach dominant in the modern synthesis, saying that not everything can be reduced to the gene (Laland et al. (2015)). Interestingly, what I have argued so far tells us that Baldwin’s legacy seemed to prepare a new movement for Darwinian evolution more than 100 years ago, yet was largely neglected in evolutionary discourse for a couple of reasons in the 20th century.

For another reason, I believe the Baldwin Effect could

contribute to the explanatory repertoire of the evolution of intelligent faculties in animals, including the human mind. Baldwin (1909) once tried to link the explanatory repertoire between disciplines, from evolutionary biology to psychology to the humanities, through his ideas of organic selection and social heredity. If the Baldwin Effect occurs through human cultural niche construction processes, this can help explain how the human brain evolved to be better at learning in the changing cultural world. The role of organic selection and social heredity in evolution is also believed to have a further value in explaining the evolution of gregarious habits and cooperative behaviour in social animals. Future work will delve deeper into these avenues of research.

References

- Baldwin, J. (1895). *Mental Development in the Child and the Race: Methods and Processes*. Macmillan.
- Baldwin, J. (1909). *Darwin and the Humanities*. Library of genetic science and philosophy. Review Publishing Company.
- Baldwin, J. M. (1896). A new factor in evolution. *The American Naturalist*, 30(354):441–451.
- Baldwin, J. M. (1902). *Development and evolution, including psychophysical evolution, evolution by orthoplasia, and the theory of genetic modes*. MacMillan Co.
- Dennett, D. C. (1991). *Consciousness Explained*. Back Bay Books.
- Dobzhansky, T. (1970). *Genetics of the Evolutionary Process*. Columbia University Press.
- Harvey, I. (1996). Is there another new factor in evolution? *Evol. Comput.*, 4(3):313–329.
- Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Horley, J. (2001). After the baltimore affair: James mark baldwin's life and work, 1908-1934. *History of Psychology*, 4(1):24–33.
- Huxley, J. (1942). *Evolution, the Modern Synthesis*. G. Allen & Unwin Limited.
- Laland, K. (2018). *Darwin's Unfinished Symphony: How Culture Made the Human Mind*. Princeton University Press.
- Laland, K. N., Uller, T., Feldman, M. W., Sterelny, K., Müller, G. B., Moczek, A., Jablonka, E., and Odling-Smee, J. (2015). The extended evolutionary synthesis: its structure, assumptions and predictions. *Proceedings of the Royal Society B: Biological Sciences*, 282(1813):20151019.
- Le, N. (2019). Evolving self-taught neural networks: The baldwin effect and the emergence of intelligence. In *2019 AISB Annual Convention – 10th Symposium on AI & Games*, Falmouth, UK.
- Le, N., Brabazon, A., and O'Neill, M. (2019). The evolution of self-taught neural networks in a multi-agent environment. In Kaufmann, P. and Castillo, P. A., editors, *Applications of Evolutionary Computation*, pages 457–472, Cham. Springer International Publishing.
- Lumsden, C. J. and Wilson, E. O. (2005). *Genes, Mind, and Culture*. WORLD SCIENTIFIC.
- Mayley, G. (1996). Landscapes, learning costs, and genetic assimilation. *Evolutionary Computation*, 4(3):213–234.
- Mayley, G. (1997). Guiding or hiding: Explorations into the effects of learning on the rate of evolution. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 135–144. MIT Press.
- Mayr, E. (1963). *Animal species and evolution*. Belknap Press. Belknap Press of Harvard University Press.
- Morgan, C. L. (1896). On modification and variation. *Science*, 4(99):733–740.
- Odling-Smee, F., Laland, K., Feldman, M., and Feldman, M. (2003). *Niche Construction: The Neglected Process in Evolution*. Monographs in Population Biology. Princeton University Press.
- Osborn, H. F. (1896). Ontogenetic and phylogenetic variation. *Science*, 4(100):786–789.
- Peter J. Richerson, R. B. (2005). *Not By Genes Alone: How Culture Transformed Human Evolution*. University of Chicago Press, 1 edition.
- Pigliucci, M. (2007). Do we need an extended evolutionary synthesis? *Evolution*, 61(12):2743–2749.
- Richards, R. J. (1989). *Darwin and the Emergence of Evolutionary Theories of Mind and Behavior (Science and Its Conceptual Foundations)*. Science and Its Conceptual Foundations”, University of Chicago Press.
- Santos, M., Szathmry, E., and Fontanari, J. F. (2015). Phenotypic plasticity, the baldwin effect, and the speeding up of evolution: The computational roots of an illusion. *Journal of Theoretical Biology*, 371:127 – 136.
- Shettleworth, S. J. (2004). Book review: Evolution and learning: The baldwin effect reconsidered. *Evolutionary Psychology*, 2(1):147470490400200.
- Simpson, G. G. (1953). The baldwin effect. *Evolution*, 7(2):110.
- Smith, J. M. (1987). When learning guides evolution. *Nature*, 329(6142):761–762.
- Sterelny, K. (2004). A review of evolution and learning: the baldwin effect reconsidered edited by bruce weber and david depew. *Evolution and Development*, 6(4):295–300.
- Suzuki, R. and Arita, T. (2007). Repeated occurrences of the baldwin effect can guide evolution on rugged fitness landscapes. In *2007 IEEE Symposium on Artificial Life*. IEEE.
- Waddington, C. H. (1953). Genetic assimilation of an acquired character. *Evolution*, 7(2):118–126.
- Weber, B. and Depew, D. (2003). *Evolution and Learning: The Baldwin Effect Reconsidered*. A Bradford book. MIT Press.
- Weismann, A., Parker, W., and Rönnefeldt, H. (1893). *The Germplasm: A Theory of Heredity*. Contemporary science series. Scribner's.
- West-Eberhard, M. (2003). *Developmental Plasticity and Evolution*. Developmental Plasticity and Evolution. OUP USA.

Morphogenetic Vase Forms

Andy Lomas¹

¹Department of Computing, Goldsmiths, University of London, SE14 6NW, UK
a.lomas@gold.ac.uk

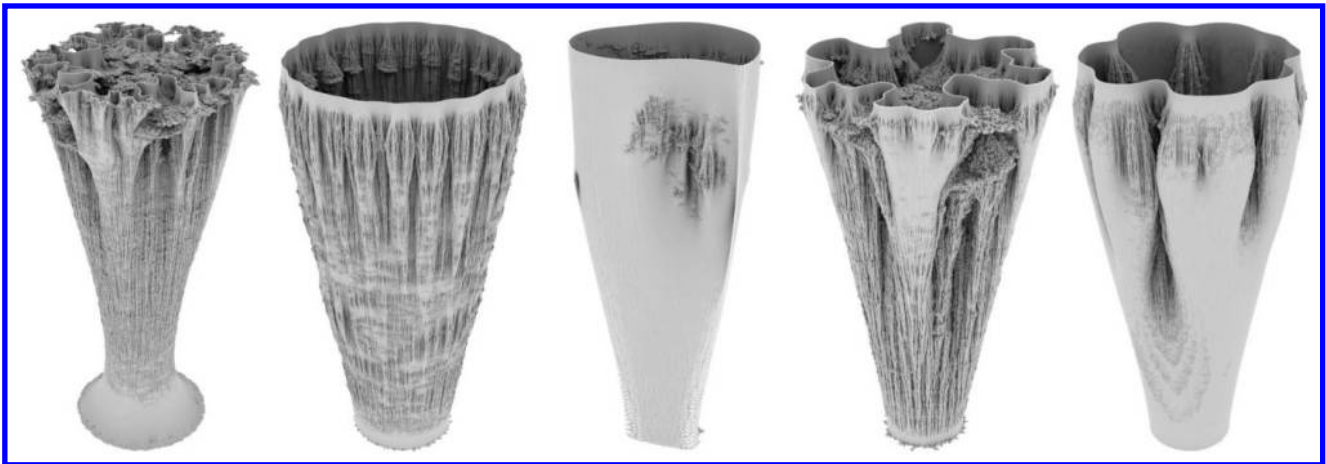


Figure 1: Examples of five Vase Forms

Abstract

This paper describes Vase Forms: a series of art works created using morphogenetic processes. A key motivation for these works was exploration of ways of working creatively with complex generative processes, such as morphogenetic systems, where the desire is to be able to influence the process in creative directions whilst achieving desired properties, such as fabricability using 3D printing, in a manner that retains rich emergence. The paper describes methods used in the creation of these works, including directly affecting morphogenetic processes using constraints and differential growth rates, combined with evolutionary search and machine learning algorithms to explore the space of possibilities afforded by the system. As well as describing the creation of Vase Forms, which have been successfully used to create sculptures, the paper looks at the closely related Mutant Vase Forms: an additional series of artworks created by accident when the system exploited bugs in the rules for the growth system resulting in unexpected but aesthetically interesting structures. These Mutant Vase Forms are not fabricable as physical sculptures with the originally intended methods, but now exist as virtual sculptures in stereoscopic installations.

Introduction

Morphogenesis and Generative Systems

Morphogenesis is a theme that has been explored by a number of artists. In 1951 Richard Hamilton curated an exhibition at the Institute of Contemporary Art (Massey, 1996) of work by a number of artists inspired by D'Arcy Thompson's 'On Growth and Form' (Thompson, 1917). In more recent years, growth has been a subject explored by computational artists including Yoichiro Kawaguchi's 'GROWTH Model' (Kawaguchi, 1983), William Latham's evolved forms (Todd and Latham, 1992) and Daniel Brown's series of digitally generated flowers (Brown, 2018). Interest by artists in morphogenesis, and D'Arcy Thompson specifically, has been sufficient for the University of Dundee to receive support from Art Fund to create a collection of artwork dedicated to this subject (University of Dundee, 2011).

One question raised by generative systems, such as those that use simulation of morphogenesis, is that of how are we to work creatively with them? In particular, how should we work with systems deliberately designed to encourage emergence: complex systems where results are intrinsically difficult to predict? There is a strong analogy with plant breeding, where we are working with a medium that is naturally rich. Through

experimentation and experience we can develop insights into what is possible and how to influence plants to develop in ways that give desired properties. We need to discover the potentialities of the system we are working with, as well as the limits of its capabilities. Which features can be independently influenced, and which are co-dependent? Whether art, design or architecture, working in this manner involves changing our relationship with the computer. Traditional top-down design methods are no longer appropriate. We need to be open to a process of exploration. Participating in a search for rich interesting behavior: selecting and influencing rather than dictating results.

Generative systems are typically based on algorithmic processes that are parametrically controlled. Given a set of parameter values the process is run to create an output. Classic examples include Conway's Game of Life (Conway, 1970) and reaction diffusion equations (Turing, 1952). Generative systems have been used by a number of artists, from pioneering early work by Algorists such as Manfred Mohr (Mohr and Rosen, 2014), Frieder Nake (Nake, 2005), Ernest Edmonds (Franco, 2017) and Paul Brown (DAM, 2009a), to more recent work by artists such as William Latham (Todd and Latham, 1992), Yoichiro Kawaguchi (DAM, 2009b), Casey Reas (Reas, 2018), and Ryoji Ikeda (Ikeda, 2018).

The most interesting systems are generally those that create emergent results: genuinely unexpectedly rich behavior that cannot be simply predicted from the constituent parts. For these systems the relationship between the input parameters and the output is often complex and non-linear, with effects such as sensitive dependence on initial conditions. This can make working with such systems particularly challenging.

Creative Exploration of Parameter Space

One problem is that of how to work with systems with large numbers of parameters. With a small number, such as two or three parameters, the space of results can be relatively easily explored by simply varying individual parameter values and plotting the effects of different combinations. One common technique is to create charts where all the parameters are sampled independently at regularly spaced values and results are plotted to show the results. What scientists would call a phase space plot. This method of parameter exploration can be effective and was used by the author for earlier work such as for his 'Aggregation' (Lomas, 2005) and 'Flow' (Lomas, 2007) series (Figure 2).

As the number of parameters increase, the number of samples needed to explore different sets of combinations using this type of method increases rapidly. This problem is commonly called the 'Curse of Dimensionality' (Bellman, 1961) (Donoho, 2000), where the number of samples that need to be taken increases exponentially with the number of parameters. One approach is to simply limit the number of parameters, but this can be at the expense of overly restricting the range of behavior the system is capable of. If we are working with richly emergent systems these problems are often further compounded: a direct consequence of complexity is that parameters that drive the system often work in difficult to comprehend, unintuitive ways. Effects are typically non-linear, often with sudden tipping points as the

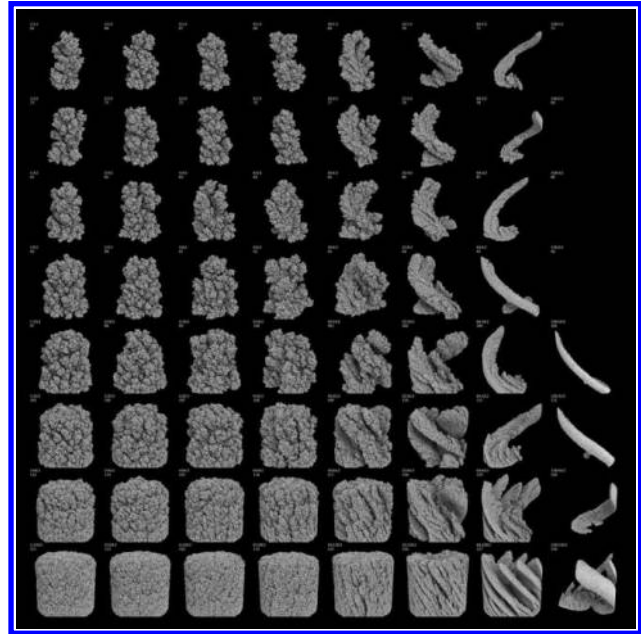


Figure 2: Phase Space plot from the Aggregation Series

system goes from one type of behavior to another. In particular, in many systems the most interesting emergent behavior occurs close to the boundary between regularity and chaos (Kauffman, 1996).

This raises the idea of working with computers not merely as a medium to generate artwork but as active collaborators in the process of exploration and discovery. The use of tools to help the process of exploration can materially change both the creative process and the complexity of systems that we can effectively work with.

One analogy is that of Advanced Chess: a form of the game where each human player can actively use a computer to assist them to explore possible moves during games (Kasparov, 2017). Computer chess programs are generally very good at quickly detecting whether a proposed move will have catastrophic results. The effect of allowing a human player to test potential moves with a computer assistant is to make the game blunder-free. By removing the stress of making easily punished mistakes the human in the collaboration is freed to approach the game in a much more actively experimental way.

Another potentially rich analogy is with fly-by-wire systems in aircraft (Sutherland, 1968), that allow designs of aircraft to be created which are inherently unstable but can perform complex maneuvers beyond the performance envelope of conventional aircraft (Stein, 2003). These include designs that would be difficult, or even impossible, for a human pilot to directly control. Through the use of digital fly-by-wire technology, where the pilot uses their controls to indicate their intent but all the data is passed through a computer before being fed to actuators on the control surfaces, such aircraft can be flown safely.

A number of authors have proposed using evolutionary methods to allow artists and designers to explore systems with large numbers of parameters. Examples include Dawkins' Biomorphs (Dawkins, 1986) and Mutator (Todd and Latham, 1992). More recent examples, that use collaborative

evolutionary interfaces for creation of images and forms, include Picbreeder (Secretan et al, 2008) and Endless Forms (Clune and Lipson, 2011). A number of systems that use interactive evolutionary computation for art and design are described in (Bentley, 1999) and (Takagi, 2001).

As demonstrated by natural processes, evolutionary methods can be effective even with extremely large numbers of parameters. One problem, though, can be that these methods generally lead to exploring a small number of paths within the space of available possibilities. The nature of these types of methods are to bias the search towards the most successful areas of the parameter space that have already been highly sampled. New samples are taken by mutation or cross-breeding of the gene codes from previous samples that are deemed fittest according to a specified fitness function. This means that previously highly sampled areas are likely to be even more highly sampled in the future as long as they contain 'fit' individuals. This is a good strategy for exploiting the best results that have been previously found, but is potentially a bad strategy for actively finding novel solutions which may be in areas of the landscape that have been very sparsely sampled.

In recent years there have been a number of studies into methods to keep diversity when working with evolutionary techniques, such as Novelty Search with Local Competition (Lehman and Stanley, 2011) and MAP-Elites (Mouret and Clune, 2015). These methods generally require the defining of a domain specific feature vector that represents the behavior of the system to enable a meaningful measure of the distance between individuals in behavior space to be calculated. The creation of such a function to represent behavior is often not easy (Lehman and Stanley, 2008).

A number of authors have proposed using machine learning techniques to assist human designers. In general these are for domain specific applications, such as for architectural space frame structures (Hanna, 2007), structurally valid furniture (Umetani et al. 2012) or aircraft designs (Oberhauser et al. 2015). In these systems, machine learning is typically used to learn about specific properties of the system. This is then used to provide interactive feedback for the user about whether an object designed by them is likely to have desired properties, such as being structurally feasible, without having to do computationally prohibitive tasks such as evaluation of structural strength using finite element analysis.

One thing that needs consideration is that creative work with generative systems often needs different phases of exploration, with the intent of the artist or designer changing over time. Initially they may be actively experimenting: trying to get a feel for the capabilities of the system they are working with. Once they have done some initial experiments they may want to continue to explore broadly, but with a general focus on regions that seem to have promise. When some particularly interesting results have been found they may wish to further refine them into presentable artefacts, or want to switch to actively looking for novel results that are significantly different to those they have found so far. These considerations mean that if a computer is being used to assist them explore the space of possibilities they may want it to work in different ways depending on their current intent.

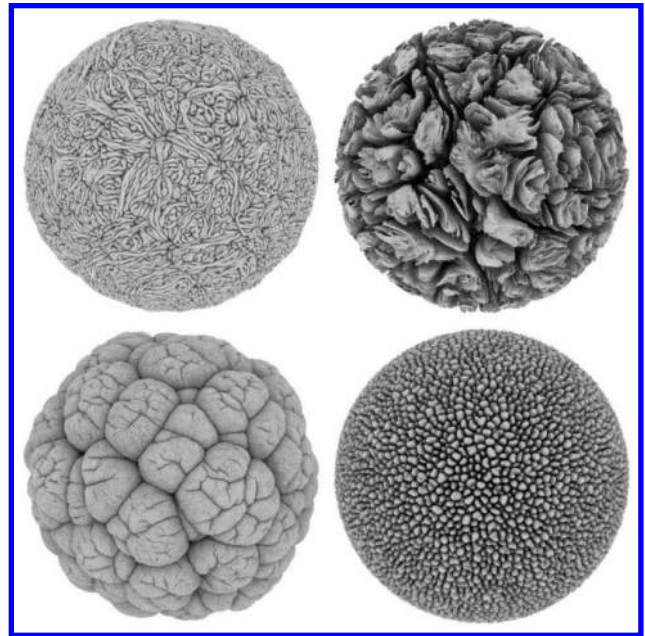


Figure 3: Examples of Cellular Forms

Motivation

The author is a practicing computational artist, whose work explores how complex organic forms can be created through digital simulation of morphogenetic processes. Inspired by Alan Turing's use of simple equations to create rich self-organizing patterns (Turing, 1952), the author's work focuses on creating simplified models of growth at the level of individual cells and exploring the emergent forms that can be created for these low-level rules (Lomas, 2014) (Figure 3). To explore the space of possibilities the author uses a hybrid system that combines several techniques, including evolutionary design search methods and lazy machine learning, to discover and fine-tune parameter combinations that appear to create particularly interesting results (Lomas, 2016).

The motivation behind the work described in this paper was how the cellular growth system used for works such as Cellular Forms (Lomas, 2014) could be modified to direct it towards the more specific goal of creating three dimensional structures that could be turned into physical sculptures. In particular, it was desired that the forms could be physically realized using computationally controlled additive fabrication techniques such as 3D printing using fused filament fabrication, or at larger scales using robots that deposit sequential layers of molten material. It was also desired that the sculptures created could potentially be suitable for use as the supports for tables, so should have flat bases and tops and potentially be strong enough to be load bearing.

One of the main restrictions with fused filament fabrication is that every layer to be printed needs to be supported by the previous layers. This means that overhang areas in the structures have to be within maximum size and angular limits or additional support structures have to be printed and removed after fabrication. The aim in this work was to explore

how the structural needs for physical fabrication could be combined with aesthetic goals when working with a morphogenetic system to create forms that, though completely synthetic, exhibit complex detailed organic structures such as are typically found in natural forms.

Methods

Cellular Growth System

The generative system for this work was based on the model of growth using cellular division that the author created for his Cellular Forms work (Lomas, 2014). This system uses a simple particle spring model (Reeves, 1983), with each cell represented by particles with links to a number of other connected cells. The topology of connection between cells means that they form a sheet-like structure embedded in three-dimensional space, with the sheet free to fold into complex geometrical shapes. Interactions between connected cells are implemented using spring-like rules. The key elements of the system are:

- Rules for the generation of food, with rates for the spontaneous generation of food by all cells, and for the generation of food by simulated photosynthesis by firing light rays at cells and generating food in each cell proportional to the number of rays that hits it.
- Rules for whether cells are 'greedy' and directly accumulate the food they generate themselves or are 'cooperative' and share the food they generate with the other cells they are connected to.
- A threshold for how much food a cell needs to accumulate before it is selected for dividing into two cells.
- Rules for the direction that a cell splits in, which can be influenced by factors such as the curvature of the sheet of cells, the local direction of most tension in the surface, or in a randomly selected direction.
- Forces that try to maintain a constant rest-length separation between linked cells.
- Forces that tend to make the sheet of cells bulge outwards if that local area of the sheet is in compression (cells packed together closer than the rest-length).
- A relaxation rule that tends to move cells towards the average of their neighbors.

The system aims to be sufficiently simple to enable the simulation of morphogenesis to be implemented using massively parallel processing on consumer level graphics processing units, with simulations that can scale to tens of millions of cells and tens of thousands of simulation steps using conventional PC hardware. The code for the simulation engine is implemented in C++ and CUDA. For more details of the algorithms and implementation see (Lomas, 2014).

Additional Constraints and Influences

For this work the cellular division system previously described was augmented with a number of additional constraints and influences to steer the growth system to create structures with desired properties.

As previously described, the aim was to create forms that could be fabricated as sculptures that could also potentially be

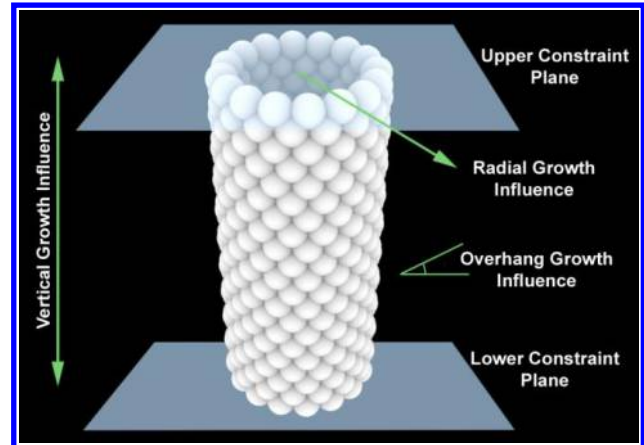


Figure 4: Initial configuration, constraints and influences

used as supports for tables. With this in mind, a decision was made to create forms with the topology of a tubular structure with open edges at the top and bottom. The initial configuration of cells was a simple cylinder (Figure 4).

To make the top and bottom of the structure stay flat, so that the form should stand on the ground and to potentially support a table top, the cells along the open edges were constrained to horizontal planes so that these cells were only allowed to move in two dimensions within those planes. The constraining planes were maintained at a constant separation distance from each other. When cells divide, a test was run to check whether the resulting cells were in one of the open edges of the sheet. If so, they were added to the sets of constrained cells.

Since the horizontal constraining planes were kept at a constant separation distance, as the cells in the sheet grow there could be a tendency to over crowd the space between the planes if the rest-length between cells was kept constant. To prevent this from happening a rule was added to the system which adaptively reduced the rest-length between cells by a constant factor each time step. The value of this factor was one of the parameters used to drive the system.

Finally, a number of differential growth rates were added to the system which affect the rate of cell division in different parts of the structure. An analogy can be made with controlling the growth of plants by the selective application of nutrient in certain areas and a growth retardant in others. Differential growth rates were implemented by modifying the rate at which food is generated and accumulated in cells, hence controlling the rate at which they divide. Three different factors were allowed to affect growth rates:

Vertical Growth Influence. This allowed the position of the cell along the vertical axis to affect the growth rate, with cells closer to the top of the structure growing faster than cells lower down. The aim was to encourage the formation of generally vase-like forms that are wider at the top than the bottom.

Radial Growth Influence. To stop structures growing too large horizontally, the radial distance from the central vertical axis was also allowed to affect growth rates, retarding the growth of cells further away from this axis.

Overhang Growth Influence. As previously described, one of the main limitations of using fused filament fabrication is that structures can only have limited overhangs if they are to be made without the generation of additional support structures. To try to naturally encourage the growth of forms within overhang limits the local angle between the sheet of the cells and the vertical axis was also allowed to affect the growth rate, reducing the rate of growth if it exceeded a threshold angle.

This set of influences using differential growth rates was selected to work with the requirements of this specific system, but this approach should be suitable to be generalized to give creative influence over a number of other similarly morphogenetically based generative systems.

Parameter Selection

With the addition of these constraints and influences the simulation system for generating Vase Forms had 29 parameters, each of which could be set independently. As described in the introduction, having a system with this number of parameters raises challenges of how to explore the space of possibilities of the system in order to find parameter combinations that produce desired results.

In response to these issues, the author has developed a program called Species Explorer (Figure 5) to assist the process of generating parameter values to be used with generative systems. The initial requirement for such a system came from the number of parameters that the author found he needed when he was developing the simulation engine for his Cellular Forms work (Lomas, 2014), but is designed to work in general with systems driven by a fixed number of parameters, and provides a framework for various methods to be used to assist in exploring the landscape of possibilities.

The software provides an interface for the user to specify the programs that need to be run to generate each individual. Once a set of parameter values has been chosen the system writes out a ‘creation script’ (Linux shell script, Windows batch file or Python script) that can be executed on the computer to run the generative system with the specified values. Once an individual has been generated the user can then use the interface to rate and categorize the results.

The software allows the user to select from a variety of ‘creation methods’ each of which use different techniques to generate sets of new parameter values to try. Examples of the creation methods the user can select from include:

- Simple random selection of parameter values from uniform distributions within a specified range.
- Evolutionary search methods using mutation and cross-breeding between the parameters used for previous individuals.
- A ‘fitness landscape’ method where parameter values are selected using lazy machine learning to estimate how the user would rate and categorize individuals at new coordinates in parameter space. The system implements two different options for lazy machine learning: nearest k-neighbors and interpolation using radial basis functions.

Using the interface, different creation methods can be used for each generation and fitness functions (such as for use with evolutionary search or lazy machine learning) can be

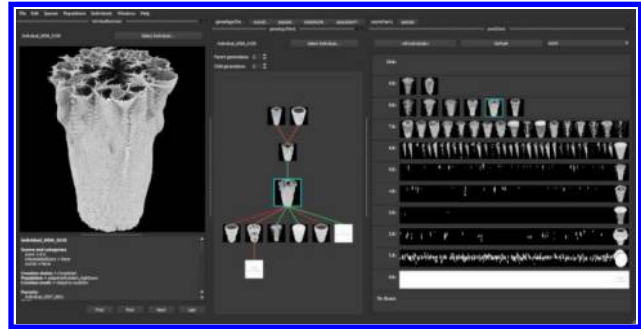


Figure 5: Species Explorer user interface

customized using a simple Python based expression syntax. One feature of these expressions is that they can include the distance in parameter space to the nearest previous sample that has already been taken, which allows a simple implementation of methods to maintain diversity in the genotype space.

The use of a variety of creation methods provides the flexibility to allow the user to explore the space of possibilities in different ways depending on their intent (such as focused refinement based on some previous samples, or an active exploration for potentially novel results). The software also provides a framework for plugins to implement new creation methods, so the user can specify custom ways for how parameters for new individuals are chosen. For more technical detail about Species Explorer see (Lomas, 2016).

Results

Mutant Vase Forms

The initial results from the system were genuinely unexpected: instead of creating structures that were likely to be fabricable using 3D printing, the system would often create forms with finely detailed approximately horizontally-oriented branching structures (Figure 6). This was the exact opposite of the type of structure the author was hoping to create by differentially adjusting growth rates to slow down cell division in regions with high overhang angles.

Analyzing the results led to the realization that these unexpected results were due to bugs in how the system had

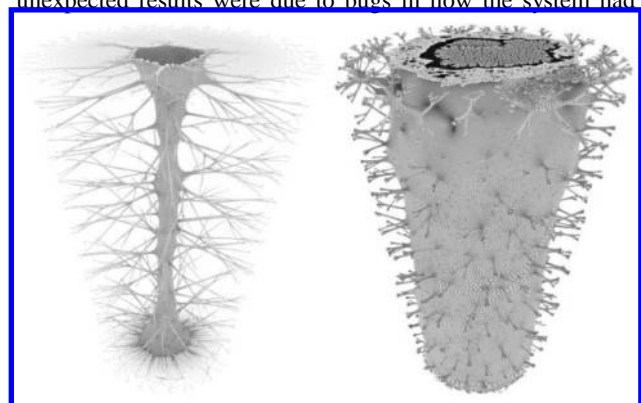


Figure 6: Mutant Vase Forms

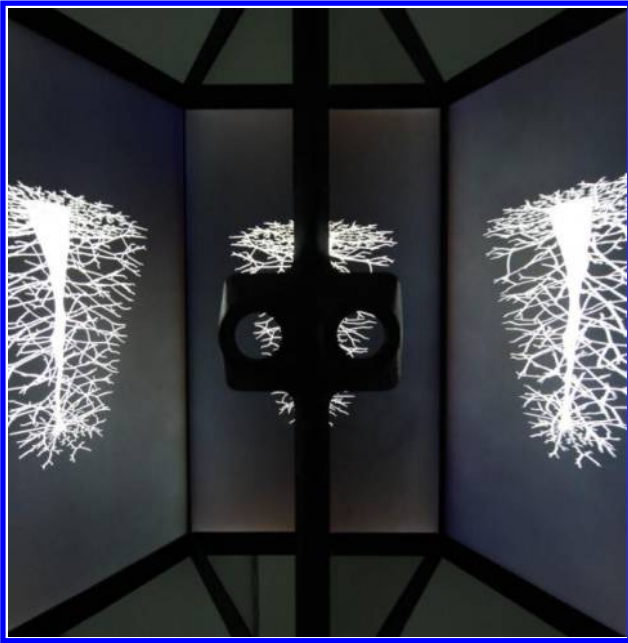


Figure 7: Mutant Vase Forms Stereoscopic Installation

been implemented:

- As previously described, during the growth simulation the rest-length between cells was reduced by a constant factor each time step. However, in the code for generating food by light rays hitting cells, the spheres used to represent the cells were being left at their initial size instead of using the modified rest-length to adjust their radius. The result was that the cells were effectively a lot larger than they should have been, meaning that cells that stuck out further than their neighbors were struck by all the light rays and became the places that all the food from photosynthesis was generated.
- The code for differentially adjusting growth rates was initially implemented by affecting the size of time steps, which was also affecting the physics simulations governing how the cells move. This meant that areas of low growth were also areas where structures became ‘frozen’, so if a region developed an overhang this feature would become geometrically fixed position-wise in space.

The effect of these two bugs was to accidentally create a recipe for generating horizontally oriented structures with growth focused at the tips. Though this wasn’t what was originally intended, the author considered the results to be surprisingly aesthetic, particularly when they create structures with multiple filigree branches. The results can also be seen as genuinely emergent: they are the consequence of bugs in a system which has sufficient complexity so that changes to the rules can lead to rich unexpected consequences.

Though the resulting forms were not suitable for 3D printing using fused filament fabrication, the author considered that the results were sufficiently interesting to make them into a series of artworks in their own right: the ‘Mutant Vase Forms’. These have been exhibited using stereoscopic installations so that they can be experienced as animated three-dimensional forms (Figure 7) (Lomas, 2017).

Vase Forms

After investigating the reasons behind the initial unexpected results, the simulation code was modified to fix the bugs previously described. This resulted in the generation of forms with more expected properties: generally vase like structures, broader nearer the top, narrow at the base, and with flat regions at both the top and the base (Figure 1). Though the forms often have overhangs, these are typically within angular limits or are of a sufficiently small size to allow 3D printing using fused filament fabrication methods without the need for additional support structures.

The generated Vase Forms appear to exhibit an interesting range of morphologies, with structures reminiscent of coral and plant-like forms. Structures often have complex ridges and folds, which as well as being aesthetic have the potential to have useful structural performances. There are also often structures that have surprised the author, such as the spontaneous generation of canopy-like structures at certain height ranges, which are probably the results of the differential growth rates. Many forms exhibit a variety of different surface patterns, from regions where the surfaces are relatively smooth, to other sections where the sheet of cells is folded into complex structures.

The author has printed a number of Vase Forms using an Ultimaker 2+ 3D printer, with conventional PLA filament. For exhibition the works have been presented as a combination of 40cm high final sculptures (which needed to be printed in two parts due to the build volume restrictions of the Ultimaker 2+) together with series of smaller 20cm high maquettes that show a number of different stages of development of each form from its initial configuration of a small number of cells in a cylindrical shape to the final structure with several million cells (Figure 8). These ‘developmental series’ can be seen as echoing the models illustrating embryo development that are commonly seen in natural history museums. Typical final forms used for fabrication have between 5 million and 20 million cells.

As well as the sculptures 3D printed using fused filament fabrication, the author has been able to fabricate a larger 60cm high form in polyamide using selective laser sintering. This is a process that requires less constraints on the shapes that can be fabricated due to overhang limits or the need for support structures.

The author has also used computer rendered image files from data at different timesteps during the growth simulation to create animations showing the process of forms growing by cell division (Lomas, 2018). In a number of exhibitions these animations have been shown together with the physical sculptures (Figure 9), giving another view into the story of how simulation of morphogenesis was used to create the forms.

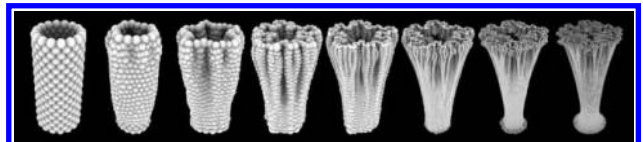


Figure 8: Vase Forms Developmental Series

Discussion

The creation of Vase Forms, and the accidental creation of the closely related Mutant Vase Forms, has been an exploration in trying to take an unruly, but potentially interesting, system of generating structure using morphogenesis and steer it in directions with both aesthetic and functional goals. The combination of using constraints and specific modifications to the growth rules, together with the use of evolutionary and machine learning methods to discover parameter combinations that give desired results, appears well suited to working with morphogenetic systems.

The work can be seen as a case study of engaging with rich emergent systems, emulating the way that nature works through evolution and natural selection but with a design intent. The use of influences that affect growth rates in different parts of the structure were designed for the specific needs of the Vase Forms, but should be generalizable to give a degree of creative influence over other similar morphogenetic systems.

The discovery of parameter combinations to create the final exhibited forms used a hybrid set of methods including interactive evolutionary computing and machine learning. One thing that needs to be considered for creative tools is the different needs of artists or designers that are committed to an extended process of exploration, but whose intention changes during the course of the development of a work, and more casual users, such as gallery visitors or visitors to a website, who are probably only going to engage with a system for a

limited amount of time and for whom a simple interface that offers a single mode of interaction is probably more suited. For an artist or designer developing their own work, issues of user fatigue and perceived loss of control can be important, but we can also assume an extended commitment over time. Having a range of customizable tools that allow the user to direct the exploration with different intents can be important.

Conclusion

The aim with this work was to create sculptural forms that could be fabricated without the need for extensive support structures, while avoiding overly constraining the system and losing the potential for rich emergence. This appears to have been successful, creating vase-like structures that have a surprisingly natural appearance even though they have been completely synthetically generated and fabricated. The project can be seen as having genuinely emergent results, particularly with the Mutant Vase Forms where bugs in the algorithm used to generate structures resulted in completely unexpected, but aesthetically interesting, consequences.

The Vase Forms have been well received, including an invitation to feature in a special exhibit at the Victoria and Albert Museum for the 2018 London Design Festival (London Design Festival, 2018). The work has also been shown together their closely related siblings, the Mutant Vase Forms, in a number of exhibitions, including 'bubble, bulge, bleb' (LifeSpace, 2017), an exhibition celebrating the centenary of



Figure 9: Vase forms exhibit for the London Design Festival at the V&A Museum, London

the publication of D'Arcy Thompson's 'On Growth and Form'.

With the advent of new techniques, such as multi-material fabrication and the control of structure at microscopic levels, morphogenetic systems that can produce rich continually varying complex patterns and forms have the potential to contribute novel solutions beyond those that can be created by conventional assembly out of discrete parts. The question of how to appropriately steer morphogenetic systems, allowing humans to creatively engage with them, whilst keeping rich emergent behavior deserves further study. Working with such systems using a combination of 'hard constraints' and 'soft influences', such as differentially influencing growth rates, appears to be fruitful.

Acknowledgments

I would like to acknowledge the help of Alisa Andrasek and Daghan Cam at UCL The Bartlett School of Architecture, with whom discussions about creation of structures suitable to act as table supports was an initial inspiration for this work. I would also like to acknowledge the support from Materialise in fabricating a Vase Form using selective laser sintering.

References

- Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton: Princeton University Press.
- Bentley, P. J. (1999). *Evolutionary Design by Computers*. San Francisco: Morgan Kaufmann.
- Brown, D. (2018). *Daniel Brown's*. Available at <http://danielbrowns.com/>
- Conway, J. H. (1970). The game of life. *Scientific American* 223: 4.
- Clune, J., and Lipson, H. (2011). Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *ECAL 2011*, pages 141-148.
- DAM, (2009a). *Digital Art Museum: Paul Brown*. Available online: <http://dam.org/artists/phase-one/paul-brown>
- DAM, (2009b). *Digital Art Museum: Yoichiro Kawaguchi*. Available online: <http://dam.org/artists/phase-one/yoichiro-kawaguchi>
- Dawkins, R. (1986). *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*. New York: WW Norton & Company.
- Donoho, D. L. (2000). High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Paper presented at *AMS Math Challenges Lecture*, Los Angeles, CA, USA, August 6-11.
- Franco, F. (2017). *Generative Systems Art: The Work of Ernest Edmonds*. Abingdon: Routledge.
- Hanna, S. (2007). Inductive machine learning of optimal modular structures: Estimating solutions using support vector machines. *AI EDAM* 21: 351-66.
- Ikeda, R. (2018). *Ryoji Ikeda*. Available online: <http://www.ryojiikeda.com/>
- Kasparov, G. (2017). *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. London: John Murray.
- Kauffman, S. (1996). *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*. London: Penguin.
- Kawaguchi, Y. (1983). *Yoichiro Kawaguchi: Growth*. Available online: <http://coppergloth.net/siggraph/art.php?id=71>
- Lehman, J., and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE 2008*, pages 329-336.
- Lehman, J., and Stanley, K. O. (2011). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211-218. ACM.
- LifeSpace, (2017). *bubble, bulge, bleb*. Available online: <http://lifespace.dundee.ac.uk/exhibition/bubble-bulge-bleb>
- Lomas, A. (2005). Aggregation: Complexity out of Simplicity. In *ACM SIGGRAPH 2005 Sketches*. New York: ACM.
- Lomas, A. (2007). *Flow*. Available online: <http://www.andylomas.com/flow.html>
- Lomas, A. (2014). Cellular Forms: An Artistic Exploration of Morphogenesis. Paper presented at *AISB-50*, London, UK.
- Lomas, A. (2016). Species Explorer: An interface for artistic exploration of multi-dimensional parameter spaces. In *EVA London 2016*, Electronic Workshops in Computing (eWiC). London: BCS.
- Lomas, A. (2017). *Mutant Vase Forms: New Growth and Light Rays*. Available online: <https://vimeo.com/321525568>
- Lomas, A. (2018). *Vase Forms*. Available online: <https://vimeo.com/321534705>
- London Design Festival (2018). *Vase Forms*. Available online: <https://www.londondesignfestival.com/event/vase-forms-0>
- Massey, A. (1996). *The Independent Group: Modernism and Mass Culture in Britain, 1949-59*, pages 42-45. Manchester University Press.
- Mohr, M. and Rosen, M. (2014). *Der Algorithmus des Manfred Mohr: Texte 1963-1979*. Oakland: Spectator Books.
- Mouret, J. B., and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.
- Oberhauser, M., Sky S., Thomas G. and Kristina S. (2015). Computational Design Synthesis of Aircraft Configurations with Shape Grammars. In *Design Computing and Cognition '14*, pages 21-39. Springer, Cham.
- Nake, F. (2005). Computer art: A personal recollection. In *Proceedings of the 5th Conference on Creativity & Cognition*, pages 54-62. New York: ACM.
- Reas, C. (2018). *Home Page of Casey REAS*. Available online: <http://reas.com/>
- Reeves, W. T. (1983). "Particle Systems: A Technique for Modelling a Class of Fuzzy Object", *ACM Transactions on Graphics, Volume 2 Issue 2*, April 1983: 91-108.
- Secretan, J., Beato, N., D Ambrosio, D. B., Rodriguez, A., Campbell, A., and Stanley, K. O. (2008). Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1759-1768. ACM.
- Stein, G. (2003). Respect the unstable. *IEEE Control Systems* 23: 12-25.
- Sutherland, J. P. (1968). *Fly-by-Wire Flight Control Systems*. Dayton: Air Force Flight Dynamics Lab Wright-Patterson AFB.
- Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. In *Proceedings of the IEEE*, 89(9): 1275-1296.
- Thompson, D. W. (1917). *On Growth and Form*. Cambridge University Press.
- Todd, S. and Latham, W. (1992). *Evolutionary Art and Computers*. London: Academic Press.
- Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 237: 37-72.
- Umetani, N., Igarashi, T. and Mitra, N. J. (2012). Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics* 31: 86-1.
- University of Dundee, (2011). *D'Arcy Thompson Zoology Museum Art Collection*. Available online: <https://www.dundee.ac.uk/museum/collections/zoology/art>

Spatial Representation of Self and Other by Superposition Neural Network Model

Wataru Noguchi¹, Hiroyuki Iizuka¹, Shigeru Taguchi² and Masahito Yamamoto¹

¹Graduate School of Information Science and Technology, Hokkaido University, Japan

²Graduate School of Letters, Hokkaido University, Japan
noguchi@complex.ist.hokudai.ac.jp

Introduction

The sense of self-location is necessary for individuals to perform adaptive behavior like navigation in the environment. The evidence of the sense of self-location is identified as the place cells which was firstly found in the hippocampus of rats' brain (O'Keefe and Dostrovsky, 1971). Recently, it was reported that the locations of other individuals are also represented in the hippocampus of bats and rats (Danjo et al., 2018; Omer et al., 2018). Even some hippocampus cells represent their location without distinction between self and other, and these cells can be considered to be related to empathy mechanisms similar to the mirror neurons (Rizzolatti and Sinigaglia, 2016). However, how such spatial representation that associates self and other's locations is developed is not understood as well as the mirror neurons. One of the popular explanation of representation of the other in self is simulation theory that individuals internally simulate the other's internal recognition. However, the simulation of the other requires another module for inferring the other's internal recognition, which is called model of others. Such simulation module should be different from the module that deal with self internal recognition, and the place cells shared between self and other cannot emerge.

We considered the other's location is represented in the same module as self location rather than in the different module that simulates other's internal recognition. To realize shared representation of self and other's locations like found by (Danjo et al., 2018; Omer et al., 2018), we propose superposition mechanism that two different representations are parallelly processed by the same module at the same time. We implement the superposition mechanism by using two same modules in our proposed neural network model as described later. Through the prediction learning of subjective vision of simulated mobile agent where another agent existed, our proposed network developed shared representation of self and other agent's locations.

Network Model and Simulation

Our proposed network model is constructed to receive subjective vision of self agent \mathbf{v}_t^{self} and self motion \mathbf{m}_t^{self} and

trained to correctly predict future vision \mathbf{v}_{t+1}^{self} . The network parallelly processes two different representations for self and other through the same module, which we call the superposition module, as follows:

$$\mathbf{h}_t^{self} = \phi(\psi^{self}(\mathbf{v}_t^{self}), \mathbf{m}_t^{self}, \mathbf{h}_{t-1}^{self}), \quad (1)$$

$$\mathbf{h}_t^{other} = \phi(\psi^{other}(\mathbf{v}_t^{self}), \mathbf{m}_t^{other}, \mathbf{h}_{t-1}^{other}), \quad (2)$$

where ϕ is the function of superposition module, \mathbf{h}_t^{self} and \mathbf{h}_t^{other} are internal states of the superposition module, and ψ^{self} and ψ^{other} are visual encoders, which are different modules. The processes represented by Eq. (1) and (2) are conducted at the same time, and it is considered the two different states are superposed in the sense that the single superposition module has these two states at the same time. The two superposed states have an explicit difference on the motion input; the self motion \mathbf{m}_t^{self} is available, on the other hand, the other's motion \mathbf{m}_t^{other} is not available and assumed to be always zero; it means that \mathbf{h}_t^{self} becomes self-related representation and \mathbf{h}_t^{other} becomes other-related representation. Then, the network generates the prediction of self vision as follows:

$$\mathbf{v}_{t+1}^{self} = \hat{\psi}(\mathbf{h}_t^{self}, \mathbf{h}_t^{other}), \quad (3)$$

where $\hat{\psi}$ is a visual predictor module. In this prediction process, the superposed states are used as two different inputs and integrated for visual prediction. The whole structure of our proposed network is shown in Fig. 1 (a). The superposition module ϕ consists of LSTM (long short-term memory) (Hochreiter and Schmidhuber, 1997), which is an RNN (recurrent neural network) with gate structures, and a fully connected layer as the motion encoder. We implemented the superposition module by using two LSTM modules with same network weights, and we call these two LSTMs the self and other LSTMs for later description, although they are the same module. The visual encoders ψ^{self} and ψ^{other} and visual predictor $\hat{\psi}$ consist of CNN (convolutional neural network).

The network was trained on the visuomotor sensory sequences of a mobile agent in a simulated environment shown in Fig. 1 (b, c). There are two agents as self and other, and four colored boxes as visual landmarks. The self agent

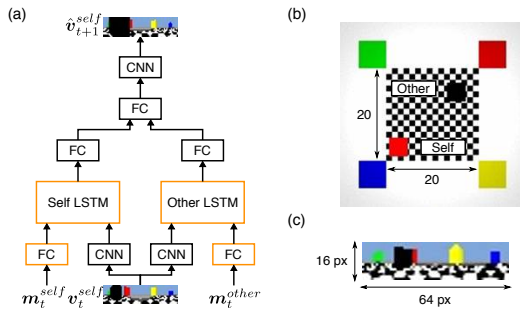


Figure 1: (a) The schematic view of the proposed network model. (b) Overview of the simulation environment. The self agent moves with omni wheels. (c) An example of self agent’s vision, which is omni-directional vision.

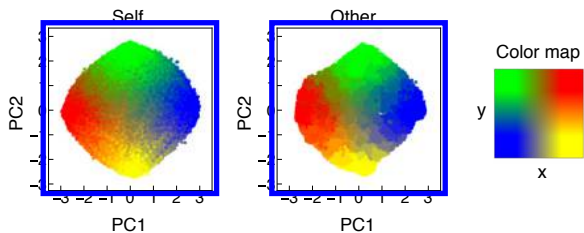


Figure 2: The visualization of the internal states of the self and other LSTMs (left and middle). The states are colored according to the agents’ location where colors are assigned for each location as shown in the color map (right).

moves around the environment, on the other hand, the other agent does not move. By relocating the other every trial, the network is trained in various placements of the other agent. We previously showed that the prediction learning of subjective vision realizes self-organization of the representation of self-location in the internal states of an RNN. In the current study, we expected that the development of shared representation of self and other’s locations is realized by introducing the superposition module.

Results

After the training, we visualized the internal states of two LSTMs on two-dimensional space by using PCA (principle component analysis) when the agent was moving around (Fig. 2). The visualized internal states are colored according to the self and other agent’s locations for self and other LSTM’s states, respectively. The same PC space was used for mapping the internal states. It is shown that the internal states are arranged according to actual agents’ location and it is considered that the network developed the representation of self and other locations. It should be noted that the self and other’s locations seemed to be represented on the same region of the internal state space of self and other LSTMs. Then, we constructed a linear regression model that predict the location of the self agent from the self LSTM’s internal states h_t^{self} ; then predict the location of the other agent from the other LSTM’s internal states h_t^{other} using the model constructed for predicting self agent’s location. As

a result, the error distance for self agent’s location became 0.40 and that for other agent’s location became 1.38. The error is small considering that theoretical expected distance between two random points sampled from a square size of 20×20 is about 10.4, and it is considered that the superposition module represented the self and other’s locations in a shared representation. There are no constraints to make the correspondence of the locations and internal states the same; however, our results show that shared representation of the self and other’s location can be developed in the network where the same module represent two different states at the same time.

Discussion

Although it is possible to develop representations shared between two different modules, i.e., not a single superposition module, some learning mechanisms or constraints are necessary to have shared representation for the different modules in addition to developing the representation of self and other’s location. On the other hand, our proposed network processes self and other’s representation in the same way. In such superposition network, the other can be recognized by applying the self model to the other without constructing the other’s model. Such superposing of self and other’s model is more efficient than separate development of self and other’s model; consequently, the shared representations of self and other’s location were developed without any additional constraints. Although such superposition structure with two separate modules with same weights in our implementation has not been found in biological systems, we considered that parallel processing of the self and other’s representation in the same network module is necessary for developing the shared representation of the self and other’s locations.

Acknowledgements

The authors would like to thank Prof. Masanori Takezawa and Prof. Kenji Ogawa for useful discussions. This work was supported by JSPS KAKENHI Grant Number JP18J20404.

References

- Danjo, T., Toyozumi, T., and Fujisawa, S. (2018). Spatial representations of self and other in the hippocampus. *Science*, 359(6372):213–218.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- O’Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175.
- Omer, D. B., Maimon, S. R., Las, L., and Ulanovsky, N. (2018). Social place-cells in the bat hippocampus. *Science*, 359(6372):218–224.
- Rizzolatti, G. and Sinigaglia, C. (2016). The mirror mechanism: a basic principle of brain function. *Nature Reviews Neuroscience*, 17(12):757.

Artificial Ant Colonies for Adaptive Rewards in Serious Games

Yann Semet¹, Bruno Marcon¹, Konstantinos Demestichas², Nikos Koutsouris² and Antonio Ascolese³

¹Thales Research and Technology, Palaiseau, France

²National Technical University of Athens, Athens, Greece

³Imaginary srl, Milano, Italy

yann.semet@thalesgroup.com

Abstract

We apply Ant Colony Optimization concepts to the problem of finding appropriate reward values after successful task completion in serious games. Our algorithm is deployed within the InLife platform, which leverages the power of serious games augmented with real-world IOT sensors for educational purposes. The platform is deployed on four actual pilot sites in Spain, France and Greece with two distinct applications: teaching sustainable behavior to university students and improving social interaction skills for autistic children. In a decentralized, swarm intelligence fashion and based on individually released success and failure pheromones, our generic reward computation strategy seeks, by adjusting reward amounts on the fly, to achieve maximum efficiency in catalyzing behavior change while balancing adaptivity, parsimony, fairness and variety. On top of the necessarily limited real-world data, large-scale numerical validation of the algorithm is obtained with a specifically designed simulator, whose underlying cognitive model was validated by a clinical psychologist. Conducted experiments confirm the relevance and adaptive nature of the obtained pheromone map: the system automatically adjusts to changes in the environment such as the introduction of new students or pedagogical items. Experiments also validate all aforementioned desired characteristics and show substantial quantitative performance gains with respect to a static reward scheme in behavior change metrics, speed and success rates, of up to 40 percent with equal reward budget.

Introduction

European H2020 project InLife, conducted between 2016 and 2018, has focused on ways to augment education with modern information technologies. To that end, it developed a service oriented platform based on three technological pillars: Serious Games, Connected Objects (IOT) and Artificial Intelligence. The developed platform intends to synergize these three powerful levers in order to maximize learning speed and positive behavioral evolution in populations of students.

Two distinct applications of the platform were conducted through the development of two serious games: Aksion, by Imaginary srl and Iceberg by Five Flames Mobile. The first one, Aksion (see sample screen capture in figure 1) is devoted to teaching or improving social inclusion behaviors

to children with autistic spectrum disorders by simulating interactions with their peers in familiar contexts such as the classroom, the beach or the library. These simulated situations come with various games, quizzes and multiple choice questions to evaluate the child's skills and monitor her progress. The second serious game, Iceberg (see figure 2), aims at teaching sustainable notions and behaviors on two levels: in the real-world and in the game. Environment friendly actions in the real-world (e.g. turning off the lights when leaving an empty room, using recycling bins, avoiding unnecessarily using the printer, etc.) are monitored by IOT sensors and rewarded with bonus objects to be used in a real-time strategy game in which players are responsible for the well-being and growth of a sea ice ecosystem inhabited by animals. The in-game rewards can take several forms: special objects, game currency to be spent for upgrades or skins, reputation points, or, conversely for negative rewards (punishments), general degradation of the ecosystem (animals dying, ice melting, etc.). Through carefully crafted game mechanics and sustainability indicators, the game teaches environmental mechanisms and their preservation by encouraging strategies that maximize sustainability, safety and diversity.

For evaluation purposes, the platform was deployed and the games tested in four distinct real-world pilot sites in France, Spain and Greece. For Iceberg, these pilot sites are public libraries or universities and for Aksion, they are specialized institutions where children interact and play the serious game during sessions organized and controlled by professional educators and psychologists. In this context, we developed a playing data processing module called "Incentive Server" to compute reward values as smartly as possible to satisfy or maximize several possibly antagonistic criteria. These rewards are attributed to players when they succeed, in either the real-world or the digital world, in completing a task or in behaving the right way. The rewards should be valued so as to maximize overall system efficiency in pursuing objectives specified by educators. Typically, these objectives are linked to positive behaviors or specific skills whose success probability should increase harmoniously for



Figure 1: Screen capture of the Aksion serious game: children with ASD encounter simulated social inclusion situations they must handle with adequate behaviors. Copyright Imaginary srl.



Figure 2: Screen capture of the Iceberg serious game: the player is responsible for the management of an ecosystem (a portion of floating ice with animals, buildings, energy and storage needs) in a real-time strategy fashion. Copyright Five Flames Mobile.

all members of the population of students. The algorithm's responsibility is therefore to make the most efficient use of limited reward resources according to both individual and global characteristics of learners.

This paper is organized as follows. We start by stating the problem and detailing the desired characteristics that guided our algorithmic choices. Then we provide an overview of the state of the art in Serious Games, Gamification and their support by Swarm Intelligence algorithms. The two following sections describe the multi-agent simulator we developed for validation and the adaptive ant-colony based reward algorithm respectively. Experimental results are then given on several validation scenarios. We conclude by outlining why we think this contribution is an example of how Artificial Life can help address societal challenges.

Motivations and Desired Characteristics

As we have seen, in the general context described above, the problem to be solved is to find the appropriate value for rewards attributed after successful task completion or correct behavior. More specifically in this first applied use-case with Aksion and Iceberg, we seek to maximize success rates, over the whole population of players, in undertaking specific actions in the real-world, as detected by IOT devices (e.g. turning the light off when leaving an empty room).

Beyond that simple functional target, the high-level requirements and desired characteristics for the reward algorithm are as follows. First and foremost, the algorithm shall **maximize the selected criterion** to measure behavioral change or learning progress. Reward attribution shall be **player specific**, **temporally specific** and able to take the **whole population** of players and social equity criteria into account. The algorithm shall be **scalable**, respond in **real-time** and be simple enough to be **explainable** in its pedagogical consequences so as to facilitate its adoption by non technical end-users. Most importantly, the reward algorithm shall be **adaptive**, which means able, without external human intervention, to adjust attributed rewards according to the various changes that might occur in the game environment. Finally, the algorithm shall be **generic** because the InLife platform should be able to accommodate any future serious game.

Literature review

The recent development of numerous forms of participative technologies in the wake of Web 2.0 and the generalized interconnection of information systems indirectly led to the emergence of a new field of research and innovation focused on catalyzing that participative collective energy with game-inspired mechanisms used in serious, not entertainment oriented contexts such as education or knowledge management. Known as "gamification" (Hendrikx et al., 2015), this design strategy consists in increasing user or player engagement by acting on their intrinsic motivation and its social ramifications with game-inspired rewards. It also allows to facilitate participation according to specifically chosen criteria, for example to compensate for disequilibria in attendance or browsing. To that end, "gamified" applications or serious games designers resort to a number of heuristic principles that consist in defining reward attribution rules or reputation points systems (Seaborn and Fels, 2015). These systems have two advantages: not only do they help increasing participant motivation or success rates but they produce, as a side-effect, a very rich information repository that will allow for a better knowledge and understanding of both the global population and its individual members. That repository can be used to improve the game to make it more efficient. Such improvements can be performed manually or automatically, using data processing, machine learning or optimization algorithms.

These recent years, “Serious Games” have known a steadily increasing popularity (Boyle et al., 2016) and proceed with the same strategy: using game inspired mechanisms to educate and transmit knowledge and good practices or behaviors. Learning can indeed be more efficient when it is enhanced with mechanical motivation mechanisms such as rewards, scoring boards or reputation points. Serious Games also rely on the psychological mechanisms of impersonation (role-playing games) but also on narrative immersion that can be augmented by links to the real world through IOT sensors (Kurilovas et al., 2014).

As an application example, many contributions (see for example (Bernardini et al., 2014; Grossard et al., 2017)) tend to indicate that using Serious Games can have a positive impact on the education, particularly for social inclusion skills, of children afflicted with an Autistic Spectrum Disorder (ASD), as illustrated by the Aksion game developed during the project.

While consensual good practices start to emerge in intelligently quantified and oriented gamification, there are relatively few attempts at systematically resorting to mathematical models or even statistics or algorithms to automate design of gamification systems. There is therefore a widely open field for applying optimization techniques or data processing through machine learning. A few attempts have already been made at adaptive gamification: (Ososky, 2015; Richter et al., 2015; Semet et al., 2003). They usually consist, however, in dynamically adjusting presentation probabilities for pedagogical items according to statistics on the player population. Very few works (Cheng and Vassileva, 2005, 2006) also try to offer adaptive scores or rewards in order to encourage participation on selected topics.

In Artificial Intelligence, the Swarm Intelligence paradigm (Bonabeau et al., 1999) consists in trusting the intelligence emerging out of the interactions of a plurality of “simple-minded” entities with the completion of a possibly complex task. As is now commonly known, this way of decomposing problems has many interesting properties and traits among which: parallel processing, distribution, specialization/synergy, scalability, computational efficiency or robustness to fault. Among the numerous algorithmic forms of Swarm Intelligence, Ant Colony Optimization (ACO) (Dorigo and Stützle, 2004), introduced in the 1990’s, is a popular variant of stochastic search inspired the collective behavior of social insects, ants, that collectively solve problems through indirect communication. Initially very successfully applied to shortest path finding in graphs and ensuing computational problems such as the Travelling Salesman’s or network packet routing, they have been applied to many other domains (Bonabeau et al., 1999; Dorigo and Stützle, 2004) and form today a body of algorithms and heuristics for optimization and machine learning at large. These algorithms are particularly apt at dynamic optimization (Mavrouniotis et al., 2017) as they have

astonishing reactive and adaptive properties.

Applying Swarm Intelligence to optimized gamification and, more generally, to adaptive learning feels both natural and promising. One can indeed imagine a straightforward mapping between a population of students or learners in a serious game and a population of simple interacting entities that collaborate to form the pheromone map that forms the basis of an ant-inspired algorithm. It is therefore straightforward to imagine an information collection strategy, instantiated at the learner level, that procures, individual bit by individual bit, information for the emerging global picture and algorithm. Ant Colonies, natural and otherwise, do that by following the “stigmergy” principle, which consists in using the common environment as an information repository for indirect communication. Through pheromones and stigmergic communication, ants gradually accumulate a treasure of global information that can be used for various purposes such as retrieving shortest paths or circuits from a well informed map of the surroundings. Making efficient use of that information however, is not trivial and shall be the responsibility of carefully designed algorithms that will steer the search toward interesting solutions to the problem at hand.

This analogy is promising for the application described in this paper, namely adaptive rewards for serious games, because many of the key properties of these algorithms correspond exactly to the high-level requirements outlined in the previous section, particularly adaptivity, reactivity and scalability. We identified a number of works that attempt to apply ant colony algorithms or swarm intelligence more generally to e-learning optimization or serious games: (Semet et al., 2003; Gutierrez et al., 2007; Huang and Liu, 2009; Dharshini et al., 2015) but they are mainly focused on improving pedagogical paths by trying to optimize presentation probabilities for the various teaching items according to statistics on both individual students and the global population in a spirit akin to the abundant literature in Recommender Systems and Collaborative Filtering. (Semet et al., 2003; Gutierrez et al., 2007) are noteworthy for introducing the notion of accumulated success and failure statistics as an adaptive pheromone map. The natural application of this global information cartography to intelligent and adaptive reward allocation according to success rates, as described in the present work, has not been reported in the available literature to our knowledge.

Multi-Agent Validation Simulator

Hypotheses and Principles

In order to procure numerically significant validation and compensate for the unfortunate lack of real-world data in sufficient provision and so as to be agile in designing a trustworthy algorithm, we designed and implemented a multi-agent simulator for the population of learners. Its design started from a common-sense analysis to describe a numer-

Individual decision model

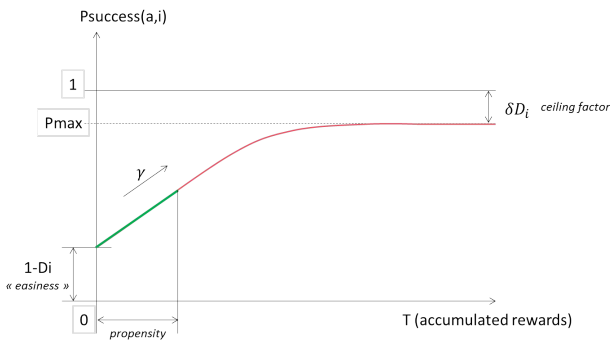


Figure 3: Simulation Model: plot of the essential decision function: according to how much reward was received (x axis), success probability (P_{success}) for rewardable actions is computed (y axis) by a function that incorporates the various aspects of the model. D_i is action difficulty, T stand for the accumulated reward “treasure”. γ and δ are empirically set scaling constants.

ical function reproducing the decision process and the phenomenon that sees propensities to do an action increase with respect to the amount of attributed reward. The principles we chose to follow, as illustrated in figure 3 are the following. **Convergent Positive Reinforcement:** The probability for an individual, to undertake an action, increases with the attribution of rewards related to that action but with decreasing marginal impact (the more reward you get the less impact it has). **Varying Propensities:** before any reward is attributed, the spontaneous probability to undertake an action, varies among individuals. **Varying Difficulties:** actions or tasks are not homogeneously easy, they vary in difficulty and the ensuing success/undertaking probability varies correspondingly. **Bounded Probabilities:** individuals cannot reach a perfect realization probability. **Reward Evaporation:** rewards have immediate impact but that impact spontaneously fades back to zero over time.

These design principles lead to the numerical function in figure 3: the function, an adapted sigmoid, is monotonously increasing (positive reinforcement) and converges to a maximum value based on action difficulty. Individual propensities provide a head start on the x axis and rewards move the player back and forth on that axis when they are attributed and evaporated respectively. Action difficulty finally, sets the basis value for the probability when the natural propensity is 0 and no rewards have been attributed yet.

That simulation model is important for the validation phase of the reward algorithm but it is important to note that it is not intended to be a realistic model of the cognitive learning process, it is a mathematical tool designed to examine the scaling, statistical behavior of reward algorithms

when confronted to a population of probabilistic agents that conform to the simple but central reinforcement learning hypotheses we make about rewards. It has, however, to be realistic enough to form a sensible basis for early stage validation. We therefore had our simulation model controlled and validated by a professional psychologist, Dr Antonio Ascolese. Although conducted conscientiously, this preliminary form of validation is modest and shall be expanded with further work on state-of-the-art cognitive modeling.

Algorithms and Formulae

To reproduce the temporal succession of playing sessions for all players, we follow the simple loop given in algorithm 1 where the *play* procedure is implemented straightforwardly by comparing a floating point number randomly drawn between 0 and 1 to the probability of success $p(a, i, t)$ for the considered player i , action a at time t , as given by equation 1 where σ is an adapted sigmoid function and D_a is the difficulty level of action a . U_t is the amount of reward unspent at time t .

Algorithm 1 Simulating learning occurrences

```

Randomly generate player population
for each timestep  $t$  do
  for each action  $a$  do
    for each player  $i$  do
       $R = \text{computeReward}(a, i, t)$ 
       $\text{play}(a, i, t)$ 
      if success then
         $T(i, a) += R$ 
      else
         $U_t += R$ 
      end if
    end for
  end for
  Evaporate accumulated rewards
end for

```

$$p(a, i, t) = (1 - D_a) \times \sigma(P_{a,i} + T_{a,i}, (1 - \delta)D_a) \quad (1)$$

$$\sigma(x, A) = 2A \times \left(-\frac{1}{2} + \frac{1}{1 + e^{-\xi x}} \right) \quad (2)$$

We also consider that accumulated rewards, the so called “treasure” T , see their impact decrease over time with the following evaporation formula:

$$T_{a,i,t} = \rho \times T_{a,i,t-1} \quad (3)$$

Reward Attribution Algorithm

In the light of the high level requirements outlined above, we chose to use algorithms in the family of Artificial Ant Colonies as they offer remarkable dynamic optimization capabilities in a decentralized, scalable way and provide a

readable, very useful adaptive information map to fuel decision support. Additionally, because they work as a form of reinforcement learning algorithm, they provide dynamic “controllers” that can be conveniently used in real-time as is necessary here. The reward computation module works within a global platform architecture connecting the player, the serious game, connected objects and the Incentive Server hosting the ant reward algorithm.

Hypotheses and Principles

Generally speaking, we chose to follow popular wisdom: “if it ain’t broke, don’t fix it!”. In other words, we chose to direct reward where it is most needed and conversely, to avoid spending some where results are already satisfying. This can be stated more precisely as follows: more reward should go to poorly successful players, more reward should go to poorly undertaken actions, unspent reward should be used similarly and, finally, it is pointless to spend reward on already successful players or actions.

It is important to note that these particular strategic choices are just one possibility out of many possible options, as opposed notably, to the choice that would suggest to spend rewards on already successful actions or players so as to make them even more successful to reach a globally better outcome. This is a political design decision that depends on the context, on end-users, possibly on experimental validation and, most importantly, that is entirely independent of the underlying, generic approach we propose.

Algorithms and Formulae

We follow the classical Ant Colony Optimization metaphor by mapping individual players to individual ants and success or failure events to pheromone release which is memorized locally at the action level as normalized success rates. Evaporation is implemented coarsely by using a sliding time window that prevents too old information to have impact on present reward computation.

When browsing graphs in search of shortest routing paths or tours, artificial ants traditionally base their decisions on a power mean function that blends collectively accumulated local information read in pheromone concentration and heuristic clues read from the surroundings, usually outgoing edge lengths. In our case, since we seek to maximize, with parameterizable importance, both player success rates and action success rates, two dimensions that are distinct although correlated to some extent, we incorporate both in the formula, as outlined below:

$$\tau_{t,a}^A = \frac{\sum_{k=t-W}^t S_{a,k}}{W} \quad (4)$$

$$S_{a,k} = \frac{\sum_{i \in \Omega_{a,t}} s_i}{|\Omega_{a,t}|} \quad (5)$$

$$\gamma_{a,t}^A = \frac{(1 - \tau_{t,a}^A)^\alpha}{\sum_{a \neq b} (1 - \tau_{t,a}^A)^\alpha} \quad (6)$$

where $\tau_{t,a}^A$ is the success rate of action a at time t over a sliding time window of size W , $S_{a,k}$ is the average success rate of action a over all occurrences of a during time step t , $\Omega_{a,t}$ is the set of occurrences for action a (i.e. any player attempting action a) at timestep t and s_i is an individual, binary success variable valued at 1 when the attempt is a success and at 0 otherwise. $\gamma_{a,t}^A$ finally, represents the normalized failure rate, which we want positively correlated with reward amounts, augmented with an exponent α to control the importance of that particular criterion.

Similarly, we keep track of success rates at the player level by calculating $\gamma_{i,t}^P$ in a similar way but by looping over all players instead of looping over all actions:

$$\gamma_{i,t}^P = \frac{(1 - \tau_{t,i}^P)^\beta}{\sum_{i \neq j} (1 - \tau_{t,i}^P)^\beta} \quad (7)$$

Next, we define the basis for the attributed reward, the so called “Reward Portion” RP by aggregating both success rate based proportions described above and the daily budget of reward available for distribution:

$$RP_{a,i,t} = (DB + U_t) \times \gamma_{a,t}^A \times \gamma_{i,t}^P \quad (8)$$

where DB stands for Daily Budget, the amount of reward one is allowed to spend at a given timestep and U_t is the amount of budget left unspent so far, when arriving at timestep t .

We also introduce an important correcting factor, which we call the “wealth factor” which biases reward attribution so as “not to favor the already wealthy”:

$$WF_{a,i,t} = \frac{1}{2} + (1 - \sigma(T_{a,i}, \frac{1}{2})) \quad (9)$$

$$\sigma(x, A) = 2A \times \left(-\frac{1}{2} + \frac{1}{1 + e^{-\delta x}}\right) \quad (10)$$

where σ is a sigmoid function and $T_{a,i}$ is the player i ’s “treasure”, i.e. the total amount of reward she has accumulated so far.

The final, central equation of the system is therefore as follows:

$$R(a, i, t) = WF_{a,i,t} \times RP_{a,i,t} \quad (11)$$

it gives the final amount of reward $R(a, i, t)$ to be given to player i , successfully attempting action a at timestep t .

Discrete Rewards

In order to accommodate discrete choices of rewards, which is a requirement for certain games that attribute object rewards (characters, bonuses or customization items) among a finite set of options as opposed to a continuously valued amount of game currency, we proceeded by transforming the attributed continuous reward value into a basis probability for iterated Bernoulli attempts at climbing the ladders of the table of possible rewards ordered by increasing utility value. Further details on this seemingly simple feature that proved particularly difficult to design without loss of mathematical efficiency in validation scenarios or variety related issues, shall be reported elsewhere.

Experimental Results

Real-World Deployment

During the course of the project, the InLife platform was deployed and tested on four pilot sites in the spring and summer of 2018: the School of Electrical and Computer Engineering, Zografou Campus in Athens, Greece (Iceberg), the PanHellenic Association of Adapted Activities in Athens, Greece (Aksion), the Municipal library “Adolfo Miaja de la Muela” in Valladolid, Spain (Iceberg) and the Association for Living the Self-Governance (A.V.A.G) in Les Ulis, France (Iceberg).

Unfortunately, the short duration of the project did not allow for statistically significant analysis regarding the positive influence of the reward algorithm specifically. We however reckon it worth reporting that the system was actually deployed in the field, worked, and generated very positive feedback on the use of game-inspired mechanisms such as rewards in stimulating player engagement.

Simulator Based Validation

This section shall therefore focus on our in-lab experimental campaign which was based on the comparison between the proposed adaptive reward algorithm and a naive, static, strategy that always attributes the same reward for all actions and all players, regardless of the specifics.

Beyond qualitative appreciation of the system’s properties, we introduce a metric as a basis for quantitative comparisons. This metric, called LIM for Loss Integral Metric, is an approximation of the integral of the minimum success probability, across all rewarded actions, as a function of time over the experiment’s time frame. In other words, to compute this metric, at each time step we identify the action with the smallest success rate, over a considered past time window, and add that probability to a running counter. This results in a metric that favors situations where the worst success rate among actions is as high as possible for as long as possible and as soon as possible.

On that basis, we seek to validate our algorithm’s properties and comparative efficiency on many validation scenarios. We report the outcome of six of the most illustrative

below with plots, for lack of space only for the three most significant ones. Since part of the system is stochastic, validation of the comparisons with statistical significance testing was conducted on 100 independent runs each time with differences in averaged metrics confirmed with a pairwise bilateral Student’s T-test with a 5% confidence interval. Parameter values are the following: $\alpha = 1$, $\beta = 3$, $\delta = 0.1$, $\rho = 0.98$, $W = 10$. In all experiments below, we compare success probabilities between actions over timesteps, as sampled over a running time window of size W .

Scenario 1 is the simplest possible validation. It features three actions with different reward strategies. Action 1 has no reward at all, action 2 has a static reward and action 3 the ACO adaptive reward. There is only one player. In the strict absence of reward, action 1 does not improve at all, in its success probability, while actions 2 and 3 efficiently converge to satisfying levels. The absence of difference between 2 and 3 is explainable by the absence of disequilibrium, player-wise or action-wise to compensate for thanks to adaptivity.

Scenario 2 (figure 4) and scenario 3 (not plotted, qualitatively similar) are sample calibration test cases and show the difference between a static reward scheme and the ACO adaptive reward scheme when there is a disequilibrium. In scenario 2 indeed, there is one “hard” action A1 with difficulty $D = 0.1$ and low initial success rate and one “easy” action A2 with difficulty $D = 0.9$ and high initial success rate. With our adaptive algorithm, the difference in success rates is quickly compensated and both actions converge quickly to satisfying values while the static scheme fails at fixing the situation. The upper plot shows success probabilities for both actions under both reward schemes (static with dashed lines and adaptive with continuous lines). The distribution of the difference between A1 and A2 in both cases is shown in the lower plot together with statistical testing outcome over 100 runs, which unambiguously confirms the superiority of the adaptive scheme in this case after timestep 15. Scenario 3 offers a symmetrical situation with a “good” student and a “bad” one whose success rates are compared on one same action.

Scenario 4 is a full scale validation exercise with dimensions sampled around those seen in real-world pilots. It has 100 randomly generated players and 5 actions, 2 of which are “hard” and 3 of which are “easy”. 50 of the players are initially good (high propensity values ($P_i = 0.9$)) and 50 are bad (low propensity values ($P_i = 0.1$)). The plotted curves (top), corresponding to one sample run with the adaptive scheme, show quick harmonization of success rates for all actions and convergence to high values around 0.91 on average. The LIM metric is worth 94.12 for the static reward scheme and 155.67 for the adaptive scheme. Over the course of the experiment, 200 simulated days, there is a total of 22059 failures (occurrences of players failing at a task or failing to have to correct behavior) on average over 100

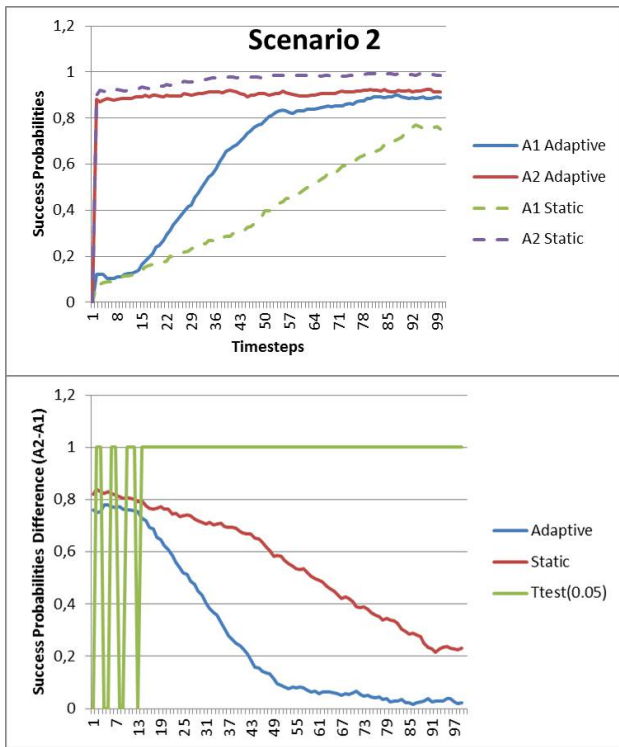


Figure 4: Scenario 2: 1 player, 1 hard action A1 and 1 easy action A2. Static rewards (dashed lines in top plot) are much slower than adaptive rewards to compensate this disequilibrium. The bottom plot shows the mean difference between A1 and A2 with both schemes as well as the outcome of the corresponding T-test, 1 meaning there is a significant (95%) difference in the superiority shown by the adaptive scheme.

runs with the static scheme and of 12579 failures with the adaptive scheme, which represents a gain of 42.97%. The static scheme spends 89631 reward units on average and the adaptive scheme slightly more, 92425.

Scenarios 5, not reported here for lack of space confirms that discrete rewards as we introduced them, work just as well as continuous ones to steer heterogeneous success rates in the right direction in the context used for scenario 4.

Scenario 6 (figure 6) finally, sheds lights on the adaptive feature of the ant algorithm. By suddenly changing the difficulty value of a specific action at time step 100, thus simulating a very significant change in the game environment (such as a modification of the IOT conformation or an educator induced change in task difficulty), one can see the system automatically detect and adapt to this change only by implicitly sampling success rates through pheromones and by accordingly increasing reward amounts allocated for the newly difficult action so as to bring the corresponding success rate back to the level reached by the other actions, yet without sacrificing those. For comparison, the plot also

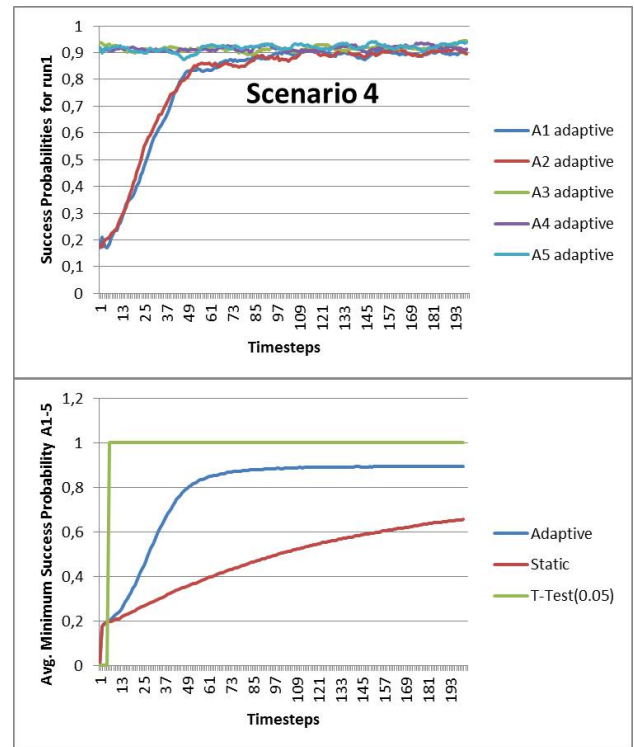


Figure 5: Scenario 4: main validation result with full scale. 100 players, 200 simulated days, 5 actions (2 hard ones (red and blue in top plot), 3 easy ones). Again the disequilibrium in success rates, both for actions (see plot) and players (not shown) is compensated much faster by ant-based adaptive rewards, as can be seen in the lower plot, which shows the minimum success probability across all actions averaged over 100 runs for both reward schemes as well as the corresponding t-test outcome (95% confidence) to confirm the superiority of the adaptive scheme with a similar reward budget. The gain in decreased number of failed attempts is of 42.97%

shows, with dashed lines, the unsatisfying evolution of success probabilities for actions A1 and A2 with a static reward scheme. Results are averaged over a hundred runs and tested for statistical significance with 95% confidence.

Conclusions

We introduced the use of Ant Colony Optimization for the computation of adaptive rewards in serious games. Experimental validation results, obtained with a specific, simple but validated multi-agent simulator, confirm the qualitative properties one expects from a swarm intelligence algorithm and show significant quantitative improvements with respect to a static reward scheme. The system was also deployed in the real world on four distinct pilot sites in Spain, France and Greece.

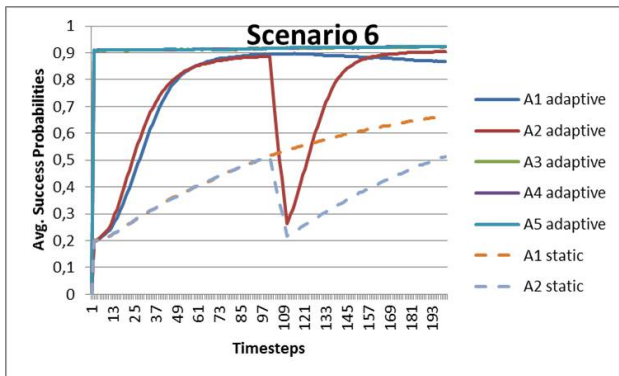


Figure 6: Scenario 6: at time step 100, a sudden change occurs in the game environment and the success rate of action A2 drops significantly. The ant based algorithm is able, without external intervention, to implicitly detect the change and adapt its reward attribution strategy accordingly.

We argue that this work constitutes a double example of how Artificial Life can help address modern challenges. Serious games in general and Iceberg in particular, use Artificial Life both to procure immersive narration with behavior assessment and to help understand, and therefore fix or support, complex and critical social or natural phenomena such as those implicated in sustainable development concerns. Additionally, artificial life algorithms such as the Ant Colony we used in this work, are, thanks to their fantastic mathematical properties of adaptivity, resilience and distributed intelligence, invaluable tools in handling the many complex systems that make up our world and in increasing operational benefits or efficiency, exemplified here in the area of education, while limiting environmental costs.

Acknowledgments

This work was supported by the H2020 InLife project.

References

- Bernardini, S., Porayska-Pomsta, K., and Smith, T. J. (2014). Echoes: An intelligent serious game for fostering social communication in children with autism. *Information Sciences*, 264:41 – 60. Serious Games.
- Bonabeau, E., Dorigo, M., Théraulaz, G., Theraulaz, G., and Santa Fe Institute (Santa Fe, N. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Proceedings volume in the Santa Fe Institute studies in the sciences of complexity. OUP USA.
- Boyle, E. A., Hainey, T., Connolly, T. M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus, M., Ribeiro, C., and Pereira, J. (2016). An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers & Education*, 94:178 – 192.
- Cheng, R. and Vassileva, J. (2005). Adaptive reward mechanism for sustainable online learning community. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology*, pages 152–159, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- Cheng, R. and Vassileva, J. (2006). Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Modeling and User-Adapted Interaction*, 16(3):321–348.
- Dharshini, A. P., Chandrakumarmangalam, S., and Arthi, G. (2015). Ant colony optimization for competency based learning objects sequencing in e-learning. *Applied Mathematics and Computation*, 263:332 – 341.
- Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Grossard, C., Grynspan, O., Serret, S., Jouen, A.-L., Bailly, K., and Cohen, D. (2017). Serious games to teach social interactions and emotions to individuals with autism spectrum disorders (asd). *Computers & Education*, 113:195 – 211.
- Gutierrez, S., Valigiani, G., Jamont, Y., Collet, P., and Kloos, C. D. (2007). A swarm approach for automatic auditing of pedagogical planning. In *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pages 136–138.
- Hendriks, F., Bubendorfer, K., and Chard, R. (2015). Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184 – 197.
- Huang, Y.-M. and Liu, C.-H. (2009). Applying adaptive swarm intelligence technology with structuration in web-based collaborative learning. *Computers & Education*, 52:789–799.
- Kurilovas, E., Zilinskiene, I., and Dagiene, V. (2014). Recommending suitable learning scenarios according to learners’ preferences: An improved swarm based approach. *Computers in Human Behavior*, 30:550 – 557.
- Mavrovouniotis, M., Li, C., and Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33:1 – 17.
- Osofsky, S. (2015). Opportunities and risks for game-inspired design of adaptive instructional systems. In Schmorrow, D. D. and Fidopiastis, C. M., editors, *Foundations of Augmented Cognition*, pages 640–651, Cham. Springer International Publishing.
- Richter, G., Raban, D. R., and Rafaeli, S. (2015). *Studying Gamification: The Effect of Rewards and Incentives on Motivation*, pages 21–46. Springer International Publishing, Cham.
- Seaborn, K. and Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74:14 – 31.
- Semet, Y., Lutton, E., and Collet, P. (2003). Ant colony optimization for e-learning: observing the emergence of pedagogic suggestions. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No.03EX706)*, pages 46–52.

The Effects of Individual and Social Learning on the Evolution of Cognitive and Communicative Aspects of Language Abilities

Hiroto Yonenoh¹, Reiji Suzuki¹ and Takaya Arita¹

¹Graduate School of Informatics, Nagoya University, Japan
yonenoh.hiroto@a.mbox.nagoya-u.ac.jp

Introduction

Recently, it is recognized that human linguistic abilities and human language are likely to have been shaped by their co-evolutionary processes, in other words, the interplaying process of the biological evolution of the language faculty and the cultural evolution of language (Smith, 2018).

In a general context of co-evolution of biological and cultural processes, including that of language (Christiansen and Kirby, 2003), effects of individual and social learning have been regarded as important factors. Higashi et al. (2018) constructed a computational model of evolution on a rugged fitness landscape which has many local optima, assuming the evolution of a complex and adaptive ability composed of mutually interacting traits. They showed that individual learning and social learning could work complementarily in the course of adaptive evolution on the rugged fitness landscape. That is, individual learning can find new adaptive phenotypes thanks to the diversity of genetic expressions created by social learning. Social learning also can keep a new adaptive phenotype found by individual learning in the population. We consider the human language faculty as one of such abilities at least in part because it is composed of various sub-faculty traits of language (e.g., brain structures, vocal organs). There are thought to be two sorts of primary abilities of human language (Harman, 1975). One is a cognitive ability or capacity of thoughts (e.g., building language hierarchy). The other is a communicative ability (e.g., intention sharing). However, they only assumed mutual interactions among traits within an individual, which could be regarded as the traits related to the cognitive ability of language. Thus, in order to deal with the evolution of communication ability, the existence of a communication partner and inter-individual interactions of their traits must be assumed.

The purpose of this study is to clarify how cognitive and communicative abilities of language can evolve in terms of the co-evolution of linguistic abilities, based on social and individual learning. For this purpose, we extended the model of Higashi et al. (2018) by incorporating inter-individual interactions among traits into it.

Model

We constructed an extended version of Higashi et al. (2018)'s model. We assume a population of $N/2$ parent and $N/2$ child individuals, each has M traits $t_i \in [1, M]$ ($i=0, 1, \dots, M-1$) represented as positive integers. These traits are assumed to be related to sub-faculty traits of language. Each gene $g_i \in [1, M]$ in a M -length chromosome represents the initial (innate)

value of the corresponding trait t_i . Each individual has another M -length chromosome, each of which is a boolean value and decides whether the corresponding trait is plastic (learnable) or not. These settings are same as Higashi et al. (2018).

In each generation, $N/2$ pairs of individuals were randomly formed without duplication, and there are L steps of within-pair interactions in parallel. We assume a basic assumption on the contribution of trait sets to the fitness: a fitness contribution x emerges if there are equal or more than x traits whose value is x among the trait set of the focal individual. This represents that the more adaptive linguistic ability is based on the larger number of sub-faculty traits. Then, we define the co-creative fitness *FITNESS* as the product of the following two types of fitness: $FITNESS=COG \times COM$. *COG* (cognitive fitness) of an individual i is the highest fitness contribution calculated from her trait set $\{t_i\}$ using the assumption above. *COG* is the fitness function used in Higashi et al. (2018). *COM* (communicative fitness) is the total fitness contribution calculated from the set $\{t_i, t_j\}$ of the trait set of the focal individual $\{t_i\}$ and that $\{t_j\}$ of her partner j in her pair. We adopted the total fitness contribution for *COM* to represent that various communication processes based on different sub-faculty traits can bring about different types of fitness contribution. The fitness landscape becomes dynamic due to the introduction of *COM*.

Each individual also has a social learning rate s , as a genetic value. At each step, each individual chooses social learning with the probability s or chooses individual learning with the probability $1-s$. When choosing individual learning, an individual creates a slightly modified trait set from her innate trait set $\{g_i\}$ by adding a random value $\{-1, 0, 1\}$ to the value of g_i of each plastic trait. When choosing social learning, the focal individual tries to imitate the traits of the best individual in the previous step. The process makes each plastic trait closer to the corresponding trait of the selected individual, by adding -1 or $+1$ to g_i . The focal individual adopts the new trait set if its *FITNESS* is higher than the one



Figure 1. The average fitness during the last 1,500 generations in the three learning conditions. The values were the average over 20 trials. The red arrow represents that a value on the start point is significantly lower than a value on the end point by Ryan's method with Wilcoxon signed-rank test ($\alpha=0.05$).

in the previous step, assuming that her partner has the trait set in the previous step, otherwise keeps the previous one for fitness calculation. At the end of the generation, all the parent individuals die and the child individuals become parent individuals. $N/2$ new child individuals are created by using a roulette wheel selection from the new parent individuals based on the average co-creative fitness (*FITNESS*) over time steps, single-point crossover, and mutation.

Results and discussion

We conducted experiments with the parameter settings: $N=1000$, $M=10$, $L=50$ for 15,000 generation. We focused on the three learning conditions: “No learning condition”, where the fitness of an individual is calculated with only the initial traits g_i ; “Individual learning condition”, where an individual performs individual learning only ($s=0.0$); and “Both learning condition”, an individual can choose either individual or social learning according to her own social learning probability s .

Figure 1 shows the average fitness during the last 1,500 generations in the three learning conditions. Each of the lifetime fitness (right) means the average fitness values of *COG*, *COM* and *FITNESS*, respectively, and the innate fitness (left) is the fitness value calculated using the genetically determined trait values $\{g_i\}$. We see that each of the three fitness on Both learning condition was higher than the corresponding ones on No learning condition and Individual learning condition. In other words, mutual interactions between both individual and social learning facilitated the adaptive evolution of cognitive and communicative abilities of language. By contrast, if we focus on the innate fitness, there were significant differences in their evolutionary process between Individual learning and Both learning conditions. Each of the three innate fitness on Individual learning condition was higher than the corresponding one on No learning condition. In other words, individual learning facilitated the adaptive evolution of innate fitness. Our additional analysis showed that this result can be explained by the Baldwin effect (Baldwin, 1896) or genetic assimilation (Waddington, 1953) as the component of the Baldwin effect, which is expected to be due to the implicit cost of learning. However, each of the three innate fitness on Both learning condition was lower than the corresponding one on Individual learning condition, showing that the social learning process retarded the genetic acquisition of adaptive traits.

Figure 2 shows the rough trajectory of the population on the 3 fitness landscapes in a trial. On the top panel, we see that the both fitness gradually increased but there were several peaks of the fitness landscape, preventing the increase in the fitness. In addition, the difference between the lifetime and innate fitness tended to be large, implying that the learning process was getting more important to keep the high adaptivity. From the comparison between the middle and the bottom panels, we can see that the difference in the communicative fitness between the lifetime and innate fitness tended to be large and variable while the one in the cognitive fitness was small and steady. This means that it is more difficult for genetic assimilation to occur on communicative fitness than on cognitive fitness. This might be because

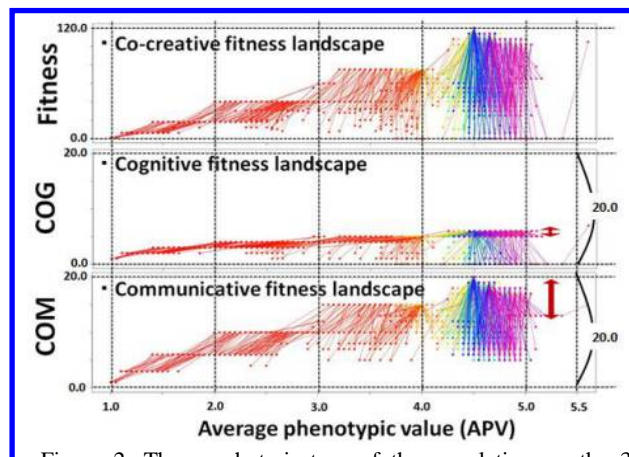


Figure 2: The rough trajectory of the population on the 3 fitness landscapes in a trial of Both learning condition. Each point represents the fitness value (y-axis) of the best individual in terms of co-creative fitness and her average trait value (x-axis). The innate and lifetime fitness of each individual were connected. The color gradation represents the generation (red->orange->yellow->blue->purple).

dynamic changes in the traits of her partner prevent an individual from genetically acquiring the high communicative ability while the innate cognitive ability tended to evolve stably because it is determined by her own traits.

Conclusion

We constructed an extended version of Higashi et al. (2018)’s model to clarify how cognitive and communicative abilities of language can evolve, based on social and individual learning. We found that roles of individual and social learning can work complementarily. It is also implied that the stable evolutionary process of cognitive abilities is based on repeated occurrences of the Baldwin effect and dynamic and less assimilated evolution of communicative traits can work together, contributing to the evolution of cognitive and communicative aspects of language abilities.

Acknowledgements

This work was supported in part by JSPS/MEXT KAKENHI: JP17H06383 in #4903 (Evolinguistics).

References

- Baldwin, J. M. (1896). A new factor in evolution. *The American Naturalist*, 30 (354): 441-451.
- Christiansen, M. H., and Kirby, S. (2003). Language evolution: consensus and controversies. *Trends in cognitive sciences*, 7 (7): 300-307.
- Harman, G. (1975). Language, thought, and communication. In: K. Gunderson (ed), *Minnesota Studies in the Philosophy of Science*, 7: 270-298.
- Higashi, M., Suzuki, R. and Arita, T. (2018). The Role of Social Learning in the Evolution on a Rugged Fitness Landscape. *Frontiers in Physics*, 6 (88).
- Smith, K. (2018). How culture and biology interact to shape language and the language faculty. *Topics in Cognitive Sciences*, DOI: 10.1111/tops.12377.
- Waddington, C. H. (1953). Genetic assimilation of an acquired character, *Evolution*, 7: 118-126.

Neuroevolution of Humanoids that Walk Further and Faster with Robust Gaits

Ben Jackson¹ and Alastair Channon¹

¹ School of Computing and Mathematics, Keele University, ST5 5BG, UK
{b.jackson,a.d.channon}@keele.ac.uk

Abstract

Bipedal locomotion requires precise rhythm and balance. Here we demonstrate two fitness-function enhancements applied to OpenAI's 3D Humanoid-v1 walking task using a replica of Salimans *et al.*'s evolution strategy (Salimans *et al.*, 2017). The first enhancement reduces control cost, following a start-up period, and the second enhancement penalises poor balance. Individually, each enhancement results in improved gaits and doubles both median speed and median distance walked. Combining the two enhancements results in little further improvement in the absence of noise but is shown to produce gaits that are much more robust to noise in their actions, with median speed, distance and time two to five times those of the default and individual-enhancement gaits at an intermediate noise level.

Introduction

Bipedal locomotion has been the focus of many studies (Winter, 1991). Its emergence has been associated with endurance running and freeing the hands for other tasks such as tool use (Hewes, 1961; Bramble and Lieberman, 2004). The main challenges relate to balance and generating cyclic motion across two limbs to produce a stable gait. In robotics, bipedal walking is often achieved through zero-moment point computation (Vukobratović and Borovac, 2004), for example in evaluating fitness (Fukuda *et al.*, 1997).

Initially, the notable instances of bipedal learning utilised Central Pattern Generators. CPGs are recurrent neural networks that produce rhythmic activity (without requiring rhythmic inputs) that is typically modulated by descending or peripheral inputs (Guertin, 2013). Modelled after pattern generators found in human and animal spines, they offer a biologically inspired solution (Taga *et al.*, 1991; Ijspeert, 2008; Van der Noot *et al.*, 2015). Reil and Husbands used a genetic algorithm to evolve a CPG's weights to produce stable bipedal walking (Reil and Husbands, 2002). Measuring fitness as distance from the point of origin, they achieved a 10% stability success rate. This was improved to 80% through the introduction of an oscillatory bonus in the fitness function. Reil and Husbands also added sensors to achieve directional walking. Directional motion was also exhibited

in (Gökçe and Akin, 2010), which used evolution strategies (Beyer and Arnold, 2001) to optimise CPG-based walking in simulation and then on real robots.

Following this, DeepMind used reinforcement learning to train a 3D humanoid model (Tassa *et al.*, 2012), using the MuJoCo physics engine (Emanuel Todorov *et al.*, 2012), to produce complex bipedal locomotion (Heess *et al.*, 2017). A distance-based reward function and policy-based gradient descent learning were used in incrementally complex environments to produce behaviours. The humanoid agents were able to demonstrate running, crouching, jumping and turning behaviours, the most notable successful model to date.

Salimans *et al.* (OpenAI) evolved walking gaits for the MuJoCo 3D Humanoid-v1 environment in OpenAI Gym (Tassa *et al.*, 2012; Brockman *et al.*, 2016) using an evolution strategy (Salimans *et al.*, 2017). The architecture described in their paper was a multilayer perceptron with two 64-unit hidden layers (using tanh units) mapping 376 inputs to 17 joint torques. In each iteration, many episodes were run using random parameter perturbations to test the robustness of the current strategy. The strategy parameters for the next iteration were then updated based on the calculated gradient estimate of the combined episode results. This produced successful biped neurocontrollers in as little as ten minutes, using 1440 CPU cores.

Subsequently, Petroski Such *et al.* (Uber AI Labs) used a genetic algorithm on the same problem (3D Humanoid-v1), using two 256-unit hidden layers (matching the configuration file included in the source code released by Salimans *et al.*), achieving success on this task but noting that their GA "took 15 times longer to perform slightly worse than ES" (Petroski Such *et al.*, 2018). They also noted that (while only just qualifying as a deep neural network, having more than one hidden layer) this network contains approximately 167k parameters, orders of magnitude greater than the previous largest neural networks evolved for robotics tasks. They encoded these parameters using a novel method that stores, for each genotype, an initialization seed and a list (that grows with each mutation) of random seeds used to generate mutations to the vector of parameters. They

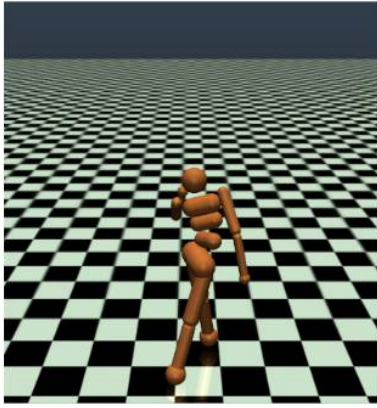


Figure 1: The Humanoid-v1 walker used in this and previous work (Salimans et al., 2017; Petroski Such et al., 2018; Conti et al., 2018).

also applied a weighted novelty-search version of the ES, which was able to produce high scores in a deceptive environment that required the humanoid to walk around a trap (Conti et al., 2018) and demonstrated that ES is more robust to parameter perturbation in the humanoid locomotion task, than both their GA and Trust Region Policy Optimization (Lehman et al., 2018).

In this paper we report on two enhancements that enable Salimans *et al.*'s evolution strategy, using two 256-unit hidden layers, to produce populations of 3D Humanoid-v1 agents (figure 1) that walk further and faster and exhibit novel gaits with much increased robustness to the addition of noise to actions. We also report on the interaction between the two enhancements and how they overcome the difficulty of the task.

Methods

The methods featured here are applied to a replica of Salimans *et al.*'s evolution strategy, which was already capable of producing gaits that move 3D Humanoid-v1 agents quickly and efficiently enough to pass the humanoid walking task. The humanoid walker's goal is to travel (in any direction) as fast and efficiently as possible, failing when its torso falls below (or above) a certain height. Fitness is defined as the sum (over time) of four rewards/penalties that are computed at each timestep: a reward for linear velocity, a control cost based on energy expended, a cost based on how hard the humanoid impacts the ground, and a reward for standing. Our methods aim to improve average speed, distance travelled and episode duration (time). We also aim to improve robustness to noise in the application of actions. Each permutation of enhancements (and the default base-case) was run 20 times, each for 600 iterations (generations). The standard evolution strategy was used due to

its superior performance (see above). As mentioned above, the architecture maps 376 inputs (humanoid state variables: position, rotations, velocities, forces and inertia values) to 17 joint torques, from which we conclude approximately 30k parameters (weights and biases). Each set of 20 runs took around three days to evolve using 40 CPU cores. Each episode (walker fitness evaluation) was limited to a maximum of 1000 timesteps.

Fitness Function Enhancements

Fitness is the sum (over time) of four rewards/penalties that are computed at each timestep: a reward for linear velocity, a control cost based on energy expended, a cost based on how hard the humanoid impacts the ground, and a bonus for having not failed:

$$r = \text{linvel} - \text{contcost} - \text{impcost} + \text{alivebonus}$$

Even in the default case, without any enhancement, this led to the evolution of fast, efficient walking in line with previous results. The impact cost prevented hopping behaviours. However, a considerable number of evolved gaits involved a shuffling motion. These walkers slid along the floor with small movements of their feet.

Control-cost enhancement The first enhancement employed involved reducing the control cost within the fitness function. To encourage gaits that use more motion in novel ways, we applied a scalar multiplier to the control cost term, allowing for new behaviours at the cost of generating less efficient walkers (when the multiplier is below 1). This multiplier can be applied throughout each simulation episode or from a set timestep during each simulation episode. The latter was thought to be potentially beneficial as the gaits evolved typically have a "catch" phase in which walkers align themselves from the starting position into cyclic motions, which can be disrupted by a low control cost. The aim of this set of runs was to produce novel gaits with longer walks by reducing the control cost in the fitness function.

Balance enhancement The second enhancement employed was a fail condition involving the balance of the walker. In the original system, balance is described as the walker's torso's center's vertical (z-) component being outside the range of 1-2 simulation units. To improve walkers' episode durations (walking times) an additional constraint was introduced for the x- and y-dimensions, to terminate walkers with less upright postures. If the torso's center of mass moved outside a circle centered at the midpoint between the walker's two feet (each projected down to the ground plane) then it was considered a failure. The circle's radius is set as a multiple of the current distance between the midpoint and (either) foot. The radius-multiplier can be chosen to give a more or less tight constraint. The aim of this set of runs was to produce novel gaits with longer walks through stricter balance enforcement.

Combined enhancement The two enhancements were also combined, using the most successful parameters (control-cost multiplier and its delay, and balance-circle radius-multiplier) for each. The aim of this was to test if such a combination would achieve superior results.

Robustness to Action Noise

The MuJoCo 3D Humanoid-v1 environment contains a parameter for the standard deviation (*ac_noise_std*) of Gaussian noise to be added to the actions taken by walkers. To test the robustness of the evolved gaits, we evaluated evolved walkers with noise levels (*ac_noise_std*) from 0 (no noise) to 1, in order to observe the degradation of each major metric (average speed, distance travelled and episode time) until the walkers no longer achieved (lengths long enough to be typical of) stable gaits. The aim of this was to test whether or not the combined enhancement would result in more robust gaits, i.e. gaits with higher values in these metrics at higher levels of noise.

Results

Control-cost enhancement

Figure 2 shows the results for the default (d, no-enhancement) evolved behaviours and for walkers evolved with a control-cost multiplier of 0.25. 0.25 was chosen due to its promising results from initial testing of several multiplier values from 0.2 to 5. The multiplier was applied throughout (xp25) or from timestep 150 (s150, an estimate of the time at which the successful default gaits were reaching cyclic motion) or from timestep 500 (s500, the halfway point for a full-length episode). Median speed (averaged over the time of each evaluation) in the s500 runs was more than twice that in the default runs, with the former distribution significantly higher than the latter (Mann-Whitney $U=112$, $n_1=n_2=20$, $p<0.05$ one-tailed). Median distance traveled in the s500 runs was also more than twice that in the default runs, with the former distribution significantly higher than the latter (Mann-Whitney $U=121$, $n_1=n_2=20$, $p<0.05$ one-tailed). For time per episode, all medians were the maximum value (1000) and no significant difference was found.

Figure 5 includes high-performing gaits produced by the default and s500 runs, without action noise. The default runs' gait (top-left) shows a shuffling behaviour based around the knee joints. The s500 runs' gait (top-right) also shows a shuffling gait using the knee joints but, unlike the gaits produced by the default, the knees cross over, putting one leg in front of the other. This improved gait may be due to reduced importance of keeping energy expenditure low (at least per timestep rather than per unit distance) once a walker has reached a cyclic motion.

Balance enhancement

Figure 3 shows the results for the default (d) evolved behaviours and for walkers evolved with the balance enhance-

ment, with balance-circle radius-multipliers 1.00 (rp100), 0.75 (rp75), 0.5 (rp50) and 0.25 (rp25). Median speeds in the rp75 and rp50 runs were more than twice that in the default runs, with the former distributions significantly higher than the latter (rp75 Mann-Whitney $U=97$, $n_1=n_2=20$, $p<0.05$ one-tailed; rp50 Mann-Whitney $U=92$, $n_1=n_2=20$, $p<0.05$ one-tailed). Median distances traveled in the rp75 and rp50 runs were also more than twice that in the default runs, with the former distributions significantly higher than the latter (rp75 Mann-Whitney $U=91$, $n_1=n_2=20$, $p<0.05$ one-tailed; rp50 Mann-Whitney $U=99$, $n_1=n_2=20$, $p<0.05$ one-tailed). For time per episode, all medians were the maximum value (1000) except for the rp25 runs, which failed to produce a successful gait; the rp75 distribution was significantly higher than the default (Mann-Whitney $U=150$, $n_1=n_2=20$, $p<0.05$ one-tailed).

Figure 5 includes (bottom-left) a high-performing gait produced by the rp75 (0.75 radius-multiplier) runs, without action noise. The walker drags itself forward with one leg while pumping its arm for momentum, a behaviour previously unseen in the gaits produced by the default.

Combined enhancement

Figure 4 shows the results for the default (d) evolved behaviours, for walkers evolved with the most successful control-cost and balance-enhancement parameters (s500 and rp75) and for those evolved with the two enhancements combined (s500 combined with rp75). Median speed and median distance in the combined-enhancement were again more than twice those in the default runs, with the former distributions significantly higher than the latter (speed Mann-Whitney $U=79$, $n_1=n_2=20$, $p<0.05$ one-tailed, distance Mann-Whitney $U=74$, $n_1=n_2=20$, $p<0.05$ one-tailed). No significant increase in median speed or median distance was found between the s500 or rp75 runs and the combined-enhancement runs, although all four one-tailed p-values were below 0.15, suggesting that further runs might be useful to better investigate whether or not there is a significant difference here; however, it seems unlikely that any difference in median (between s500 and combo or between rp75 and combo) would be large even if significant. This may be because the two enhancements work in opposite directions, in that one is a reduction in control cost, allowing greater movement, while the other is a restriction on movement. For time per episode, the combined-enhancement runs matched the rp75 runs in having all gaits reach 1000 timesteps, although it is possible that increasing the 1000-timesteps limit could reveal a difference.

In the absence of noise, the highest-performing combined-enhancement gaits showed no noteworthy novelties, with all featuring either wide-legged shuffling with no leg crossover or single-leg dragging gaits. An example of the former can be seen in figure 5 (bottom-right).

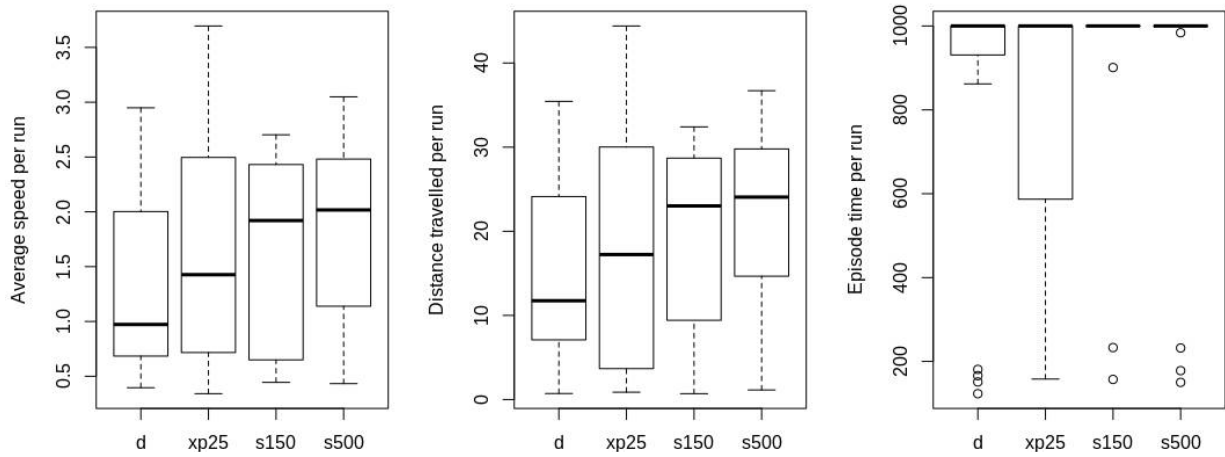


Figure 2: Results for the default evolved behaviours and for walkers evolved with the 0.25 control-cost multiplier throughout, from 150 timesteps and from 500 timesteps: speed (left), distance (middle) and time (right) for each set of 20 runs.

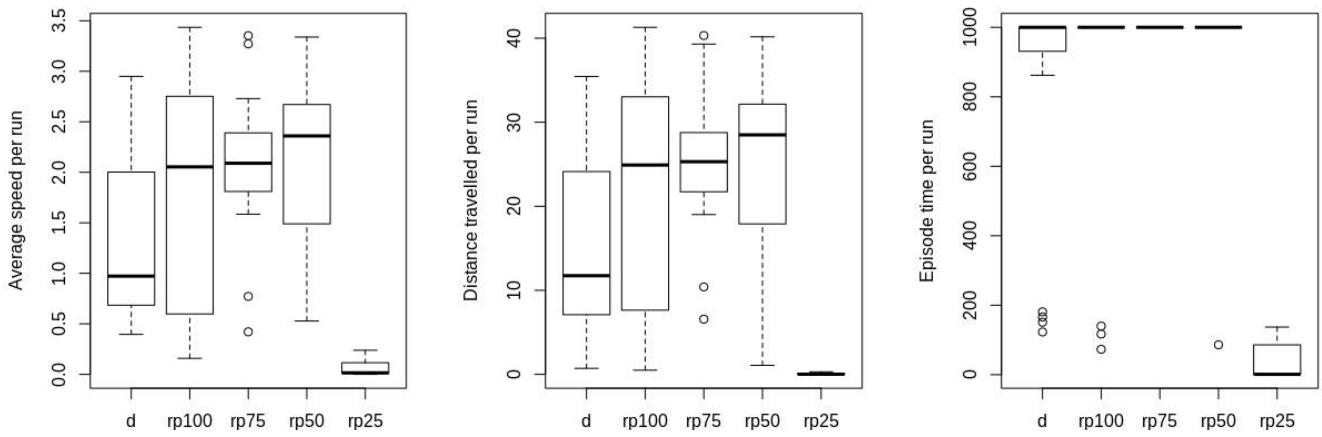


Figure 3: Results for the default evolved behaviours and for walkers evolved with balance-circle radius-multipliers 1.00, 0.75, 0.5 and 0.25: speed (left), distance (middle) and time (right) for each set of 20 runs.

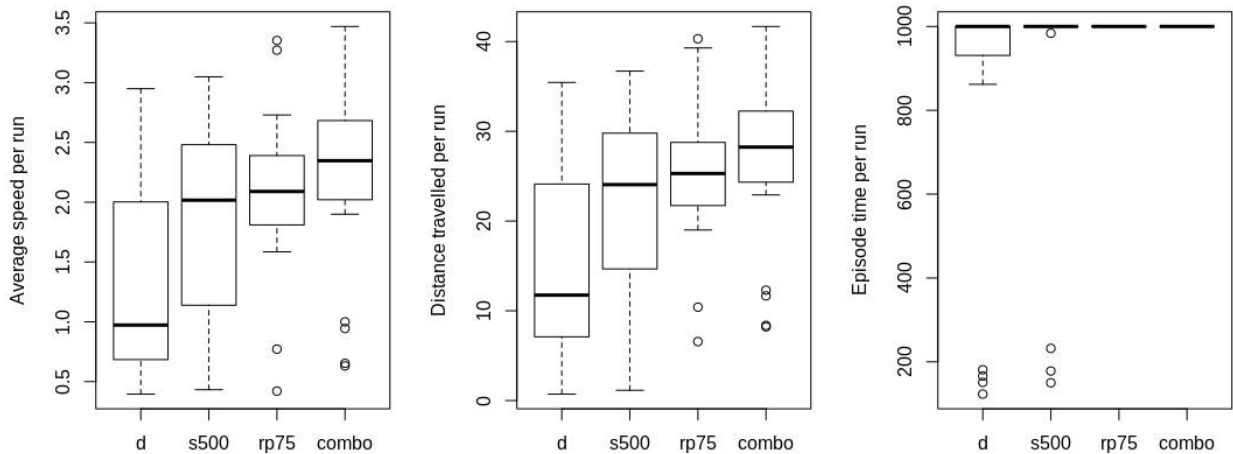


Figure 4: Results for the default evolved behaviours and for walkers evolved with the 0.25 control-cost multiplier from 500 timesteps, with the 0.75 balance-circle radius-multiplier and with a combination of the two: speed (left), distance (middle) and time (right) for each set of 20 runs.



Figure 5: Gaits without noise. Frame order is from left to right and top to bottom. In a high-performing gait from the default runs (top-left) the biped shuffles by alternating its knees in an unnatural motion. In a high-performing gait from the s500 runs (0.25 multiplier from 500 timesteps, top-right) the biped puts one leg in front of the other in succession, with a much wider range than the default's shuffling behaviours. In a high-performing gait from the rp75 runs (bottom-left) the biped pulls itself forward on one leg and pumps one arm for momentum, something previously unseen. In a high-performing gait from the combo runs (bottom-right) the biped shuffles similarly to the default gait, but with a wider spread of the legs. (The darkness at the end of the figures occurs as the humanoids walk out of the range of the white floor texture.) Videos of these gaits can be found at <https://github.com/KeeleBenJa/bipedal-methods>.

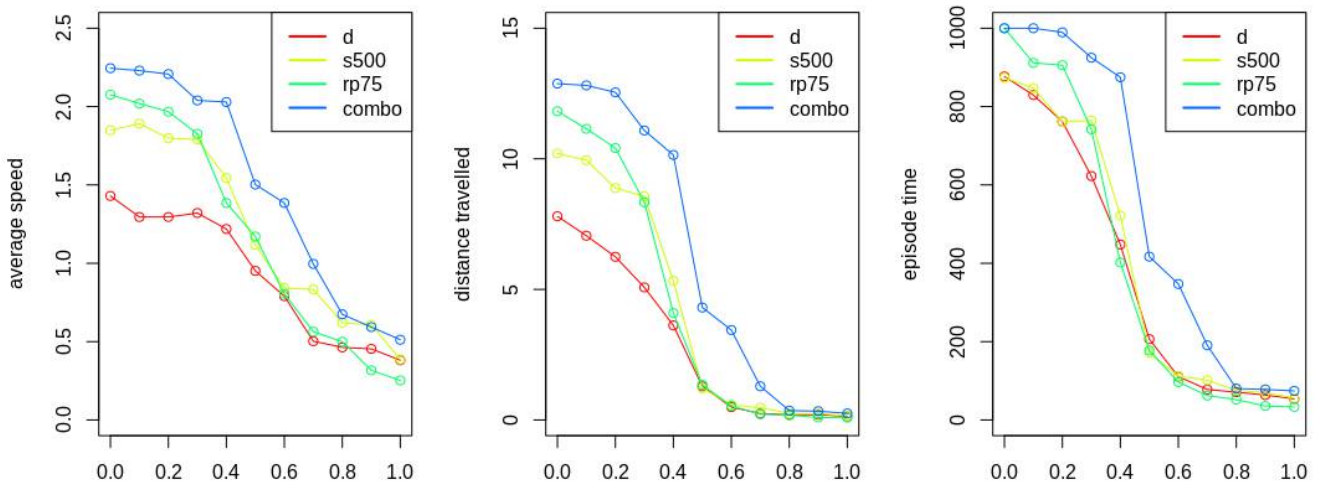


Figure 6: Results from scaling action noise from 0 to 1 for the default, 500 delay, 0.75 radial multiplier and combination of the two: speed (left), distance (middle) and time (right) averaged over each set of 20 runs.

Robustness to Action Noise

Figure 6 shows the degradation of average speed, distance travelled and time per episode with increasing levels of action noise. The combined-enhancement runs were much more robust to action noise than the default, particularly around noise=0.4. The rp75 and s500 runs produced intermediately robust gaits (i.e. more so than default but less so than for the combined-enhancement) at low levels of action noise but performed no better than default at medium to high levels of noise.

Figure 7 provides a closer look at the noise results at the 0.4 level, at which the combined-enhancement runs exhibited consistently superior results. In contrast to the without-noise results, the combined-enhancement results now show large, statistically significant improvements over the individual control-cost and balance enhancements. Median speed in the combined-enhancement runs was more than twice that in the default runs and the combined-enhancement distribution was significantly higher than the default runs (Mann-Whitney $U=79$, $n_1=n_2=20$, $p<0.05$ one-tailed), the s500 runs ($U=130$, $p<0.05$ one-tailed) and the rp75 runs ($U=99$, $p<0.05$ one-tailed). Median distance in the combined-enhancement runs was more than twice that in the default, s500 and rp75 runs, and the combined-enhancement distribution was significantly higher than the default runs (Mann-Whitney $U=53$, $n_1=n_2=20$, $p<0.05$ one-tailed), the s500 runs ($U=88$, $p<0.05$ one-tailed) and the rp75 runs ($U=70$, $p<0.05$ one-tailed). Median episode time in the combined-enhancement runs was also more than twice that in the default, s500 and rp75 runs, and the combined-enhancement distribution was significantly higher than the default runs (Mann-Whitney $U=67$, $n_1=n_2=20$, $p<0.05$ one-tailed), the s500 runs ($U=74$, $p<0.05$ one-tailed) and the rp75 runs ($U=62$, $p<0.05$ one-tailed).

Figure 8 shows the four previous high-performing gaits under noise level 0.4. The default (top-left), s500 (top-right, 0.25 control-cost multiplier from 500 timesteps) and combined-enhancement (bottom-right) runs produced similar gaits to figure 5 but with much wider motions, flailing limbs more. The rp75 (bottom-left, 0.75 radius-multiplier) gait arches the walker's back a little more but otherwise remains more stable, continuing to pump its arm.

Conclusions

Two fitness-function enhancements were tested to assess their affects on the speed, distance and duration of 3D Humanoid-v1 walks evolved using a replica of Salimans *et al.*'s evolution strategy (Salimans *et al.*, 2017). The first enhancement was to reduce control cost within the fitness function. When control cost was reduced to a quarter of the default, from the 500th timestep (the halfway point for a full-length episode), median speed and median distance both doubled. The most notable gait produced using this enhancement had a more pronounced stance and swing phase,

putting one leg in front of the other in a clearer fashion than any gait produced using the default fitness function.

The second enhancement was a balance enhancement, terminating walking when the torso's center of mass moved outside a circle centered at the midpoint between the walker's feet. When the circle's radius was 0.75 times current distance between the midpoint and (either) foot, median speed and median distance both doubled. In the most notable gait produced using this enhancement, the walker drags itself forward with one leg while pumping its arm for momentum, a behaviour previously unseen in the gaits produced by the default fitness function.

The two enhancements were also combined, using the most successful parameters for each (as above). Median speed and median distance were again double those in the default runs. However, there was little difference in these medians between the combined-enhancement and individual-enhancement runs and, in the absence of noise, the combined-enhancement gaits showed no noteworthy novelties.

To test the robustness of the evolved gaits, we evaluated evolved walkers with the addition of noise to their actions. The combined-enhancement gaits were much more robust to action noise than the default. The individual-enhancement gaits were intermediately robust at low levels of action noise but performed no better than default at medium to high levels of noise. In contrast to the without-noise results, the combined-enhancement gaits showed large, statistically significant improvements over those from the individual control-cost and balance enhancements, with median speed, distance and time two to five times those of the default and individual-enhancement gaits. This shows that the two enhancements synergise well to produce gaits that are robust to noise in their actions.

Future work will include further investigation into the reason for the large improvements exhibited by the combined-enhancement gaits in the presence of action noise, in contrast to the without-noise results. Removing or increasing the 1000-timesteps episode time limit would help to establish how much further the combined-enhancement can improve results, with and without noise. This will also help in establishing an upper bound for the optimal delay for the control-cost enhancement. OpenAI have ported the Humanoid walking task (and other tasks) from MuJoCo to the open-source Bullet Physics Engine and added more challenging tasks involving running toward a flag (<https://openai.com/blog/roboschool/> <https://github.com/openai/roboschool>). Despite the emergence of improved locomotive gaits, the resulting controllers do not yet produce gaits as lifelike as that demonstrated in (Reil and Husbands, 2002), or achieve as complex behaviour as that demonstrated in (Heess *et al.*, 2017). We intend to evaluate our individual and combined enhancements on these more challenging tasks and to investigate the use of

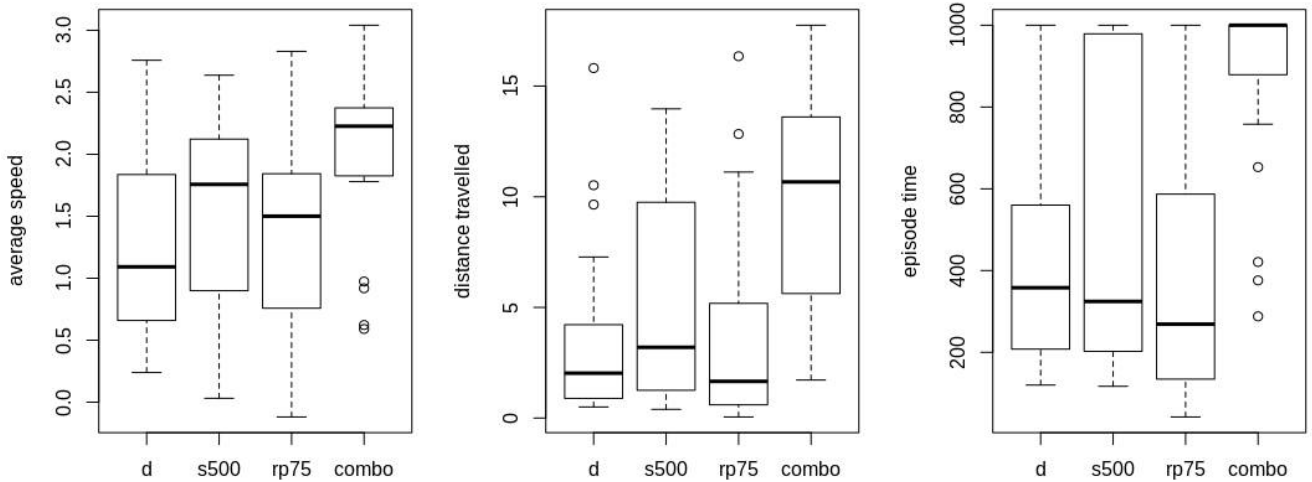


Figure 7: Results with action noise=0.4; for the default evolved behaviours and for walkers evolved with the 0.25 control-cost multiplier from 500 timesteps, with the 0.75 balance-circle radius-multiplier and with a combination of the two: speed (left), distance (middle) and time (right) for each set of 20 runs.



Figure 8: Gaits with noise = 0.4, from the controllers shown (without noise) in figure 5. Frame order is from left to right and top to bottom. In the default-runs gait (top-left) the biped shuffles by alternating its knees in a more erratic way than before. In the s500-runs gait (top-right) the biped still puts one leg in front of the other, but much more loosely. In the rp75-runs gait (bottom-left) the biped uses one arm for momentum and is not affected too heavily by the noise, only bending its back more. In the combo-runs gait (bottom-right) the biped shuffles on its knees with a wide stance, making more flailing motions than previously. (The darkness at the end of the figures occurs as the humanoids walk out of the range of the white floor texture.) Videos of these gaits can be found at <https://github.com/KeeleBenJa/bipedal-methods>.

deeper networks in all of these tasks.

Acknowledgements

Our code is based on code from Uber AI Labs (Petroski Such et al., 2018; Conti et al., 2018, <https://github.com/uber-research/deep-neuroevolution>), which was in turn based on code from OpenAI (Salimans et al., 2017, <https://github.com/openai/evolution-strategies-starter>).

We thank these researchers and organisations for making their code available. Our code can be found at <https://github.com/KeeleBenJa/bipedal-methods>.

References

- Beyer, H.-G. and Arnold, D. V. (2001). Theory of evolution strategies – a tutorial. In *Theoretical aspects of evolutionary computing*, pages 109–133. Springer.
- Bramble, D. M. and Lieberman, D. E. (2004). Endurance running and the evolution of Homo. *Nature*, 432(7015):345.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *Preprint arXiv:1606.01540*.
- Conti, E., Madhavan, V., Petroski Such, F., Lehman, J., Stanley, K. O., and Clune, J. (2018). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, pages 5027–5038. Curran Associates, Inc.
- Emanuel Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.
- Fukuda, T., Komata, Y., and Arakawa, T. (1997). Stabilization control of biped locomotion robot based learning with GAs having self-adaptive mutation and recurrent neural networks. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 217–222. IEEE.
- Gökçe, B. and Akin, H. L. (2010). Parameter optimization of a signal-based omni-directional biped locomotion using evolutionary strategies. In *Robot Soccer World Cup*, pages 362–373. Springer.
- Guertin, P. A. (2013). Central pattern generator for locomotion: anatomical, physiological, and pathophysiological considerations. *Frontiers in neurology*, 3:183.
- Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., Riedmiller, M., et al. (2017). Emergence of locomotion behaviours in rich environments. *Preprint arXiv:1707.02286*.
- Hewes, G. W. (1961). Food transport and the origin of hominid bipedalism. *American Anthropologist*, 63(4):687–710.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653.
- Lehman, J., Chen, J., Clune, J., and Stanley, K. O. (2018). ES is more than just a traditional finite-difference approximator. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 450–457. ACM.
- Petroski Such, F., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., and Clune, J. (2018). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *Preprint arXiv:1712.06567*.
- Reil, T. and Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168.
- Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *Preprint arXiv:1703.03864*.
- Taga, G., Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological cybernetics*, 65(3):147–159.
- Tassa, Y., Erez, T., and Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE.
- Van der Noot, N., Ijspeert, A. J., and Ronsse, R. (2015). Biped gait controller for large speed variations, combining reflexes and a central pattern generator in a neuromuscular model. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6267–6274. IEEE.
- Vukobratović, M. and Borovac, B. (2004). Zero-moment point – thirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173.
- Winter, D. A. (1991). *Biomechanics and motor control of human gait: normal, elderly and pathological*. University of Waterloo Press.

The Limits of Lexicase Selection in an Evolutionary Robotics Task

Jared M. Moore¹ and Adam Stanton²

¹School of Computing and Information Systems,
Grand Valley State University, Allendale, MI, USA

²School of Computer Science and Mathematics,
Keele University, Keele, ST5 5BG, UK
moorejar@gvsu.edu, a.stanton@keele.ac.uk

Abstract

Agents exhibiting generalized control are capable of solving a theme of related tasks, rather than a specific instance. Here, generalized control pertains to the locomotive capacity of quadrupedal animats, evaluated when climbing over walls of varying height to reach a target. In prior work, we showed that Lexicase selection is more effective than other evolutionary algorithms for this wall crossing task. Generalized controllers capable of crossing the majority of wall heights are discovered, even though Lexicase selection does not sample all possible environments per generation. In this work, we further constrain environmental sampling during evolution, examining the resilience of Lexicase to the impoverished conditions. Through restricting the range of samples at given points in time as well as fixing environmental exposure over fractions of evolutionary time, we attempt to increase the ‘adjacency’ of environmental samples, and report on the response of the Lexicase algorithm to the pressure of this reduced environmental diversity. Results indicate that Lexicase is robust, producing viable agents even in considerably challenging conditions. We also see a positive correlation between the number of tiebreak events that occur and the success of individuals in a population, except in the most limiting conditions. We argue that the increased number of tiebreaks is a response to local maxima, and the increased diversity resulting from random selection at this point, is a key driver of the resilience of the Lexicase algorithm. We also show that in extreme cases, this relationship breaks down. We conclude that tiebreaking is an important control mechanism in Lexicase operation, and that the breakdown in performance observed in extreme conditions indicates an inability of the tiebreak mechanism to function effectively where population diversity is unable to reflect environmental diversity.

Introduction

Generalized control remains a challenge in the field of robotics. Systems must be capable of addressing a task broadly, rather than solve only specific environmental configurations. In this paper, we use a wall crossing task, see Figures 1 and 2, wherein an agent is presented with a navigation target placed on the opposite side of a wall. It must then locomote over the obstacle and reach the target within a fixed amount of time. The wall height varies between environments, with 100 different gradations challenging a controller to learn the general task of ‘locomotion with wall crossing’.

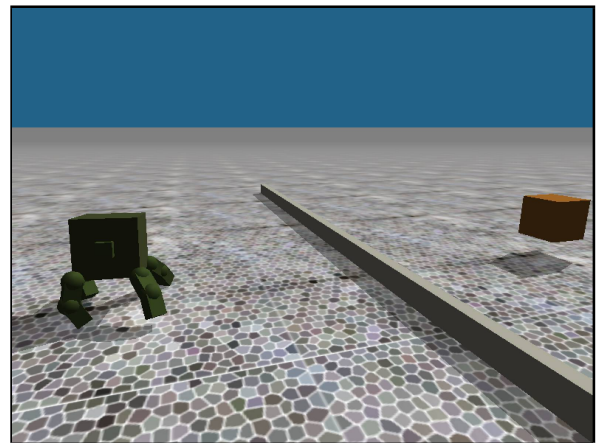


Figure 1: Neurocontrollers evolve to produce gaits that guide the quadrupedal animat (left) across a wall (center) and towards a target, represented by the box (right). Graphic originally presented in Moore and Stanton (2018)

In previous work (Moore and Stanton, 2018), we demonstrated evolved artificial neural network (ANN) controllers, generalized across different wall heights, using the Lexicase selection algorithm originally proposed by Spector (2012). We primarily focused on performance of the evolved agents, identifying a set of parameters for Lexicase that promote effective controllers. Even though there are 100 wall heights, simulating agents across 10 unique wall heights per generation was sufficient to evolve generalized behavior. From these results, we speculated that the advantages of Lexicase (aside from the exposure of species to a heterogeneous set of environments over evolutionary time) are due to automatic maintenance of population diversity facilitated by tiebreak events occurring as part of the selection process. During an individual selection event in Lexicase, a tiebreak occurs when two or more individuals have similar performance in all environments used in the selection event. A random selection is then performed choosing a parent from the set of tied individuals.

In this study, we attempt to elucidate characteristics of Lexicase that govern its behavior in extreme situations. This

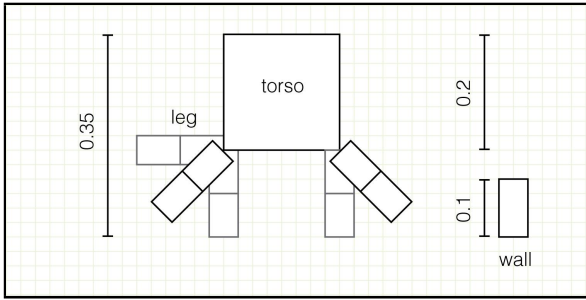


Figure 2: Schematic of robot and wall, illustrating relative sizes. Wall is shown at maximum height; robot leg shown at initial position (black) and at leg-torso joint extremes (gray).

allows us to identify areas where generalized task performance by evolved controllers begins to break down. Specifically, we use two restrictions to alter key aspects of the Lexicase algorithm. First, we constrain the sampling range of environments (i.e. wall heights) used to evaluate individuals within a generation. This increases *adjacency*, that is, the difference in wall height between environments chosen during selection. We hypothesize that this will hinder the generalizability of controllers as adjacency might also play a role in the success, or failure, of individual runs. Second, we limit how often over evolutionary time we reposition our sampling range, forcing populations to experience a limited subset of the 100 total wall heights for a number of consecutive generations.

Results show that in this task, even with fairly significant restrictions, Lexicase selection is still a resilient evolutionary algorithm. Most treatments are able to evolve at least some level of generalization with performance degrading as stronger restrictions are placed on Lexicase selection. There is a positive correlation in the number of tiebreaks and performance of the best individual for a given replicate. Although Lexicase does well to preserve mean population diversity, in this task, it does not appear to drive generalization performance across treatments.

Background

Evolutionary robotics (ER) optimizes robotic systems by employing concepts from biological evolution in a digital system (Nolfi and Floreano, 2000). ER has been employed across robotic systems to optimize both body structure alone (Auerbach and Bongard, 2010; Cheney et al., 2013; Collins et al., 2018) or together with control (Jelisavcic et al., 2018; Kriegman et al., 2018). Evolved robots have been transferred to reality (Ruud et al., 2016) although the reality gap remains a persistent issue (Stanton, 2018; Koos et al., 2010; Jakobi, 1998). While enhancing performance of systems is often the primary goal, one outstanding area is quantifying the impact that different components of a robotic system (Powers et al., 2018), or the underlying evolutionary algorithm (Dolson and Ofria, 2018), have on evolved systems.

Lexicase selection was originally introduced for many-objective problems in genetic programming by Spector (2012) where it has been effective on a variety of problems (Helmuth et al., 2014). By replacing traditional selection methods in a genetic algorithm, Lexicase uses a variable number of objectives to perform selection between individuals. One of the potential strengths for Lexicase selection is its influence on population diversity, as proposed by Helmuth et al. (2016). In addition, Lexicase may also preserve specialists in the population as originally noted by Pantridge et al. (2018). An overview of the Lexicase selection algorithm and our specific modifications for this paper are described in the next section.

Methods

The animat, simulation environment, and wall-crossing task are continuations of previous work originally reported with respect to Lexicase selection in Moore and Stanton (2017). Specific design parameters are detailed in that work and its references. Here, we report our animat’s genome parameters and simulation configuration for completeness.

Animat Figure 1 shows the quadrupedal animat in the wall crossing environment, and Figure 2 shows a schematic of the animat and obstacle to illustrate relative proportion. The torso is a cube with four legs, each placed at a corner. Hip joints can swing both horizontally and vertically. Knees move only on one axis allowing for flexion and extension. Table 1 shows the physical dimension, mass, and joint ranges for the animat.

Torso Dimension	$0.2 \times 0.2 \times 0.2$
Leg Component Dimension	$0.075 \times 0.05 \times 0.05$
Torso Mass	2.0
Leg Component Mass	0.5
Hip Vertical Axis	range $[-\frac{\pi}{4}, \frac{\pi}{4}]$
Hip Horizontal Axis	range $[0, \frac{\pi}{2}]$
Knee Horizontal Axis	range $[0, \frac{\pi}{2}]$
Max Torque	0.125

Table 1: Physical parameters of robot. Adapted from Moore and Stanton (2017, 2018).

Controllers are feed-forward ANNs with one output per leg motor. Each ANN output specifies a joint angle translated to motion using a Proportional Derivative (PD) controller (Reil and Husbands, 2002) which calculates a torque to apply to a joint for the given timestep based on Equation 1,

$$T = k_s \times (\theta_d - \theta) - k_d \dot{\theta} \quad (1)$$

where T is torque, k_s and k_d are spring and damper constants, θ_d is the ANN output angle, θ the present joint and $\dot{\theta}$ the angular change from the last timestep. $k_s = k_d = 0.5$.

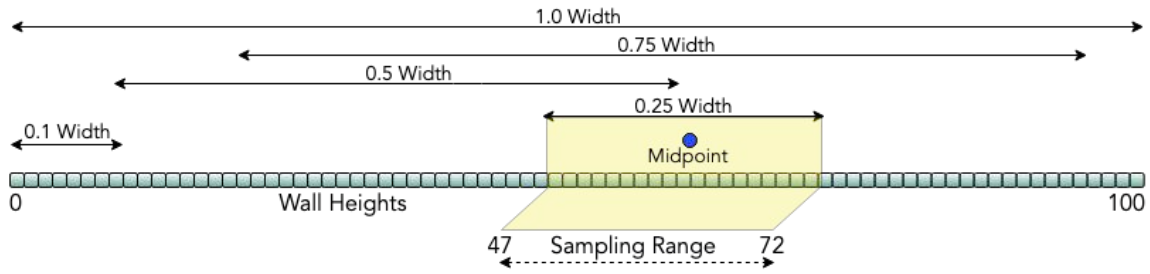


Figure 3: Five separate sampling widths are explored. The process of selecting environments proceeds as follows: 1. Pick a midpoint within the range of environments such that the maximum and minimum environments from the midpoints stay within the range of 0 to 100 exclusive. 2. The midpoint establishes the minimum and maximum wall heights that can be sampled for use in the Lexicase selection event.

Simulation The Open Dynamics Engine (ODE) (Smith, 2013) version 0.15.2 was used. ODE is a real-time rigid body physics engine handling the interaction between components of the animat, as well as interactions with the ground and wall.

Wall Crossing Task The goal for an individual is to reach a target placed on the opposite side of a single wall from the animat’s starting position. Animats must thus evolve a legged gait to move towards a target as well as an ability to cross a wall. Wall height is fixed within a simulation, but can be chosen from a range of 100 possible wall heights. The height is determined by the specific sampling determined by the Lexicase algorithm, described in the next section.

Controller effectiveness is characterized as one of five possible behaviors: (1) reached target, (2) crossed wall, (3) stuck on wall, (4) reached wall, and (5) did not reach wall.

Evolutionary Algorithm and Lexicase Selection Lexicase selection is applied to a subset of individuals sampled from the overall population between generations. To compare individuals, an objective is selected randomly within the subset ranked by performance. If two or more individuals are tied, selection moves on to the next objective only on the subset of tied individuals. If multiple individuals remain after all objectives have been used, a random selection is applied to the remaining individuals and recorded as a tiebreak event. For a full description of the Lexicase selection algorithm employed in this paper refer to Moore and Stanton (2017).

Originally proposed for GP, Lexicase selection only considered two individuals to be tied in an objective if they were exactly equal. Moore and McKinley (2016) introduced a *fuzz factor* to ease the consideration of ties to include individuals who are within a specified threshold of performance. La Cava et al. (2016) concurrently proposed ϵ -Lexicase selection as an improvement for real-valued objectives. For continuity with our previous work, in this paper we use a *fuzz factor* of 10%, that is two individuals are tied if one individual is within 10% of the performance of the other. Ongoing work by others however should use the ϵ -Lexicase terminology.

Sampling Width Figure 3 shows the various widths employed for sampling in this strategy. Treatments use widths of 0.1, 0.25, 0.5, 0.75, and 1.0. For example, 0.1 width allows for environments to be sampled within a range of 10 environments whereas 0.75 width allows sampling from 75 environments. The process of selecting environments within a given width proceeds as follows.

1. Select a midpoint within the range of 0 - 100 that will not cause the minimum or maximum of the width to exceed the range of environments.
2. Midpoint establishes the minimum and maximum.
3. Sample n environments uniformly from within the range according to the number of environments specified by the treatment. ($n = 10$ in this paper.)

Figure 3 shows how a range could be specified for the 0.25 width. A midpoint of 60 establishes a range of 47-72 (floating point numbers are rounded down). A number of environments are then sampled from within this range establishing the objectives for Lexicase within that generation.

Monte Carlo Simulation of Sampling Figure 4 plots the expected sampling frequency of various widths and 10 environments for evaluation within a generation if the midpoint were randomly selected each generation. Each pair of width plots consists of 5000 distinct sampling events with 10 environments selected per sample. As the sampling width increases, the distance between selected wall heights increases presenting agents with a larger variety of wall heights. We hypothesize that this will help increase generalizability.

Measuring Diversity We quantify diversity by calculating the mean genotype for a population every generation with the following equation from (Moore and Stanton, 2018):

$$G^{gene} = \frac{1}{P} \sum_{i=1}^P I_i^{gene} \quad (2)$$

, where G is the mean genotype, P is the population size, and I_i^l is the l th gene of the i th individual).

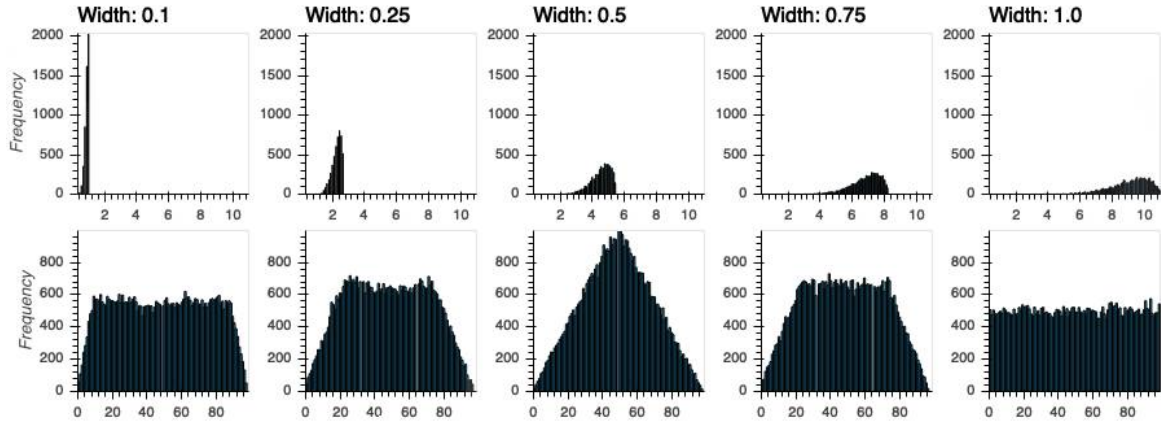


Figure 4: Simulation of 5000 sampling events per each of the sampling widths conducted in this study. Each sampling event draws 10 environments uniformly within a given sampling range discussed previously in Figure 3. Top row shows the average distance between wall heights (adjacency) in a sample. Bottom row shows the raw frequency that each wall height is sampled.

mean squared difference of each individual from the generation’s mean individual is calculated. This quantity is calculated for each individual, summed and averaged across the population. The value is then compressed to a single reportable measure in the following:

$$D = \frac{1}{P} \sum_{i=1}^P \left[\frac{1}{L} \sum_{l=1}^L (I_i^l - G^l)^2 \right] \quad (3)$$

where D is the diversity value and L is genotype length).

Treatments In prior work (Moore and Stanton, 2018), we sampled between 1 and 20 environments per generation to use in the Lexicase selection process. Results indicated that the range of 5 to 10 environments are optimal for this task. Therefore, all treatments conducted in this study simulate agents in 10 environments per generation. This also cuts down on simulation time when evolving individuals as we only need to simulate 10 out of 100 possible environments to evaluate individuals in a selection event. Treatments consist of 20 replicate runs, each initialized with a random starting seed. Each replicate is evolved for 5,000 generations with a population size of 50 individuals.

Treatments are defined by two factors. First, there are five separate sampling widths (0.1, 0.25, 0.5, 0.75, and 1.0) described in the previous sections. Second, the midpoint of a sampling width is moved according to a specified braking interval. Possible intervals are 1, 2, 5, 10, 100, 500, 2500, and 5000 generations. For example, a braking interval of 5 generations (abbreviated as 5B) means that we select a midpoint that establishes the sampling range for the next 5 generations. A new set of environments are sampled every generation, however, braking effectively limits how often the midpoint moves around the wall height space. Some limiting of moving the midpoint might encourage a popu-

lation to specialize on a specific range of environments at the expense of not being exposed to other areas of the wall height space. At the conclusion of a braking interval, we then move the sampling range by selecting a new midpoint and evolving for another 5 generations. This process repeats until 5000 generations have elapsed. Once the evolutionary phase is complete, each individual in each population is evaluated in each environment (wall height), and these results are recorded to gauge the generalizing capability of that particular controller.

Experiments and Results

The focus of this investigation is on aspects of Lexicase selection that enable generalized control. In prior work (Moore and Stanton, 2017, 2018) we have shown that Lexicase selection is an effective evolutionary algorithm for agents in this problem domain. Our results therefore are not centered on the question of “whether Lexicase can evolve effective individuals” but rather “which aspects of Lexicase selection might affect performance”.

Generalization We first examine the generalizability of the best individual per replicate per treatment across the 100 environments. Figure 5 plots the average performance of the best individual per treatment grouped by sampling width. Lower wall heights are not a challenge for any of the treatments as indicated by the brighter shading associated with wall crossing and reaching the target. Narrower widths and higher braking intervals also appear to have more individuals reach the objective attaining the highest fitness possible in these low wall heights compared to the 1.0 and 0.75 width treatments. We hypothesize that there may be an evolutionary pressure in these treatments to evolve highly effective behavior in the low wall heights to compensate for the generally poor performance in the higher walls.

Average Performance of the Best Ind Per Replicate Per Treatment

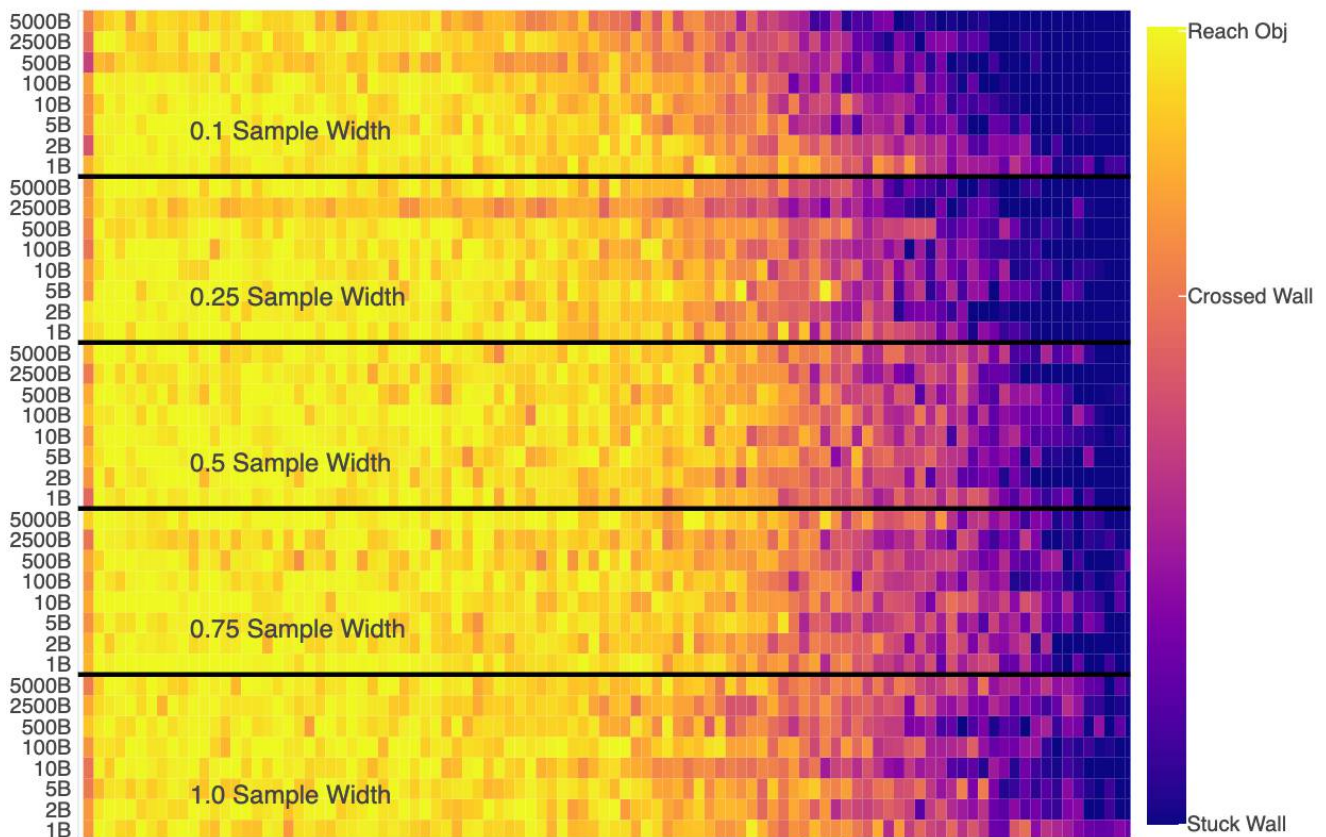


Figure 5: Average performance of the best individual per replicate by treatment during validation in each of the 100 environments. Treatments are grouped by the sampling width. Wall heights increase from left to right. Braking strategy increases from bottom to top in each width subgroup.

There is a decline in performance for wall heights above 50 with many treatments not able to cross the wall. Although the highest wall heights appear difficult for any treatment, the 1.0 sampling width group has the broadest generalization. As sampling width narrows, there is a subsequent decline as higher wall heights are more challenging. Physically these wall heights would be the most imposing for the animat, requiring gaits that lift the legs substantially to cross the wall.

In terms of sampling widths, 0.75 is not significantly different using a Wilcoxon rank-sum pairwise test compared to 1.0 counting the total number of successful wall crossings across wall heights for the best individual per replicate. As shown in Figure 4, sampling widths of 0.75 can still be expected to sample environments between 20 and 80 fairly uniformly. Generalizability declines in narrower widths (0.1, 0.25, 0.5), more drastically as the braking interval increases. The sum of the successful wall crossings for the best individual per replicate during validation for 0.25 and 0.1 sam-

pling width are significantly different than the 1.0, 0.75, and 0.5 sampling widths across braking intervals. The two factors of narrow sampling widths and long intervals evolving in the same range of environments prevent populations from evolving on a wide range of wall heights, apparently hindering generalizability.

Tiebreaks Figure 6 plots the number of tiebreaks in a replicate versus the count of wall heights with the target being reached for the best individual per replicate grouped by treatment. Within a specific braking regimen, treatments show a general increase in the number of tiebreaks as sampling width narrows. For the same sampling width this does not hold, as the number of tiebreaks does not vary due to the braking regimen. Observing the figure, it appears that there is generally a positive correlation between the number of tiebreaks and the total wall heights an individual solves for smaller braking values (1B, 2B, 5B, 10B, 100B) and wide sampling widths (1.0W, 0.75W, 0.5W). This begins to break-

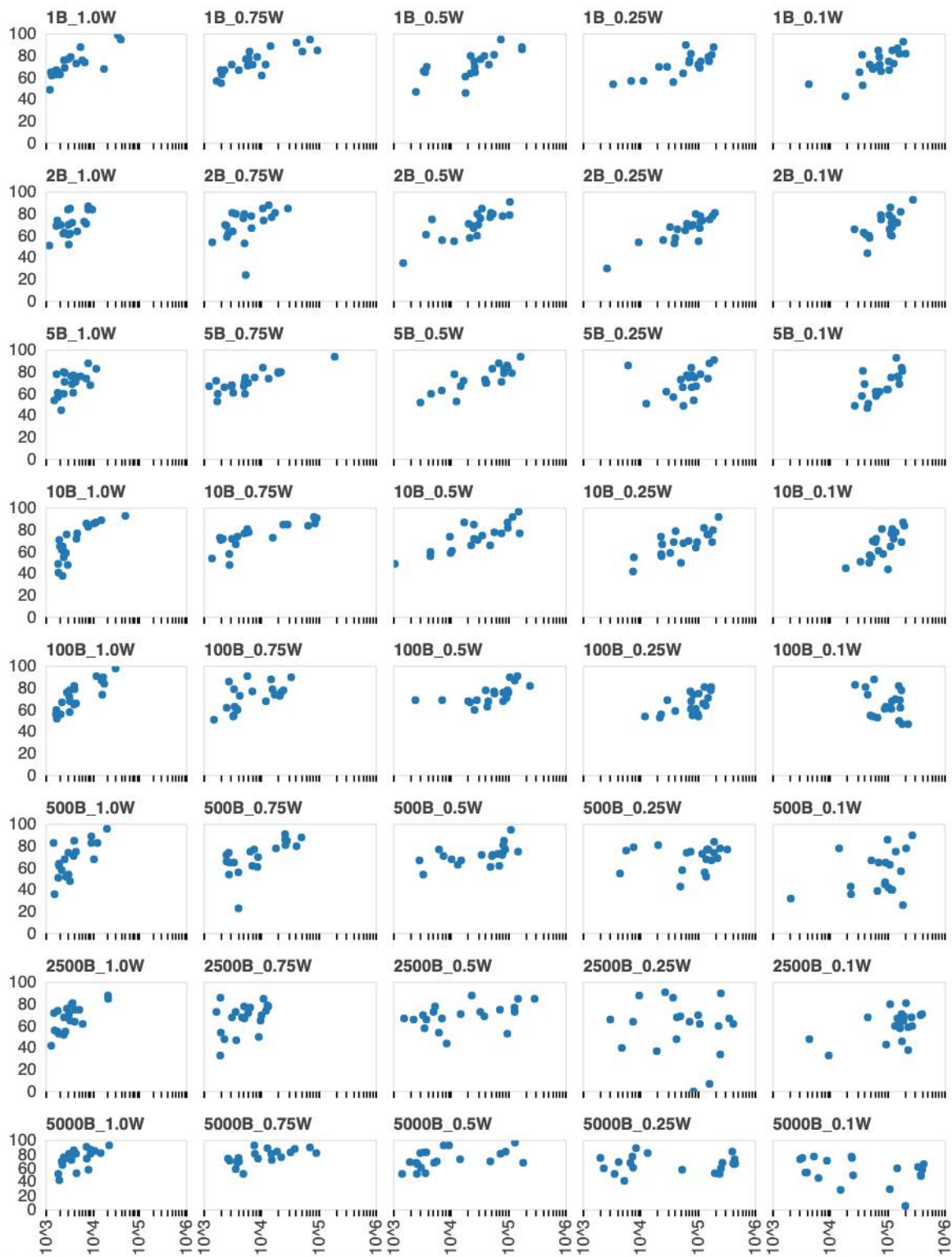


Figure 6: Number of tiebreaks in a replicate versus the number of environments where the best individual per replicate reached the objective, across treatments. Each sub-plot shows 20 replicates.

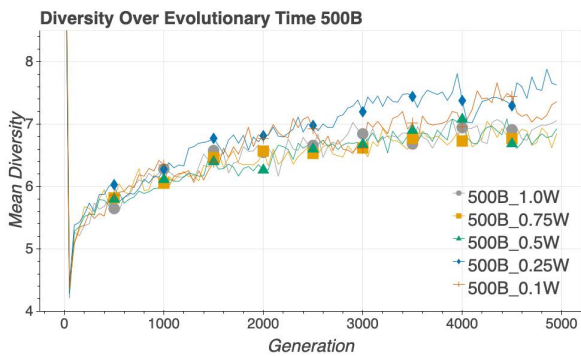


Figure 7: Mean population diversity across 20 replicates per treatment for the 500 braking treatments over evolutionary time.

down, becoming almost random, in the lower right plots as the combination of high braking and narrow width push the Lexicase algorithm to its limit. Here, the limited range of environments and lack of moving the sampling range restrict the total number of environments that a population is exposed to. Thus, individuals become specialized rather than evolving toward generalized performance.

Diversity Figure 7 plots the mean population diversity across replicates for the 500 braking interval treatments. Trends shown in this plot are similar for the other braking intervals. For widths 1.0, 0.75, and 0.5 there is no difference in mean population diversity. Mean population diversity is generally higher for widths 0.25 and 0.1. This trend holds for all braking intervals with a clear difference in mean diversity between (1.0, 0.75, 0.5 widths) and (0.25, 0.1 widths) that increases as the braking interval lengthens. We surmise that this is due to the reduction in environments that individuals are exposed to over evolutionary time. We note that in evolutionary algorithms, an increase in diversity is unusual where individuals are unable to discover regions of higher fitness and that normally the converse is true.

In terms of mean population diversity predicting generalization performance of the best individual per replicate we do not observe any relationship. Instead, mean population diversity tends to fall in the range of 6-8, with low variation in small braking intervals or wide sampling widths. As the sampling widths narrow and braking intervals increase, mean population diversity tends to increase as well as the variation between replicates. However, no positive or negative correlation exists between mean population diversity and generalizability of the best individual per replicate. This clarifies one open question that we had raised in Moore and Stanton (2018), where we hypothesized that diversity might predict performance. We do note however, that for all treatments, mean population diversity still falls in the range of 6-8, consistent with earlier results.

Conclusions and Future Work

In prior work, we identified ‘good’ parameter configurations that show Lexicase selection is the preferred algorithm for this task across a number of different algorithms investigated. Here, we identify specific factors and characteristics of the Lexicase algorithm that do, and do not, lead to generalized behavior on the wall crossing task. While there is a degradation in generalization performance as braking intervals increase and sampling widths narrow, Lexicase is still quite resilient. Even the lowest generalization treatments still on average are successful at crossing the wall in 60 environments.

Tiebreaks are a mechanism to add a random selection event during evolution. Tiebreak events occur due to multiple individuals being considered equal in the wall heights used during selection. When looking at generalization performance, there is a positive correlation between the number of tiebreaks over evolutionary time and generalization performance of the best individual per replicate until the restrictions placed on Lexicase become too strong. The correlation then disappears. As the sampling widths narrow, and braking increases, we see tiebreaks rise due to the higher adjacency of wall heights, see top of Figure 4. In these restricted treatments, populations are exposed to narrow ranges of wall heights which are likely highly related. From our results, it appears that some number of these random selection events are beneficial when Lexicase is relatively unhindered. Whereas, mean population diversity does not appear to be a predictor of performance.

In future work, we plan to continue pursuing questions regarding the underlying factors that make Lexicase selection effective. We plan to further expand the range of tasks an animat is evaluated in to include those that are not thematically related. That is, multiple tasks versus variations on a similar themed task.

Acknowledgements

We would like to acknowledge the contributions of Thomas Helmuth, Bill La Cava, Edward Pantridge, and Lee Spector for their discussions during the formulation of this work. We would also like to thank Keele University and Grand Valley State University for their continued support of this research by providing computational resources and travel funding.

References

- Auerbach, J. E. and Bongard, J. C. (2010). Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of the Twelfth International Conference on Artificial Life*, pages 451–458, Odense, Denmark.
- Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2013). Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th Annual Conference on Genetic*

- and *Evolutionary Computation*, pages 167–174, Amsterdam, The Netherlands. ACM.
- Collins, J., Geles, W., Howard, D., and Maire, F. (2018). Towards the targeted environment-specific evolution of robot components. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 61–68, Kyoto, Japan. ACM.
- Dolson, E. and Ofria, C. (2018). Ecological theory provides insights about evolutionary computation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 105–106, Kyoto, Japan. ACM.
- Helmuth, T., McPhee, N. F., and Spector, L. (2016). The impact of hyperselection on lexicase selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '16, pages 717–724, Denver, Colorado, USA. ACM.
- Helmuth, T., Spector, L., and Matheson, J. (2014). Solving uncompromising problems with Lexicase selection. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.
- Jakobi, N. (1998). Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In *Proceedings of the First European Workshop on Evolutionary Robotics*, pages 39–58, Paris, France. Springer-Verlag.
- Jelisavcic, M., Roijers, D. M., and Eiben, A. (2018). Analysing the relative importance of robot brains and bodies. In *Proceedings of the 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 327–334, Tokyo, Japan. MIT Press.
- Koos, S., Mouret, J. B., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 2010 ACM Genetic and Evolutionary Computation Conference*, pages 119–126, Portland, Oregon, USA. ACM.
- Kriegman, S., Cheney, N., Corucci, F., and Bongard, J. C. (2018). Interoceptive robustness through environment-mediated morphological development. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 109–116, Kyoto, Japan. ACM.
- La Cava, W., Spector, L., and Danai, K. (2016). Epsilon-lexicase selection for regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 741–748, Denver, Colorado, USA. ACM.
- Moore, J. M. and McKinley, P. K. (2016). *A Comparison of Multiobjective Algorithms in Evolving Quadrupedal Gaits*, pages 157–169. Springer International Publishing, Aberystwyth, UK.
- Moore, J. M. and Stanton, A. (2017). Lexicase selection outperforms previous strategies for incremental evolution of virtual creature controllers. In *Proceedings of the 14th European Conference on Artificial Life*, pages 290–297, Lyon, France. MIT Press.
- Moore, J. M. and Stanton, A. (2018). Tiebreaks and diversity: Isolating effects in lexicase selection. In *Proceedings of the 16th International Conference on the Simulation and Synthesis of Living Systems*, pages 590–597, Tokyo, Japan. ACM.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. The MIT Press.
- Pantridge, E., Helmuth, T., McPhee, N. F., and Spector, L. (2018). Specialization and elitism in lexicase and tournament selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1914–1917, Kyoto, Japan. ACM.
- Powers, J., Kriegman, S., and Bongard, J. (2018). The effects of morphology and fitness on catastrophic interference. In *Proceedings of the 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 606–613, Tokyo, Japan. MIT Press.
- Reil, T. and Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168.
- Ruud, E. L., Samuelsen, E., and Glette, K. (2016). Memetic robot control evolution and adaptation to reality. In *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, Athens, Greece. IEEE.
- Smith, R. (2013). Open Dynamics Engine, <http://www.ode.org/>.
- Spector, L. (2012). Assessment of problem modality by differential performance of Lexicase selection in genetic programming: A preliminary report. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 401–408, Philadelphia, Pennsylvania, USA. ACM.
- Stanton, A. (2018). Stochastic ontogenesis in evolutionary robotics. In *Artificial Life Conference Proceedings*, pages 214–221. MIT Press.

Improve Quadrupedal Locomotion with Actuated or Passive Joints?

Jared M. Moore¹ and Anthony J. Clark²

¹School of Computing and Information Systems,
Grand Valley State University, Allendale, MI, USA

²Computer Science Department,
Missouri State University, Springfield, MO, USA
moorejar@gvsu.edu

Abstract

Animals interact with their environment softly through interaction of muscles, tendons, and rigid skeleton. By incorporating flexibility, they reduce ground impact forces and improve locomotive efficiency. Flexibility is also beneficial for robotic systems, although it remains challenging to implement. In this paper, we explore the addition of passive flexibility to a quadrupedal animat; we measure the impact of flexibility on both locomotive performance and energy efficiency of movement. Results show that spine and lower limb flexibility can significantly increase distance traveled when compared to an animat with no flexibility. However, replacing passively flexible joints with actively controlled joints evolves more effective individuals with similar efficiency. Given these results, the number of joints and joint configuration appear to drive performance increases rather than just the addition of passive flexibility.

Introduction

Animals exhibit a remarkable ability to adapt locomotion to varying conditions. Gaits are driven by responses from the central nervous system and the morphology of the organism itself. Often, characteristics of the musculoskeletal system, such as elasticity of the tendons, contribute to their movements. For example, Alexander and Vernon (1975) found that large hind limb tendons in kangaroos and wallabies allow them to efficiently conserve energy during locomotion. Muscle-tendon systems in bipeds and quadrupeds act as energy storage contributing to running gaits in vertebrates (Alexander, 1984) while the spine has been shown to conserve energy during galloping (Alexander et al., 1985).

Robotic systems typically comprise rigid-body components, connected with single degree-of-freedom (DOF) actuators such as servo motors and linear actuators. These systems are often bioinspired, drawing upon the morphology and behaviors of biological organisms. Mechanical components, however, lack the flexibility of their natural counterparts, so compliant components are often added to these systems. Ackerman and Seipel (2013) added elasticity through springs, reducing the energetic cost of legged locomotion in a hexapedal robot. Increasing flexibility in the hexapod damped vertical movement of the torso as compared to a fully rigid-body robot. The addition of a flexi-

ble spine increased locomotive performance in a quadruped animat (Moore et al., 2015). It remains an open question whether the performance gains were due to flexibility or the increase in the DOF in the animat. Would performance increase if we replaced passive components with actively actuated joints?

In our preliminary study (Moore and Clark, 2018), we found that additional degrees of freedom improve the walking speed of our animats. In this paper, we further investigate the differences between passive and active joints, and we explore the impacts of these configurations on efficiency as well as speed. We conduct seven treatments with different animat configurations. We first examine performance, based on distance traveled, of a quadruped with legs actuated by hinge joints and no passive flexibility. Next, we increase the flexibility of the animat by adding sliding joints to the lower limbs (acting as shock absorbers) and a flexible spine. Finally, we replace the passive sliding joints with actively controlled hinge joints in the lower limbs, maintaining the DOF but reducing flexibility. We investigate both a passively flexible and active spine for this new animat.

We find that the addition of passive flexibility, whether it is in the spine or legs, significantly increases the distance traveled over a fully rigid-body quadruped. The highest performing platform with passive flexibility includes a flexible spine and lower limb sliders. Still, replacing flexibility in the lower sliders with actively controlled hinge joints produces the furthest distances traveled. The most effective individuals across all treatments include both an actively controlled spine and actively controlled lower hinge joints. Efficiency does not significantly change between passively flexible and actively controlled joints. This suggests that while flexibility can increase the performance of a robotic system, the real factor for performance increases is likely a combination of increasing DOF in the animat and joint configurations.

Background and Related Work

In evolutionary robotics (ER) (Nolfi and Floreano, 2000; Doncieux et al., 2015) both control and morphology of robotic systems are optimized using concepts derived from biological evolution. Evolutionary approaches are well

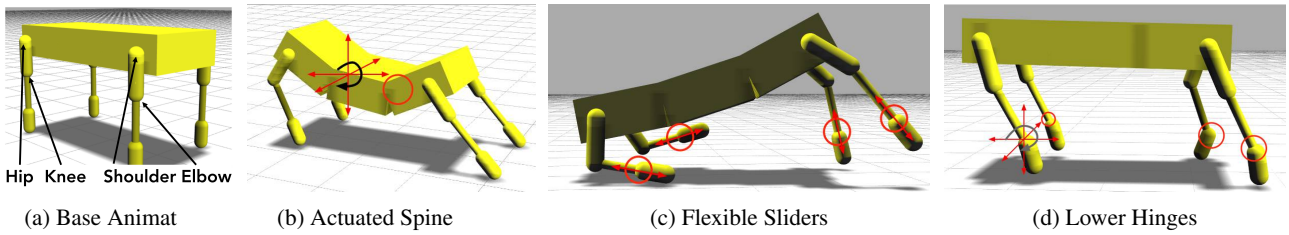


Figure 1: Different joint configurations explored in this study. (a) Base quadruped configuration with no sliders on the lower limbs. (b) Spines can be active or passive with three torso segments connected by two hinge joints. Axis of rotation indicated on rear spine, circle indicates second spine joint. (c) Flexible sliders on the lower limbs allow for a dampening of interaction with the ground. Note the compressed slider in the right rear leg compared to the extended slider in the front right leg. Arrows indicate axis of slider translation. (d) Actively controlled hinge joints on the lowest joint in each limb. Rear right leg indicates axis of rotation for all lower hinge joints.

suiting to problems where an algorithm for deriving the optimal solution is not known *a priori* (Li and Miikkulainen, 2014). They have proven effective for optimizing controllers in wheeled (Fischer and Hickenbotham, 2011) and legged (Cully and Mouret, 2013; Stanton and Channon, 2013) systems. Evolutionary algorithms are especially useful when exploring morphology (Funes and Pollack, 1998). The body plays an important role in movement, performing an innate control prior to engagement of higher-level control from the brain (Valero-Cuevas et al., 2007). Auerbach and Bongard (2010) demonstrated that optimizing brain and body together produces effective systems exploiting integration between aspects of morphology and control.

Passive flexibility plays an important role in biological organisms, helping to reduce the energetic cost of locomotion by storing energy in spring-like tendons (Baudinette et al., 1992; Ruina et al., 2005). In robotics, Rieffel et al. (2010) demonstrated that even in the absence of a higher level controller the spring systems comprising tensegrity robots can be harnessed to realize locomotion. In traditional rigid-body robots, compliant joints enable robotic systems to mimic the passive flexibility of animals (Vanderborght et al., 2013). Passive compliance in robotic systems improves climbing ability (Seo and Sitti, 2013) and swimming (Clark et al., 2014). While augmenting robotic systems with passive flexibility can improve performance, it remains difficult to determine whether it is due to the elasticity itself, or if it is perhaps the additional DOF added to the system. Evolutionary methods coupled with simulation enable exploring many configurations that would not be practical with physical systems.

Methods

Simulation Environment The Open Dynamics Engine (ODE) (Smith, 2013) is used to conduct simulations. ODE is a 3D rigid-body physics simulation engine that models forces such as gravity, friction, and collisions between objects. Actively controlled actuators include single DOF hinges and linear motors, among others. In ODE we model

flexibility by connecting rigid-bodies with spring-like joints that can be active or passive. The environment is a flat, high-friction surface. Animats are evaluated for 10 seconds of simulation time with a timestep of 0.005 seconds.

Quadruped Animat The base quadruped animat is shown in Figure 1a. The torso is composed of three segments connected by fixed joints. Each leg is three segments with hinge joints at the hip and knee. In the base treatment, the joint connecting the lowest component to the mid-leg is fixed, effectively creating a short upper segment and a longer lower leg segment.

Other animats are derived from the base treatment by adding a passive or active spine combined with a passive leg slider or active leg hinge. Figure 1b shows a quadruped animat with spine joints that are passive or actuated depending on treatment. Here, we replace the rigid joints in the torso with hinge joints that actuate along the lateral planes of the animat. Figure 1c shows the addition of flexible slider joints between the two lower limb segments. They compress during locomotion acting as shock absorbers. In the figure, the slider on the right rear leg is at maximum compression. Figure 1d shows a quadruped animat with actively controlled hinge joints on the lowest joint of each leg.

Controller The controller in this experiment is a conventional sinusoid, and each joint has its own evolved sinusoid parameters. Joint control signals are determined by the time of the simulation, evolved control signal modifiers per joint, and the maximum force output potential for each type of joint. Equation 1 generates the movement command for a single joint at each timestep in the simulation.

$$\sin(-2\pi ft + (2\pi(\phi_{leg} + \phi_{joint}))) \quad (1)$$

f is the oscillation frequency common across the joints of the animat determining how quickly the sine wave oscillates represented by a real-value ranging from 0 to 2, t is the current simulation time, ϕ_{leg} and ϕ_{joint} are the phase

offsets. Phase offsets are one of 16 set values ranging from 0 to 1.875, which corresponds to shifting the phase of the oscillating signal in $\frac{1}{8}$ increments. Each leg has its own phase offset relative to the common signal (ϕ_{leg}). Each joint type (e.g. shoulders, elbows, hips and knees) has an offset as well (ϕ_{joint}). Together, the two offsets produce common control signals for the rear legs and the front legs. Each leg pair (front and rear) can have a common behavior (specified by ϕ_{joint}) that is then shifted temporally by ϕ_{leg} . Thus a common behavior for the front or rear legs can be out of phase, similar to walking in animals as symmetry and coordination can evolve between limbs.

We evolve forces acting on the shoulder, elbow, hip, and knee, allowing for joints to be entirely passive or force limited. Under this configuration, the oscillating signal sent to a joint and its maximum force output potential determine the response of the joint to a command. For example, if a joint evolves a low force output, it will passively flex under the force of gravity and not actively assist in locomotion. However, should the joint evolve a high force output, it will not deviate from its specified motion even when large external forces are applied, such as when making contact with the ground where such forces would typically hinder normal actuation of a joint.

Passively flexible joints are governed by spring and damper constraints parameterized in ODE as ERP and CFM. ERP values evolve in the range of 0.4 to 1.0. CFM values evolve from 0.0001 to 0.15. Together, the two parameters specify the stiffness and damping of a joint. In general, high CFM values and low ERP values result in flexible joints whereas the opposite lead to stiff joints.

Treatments Seven treatments are conducted in this study.

1. *No Sliders (NS)* - 8 DOF
Base quadruped animat with no passive flexibility.
2. *Flexible Spine, No Sliders (FSpNS)* - 10 DOF
Flexible spine, no sliders on lowest joint.
3. *Rigid Spine, Flexible Sliders (FS)* - 12 DOF
Flexible sliders on lowest joint with a rigid spine.
4. *Flexible Spine, Flexible Sliders (FSpFS)* - 14 DOF
Flexible spine and sliders.
5. *Rigid Spine, Active Lower Hinge (HL)* - 12 DOF
Actively controlled hinges on lowest joint with a rigid spine.
6. *Flexible Spine, Active Lower Hinge (HLFSp)* - 14 DOF
Actively controlled hinges on lowest joint and flexible spine.
7. *Active Spine, Active Lower Hinge (HLASp)* - 14 DOF
Actively controlled hinges on lowest joint and spine.

The first four treatments (*NS*, *FSpNS*, *FS*, *FSpFS*) evaluate varying degrees of passive flexibility in the animat. *NS* is the base quadruped with 8 DOF and no flexibility. *FSpNS* adds spine flexibility, increasing the DOF to 10.

FS adds flexible lower sliders to each leg while maintaining a rigid spine. There are 12 DOF in this animat. *FSpFS* combines both flexible spine and flexible sliders with 14 DOF; 6 DOF more than the base animat configuration.

The final three treatments (*HL*, *HLFSp*, *HLASp*) replace the lower passive slider joints with actively controlled hinge joints. *HL* has a rigid spine and 12 DOF similar to the *FS* treatment. *HLFSp* has a passively flexible spine 14 DOF and the spine configuration of the *FSpFS* treatment. *HLASp* has an actively actuated spine and 14 DOF.

Evolutionary Algorithm For each treatment, we evolve 120 individuals over 4,000 generations using the DEAP framework (De Rainville et al., 2012) with a conventional genetic algorithm. DEAP is an open-source framework implementing many common evolutionary algorithms. 20 replicate runs, each seeded with a unique random number, are conducted per treatment. Fitness is the horizontal Euclidean distance from the starting point to the center of the torso after 10 seconds of simulation time. Selection of a parent is performed through a tournament of 4 randomly chosen individuals. Crossover is performed (two parents selected) with a rate of 50% and mutation of 4% per gene. An individual replicate took approximately 5 hours on a Blade system parallelized across 24 cores and a clock speed of 2GHz.

Genome Table 1 lists the genes in this study. The composition of each genome varies depending on the treatment and is indicated in the right columns of the table. All treatments have 10 genes for generating the control signals and 5 genes for joint forces (back, shoulder, elbow, hip, knee) described previously. Depending on the combination of genes included, treatments range in value from 15 (*NS*) to 23 (*FSpFS*, *HLFSp* and *HLASp*) genes.

Table 1: Genes defining the quadruped animat. “l/r sym” denotes that left/right symmetry is enforced.

Description	# Genes	Treatments						
		NS	FSpNS	FS	FSpFS	HL	HLFSp	HLASp
Oscillation Frequency	1	•	•	•	•	•	•	•
Max Joint Velocity	1	•	•	•	•	•	•	•
Phase Offset Per Leg	4	•	•	•	•	•	•	•
Joint Phase Offset (l/r sym)	4	•	•	•	•	•	•	•
Joint Max Force (l/r sym)	5	•	•	•	•	•	•	•
Slider Flexibility (l/r sym)	4			•	•			
Spine Joint Flexibility	4		•		•		•	
Hinge Phase Offset (l/r sym)	2					•	•	•
Hinge Max Force (l/r sym)	2					•	•	•
Active Spine Phase Offset	2							•

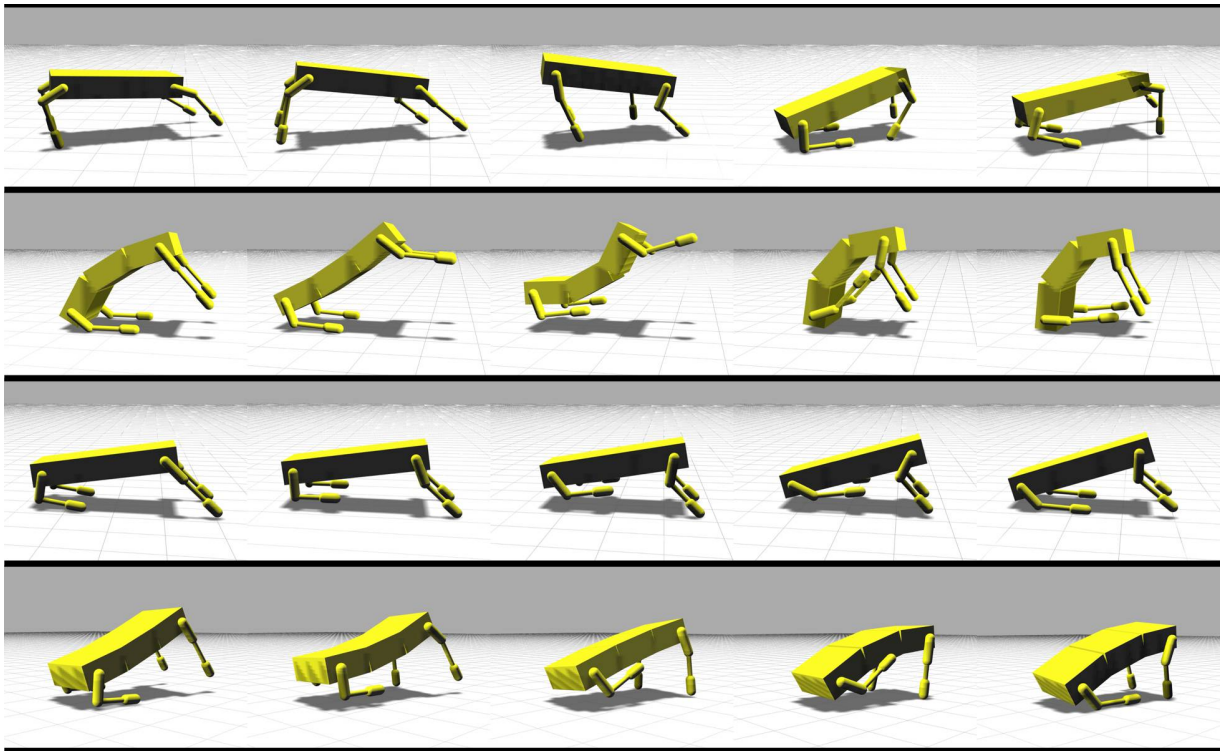


Figure 2: Sample evolved gaits. (Top) Galloping gait evolved in the *NS* treatment. (Top-Mid) Hopping gait from the *FSpNS* treatment. (Bottom-Mid) Bounding gait from the *FS* treatment. (Bottom) Bounding gait from the *FSpFS* treatment.

Results

Bounding, galloping, canters, and trots evolve across treatments, and samples of evolved gaits from all treatments can be seen at <https://youtu.be/UCNxJ3pmmkc>. Our analysis focuses on two main questions. First, does a quadruped animat with passive flexibility significantly outperform a fully rigid-body animat? If so, what combination of lower leg slider, spine, or both, leads to the most effective individual? Second, how does replacing passive joints with actively actuated hinges alter performance?

Passive Flexibility Figure 2 highlights a few of the gaits that evolve across the initial treatments. Figure 3 plots the maximum distance traveled at each generation averaged across twenty replicates per treatment over evolutionary time. Shaded regions represent the 95% confidence intervals for each treatment. For reference, the animat’s body length is 3 units. A fully rigid body animat (*NS*) yields the lowest distances traveled. The other treatments show a significant improvement over the *NS* treatment with a flexible spine (*FSpNS*) slightly better. Flexible sliders (*FS*) on the lower limbs increase distance traveled, with the combination of spine and slider flexibility (*FSpFS*) yielding the farthest traveling individuals.

Figure 4 plots the distribution of the farthest traveling individual per replicate across the treatments. We conduct the following statistical tests using a Wilcoxon Rank-Sum Test

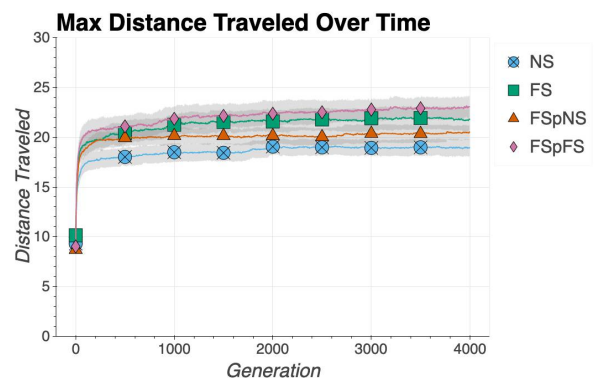


Figure 3: Maximum distance traveled across twenty replicates per treatment over evolutionary time for the passive flexibility treatments. Shaded areas represent the 95% confidence intervals.

performed on the distribution of farthest traveling individual per replicate for each treatment. Pairwise results across all treatments can be seen in Table 2. For the two treatments without sliders, *NS* and *FSpNS*, there is no significant difference in performance ($p = 0.07$). Whereas both treatments with sliders have significantly higher performance. *FS* versus *NS* ($p < 0.01$), *FS* versus *FSpNS* ($p < 0.03$), and ($p < 0.01$) for *FSpFS* versus *NS* and *FSpNS*. There is no significant difference between *FSpFS* and *FS* ($p = 0.08$).

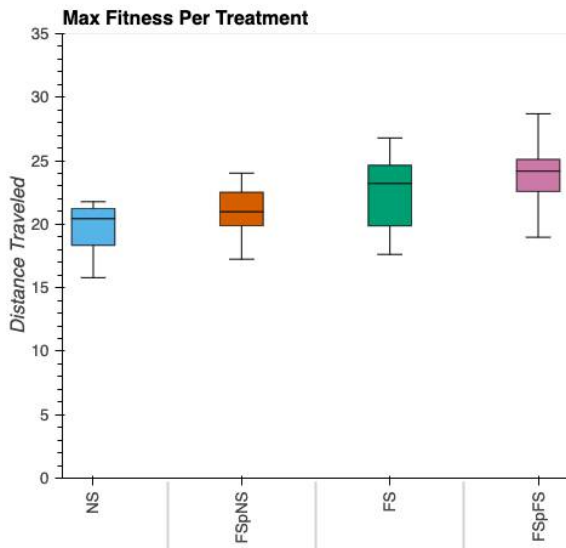


Figure 4: Distribution of the farthest traveling individual per replicate across the passive flexibility treatments.

It appears that the addition of spine flexibility alone is not enough to significantly improve locomotive performance even though two additional DOF have been added to the animat. Although the distance traveled of the best individuals for the *FS* treatment is significantly higher, performance also varies considerably. Animats with both flexible spine and flexible sliders exhibit the highest average performance across these treatments. However, the *FSpFS* treatment has 6 more DOF than the *NS* treatment. Differences in performance might not solely be due to flexibility and instead could be attributed to the possible additional behaviors the increased mobility allows.

Hinge Joints By replacing the flexible sliders with actively controlled hinge joints, we address the question of whether the increase in DOF or flexibility drives performance increases. We chose a passive slider and active hinge as these are the most effective actuator for their respective control type. Figure 5 shows three variations of bounding gaits that evolve across the three active lower hinge treatments. Figure 6 shows the maximum distance traveled per generation over evolutionary time for all seven treatments conducted in this study. *HL* has similar evolutionary performance as compared to the best of the passive flexibility treatments, *FSpFS*. The addition of the lower hinge joints, with spine mobility, produces the farthest traveling individuals observed. A combination of actively controlled lower hinges and a passive spine flexibility (*HLFSp*) outperforms just replacing the sliders with hinges while an active spine leads to the highest average maximum distance traveled across all twenty replicates.

Figure 7 plots the distribution of the farthest traveling individual across replicates for all seven treatments conducted in this study. Changing the lower joint from pas-

sive sliders (*FS*) to actively controlled hinges (*HL*) does not significantly increase distance traveled ($p = 0.12$). Furthermore, there is no significant difference between the *FSpFS* and *HL* treatments. Alone, actively controlled hinges don't generate significant improvements in performance compared to the flexible slider treatments. Further modifying the active lower hinge animat by adding either a flexible spine (*HLFSp*) or an active spine (*HLASp*) does significantly improve performance over the flexible slider treatments.

Table 2: Wilcoxon Rank-Sum Test comparing fitnesses of the farthest traveling individuals per replicate across treatments.

	<i>NS</i>	<i>FSpNS</i>	<i>FS</i>	<i>FSpFS</i>	<i>HL</i>	<i>HLFSp</i>	<i>HLASp</i>
<i>NS</i>	-	= 0.07	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
<i>FSpNS</i>		-	< 0.03	< 0.01	< 0.01	< 0.01	< 0.01
<i>FS</i>			-	= 0.08	= 0.12	< 0.01	< 0.01
<i>FSpFS</i>				-	= 0.88	= 0.01	< 0.01
<i>HL</i>					-	< 0.01	< 0.02
<i>HLFSp</i>						-	= 0.10
<i>HLASp</i>							-

Performance increases appear to be due to a combination of the number of DOF and the use of actively controlled hinge joints versus flexible sliders. The top two treatments, *HLFSp* and *HLASp*, have 14 DOF and are significantly better than any other treatment. Replacing flexible sliders with a hinge joint does not significantly increase performance, but it maintains similar performance to the *FSpFS* treatment, which has 14 DOF compared to *HLs* 12 DOF. The two lowest performing treatments, *NS* and *FSpNS*, have 8 and 10 DOF, respectively. Increases in distance traveled for the animat configurations in this study appear to be influenced more by the DOF, and addition of actively controlled joints, than the addition of passive flexibility.

Efficiency Flexibility in natural organisms can lower the energetic cost of locomotion (Alexander, 1984). In this study, our sole objective is to maximize distance traveled, but efficiency might differ between animats because of including passive flexibility. We measure efficiency as the distance traveled divided by the total power expended through actively controlled joints in an animat. Total power is the summation of force exerted by each actively controlled joint at each time step as reported by the physics engine. Passive joints replicate spring systems and therefore are not included in the total power calculation. Figure 8 plots the distribution of efficiency for the farthest traveling individual in each replicate across treatments. Here, there appears to be little difference in efficiency. Table 3 lists the pairwise Wilcoxon-Rank sum tests comparing efficiency between treatments. While some pairs are significantly different, there is predominantly no significant difference in efficiency across treat-

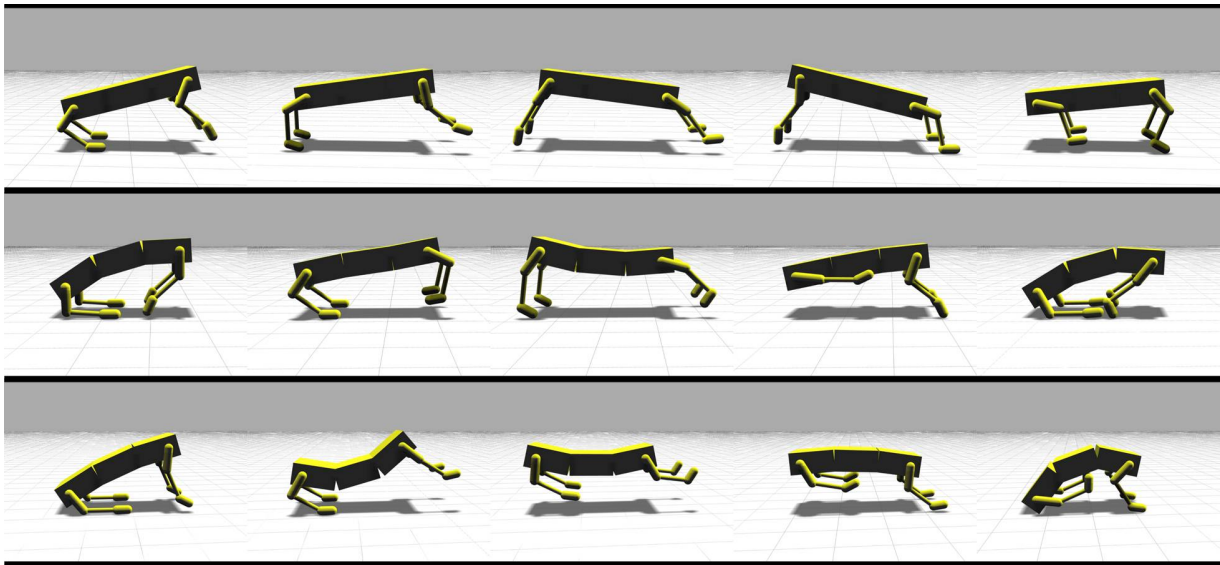


Figure 5: Sample evolved bounding gaits from the active hinge joint on lowest limb segment treatments. (Top) *HL* treatment. (Mid) *HLFSp* treatment. (Bottom) *HLASp* treatment.

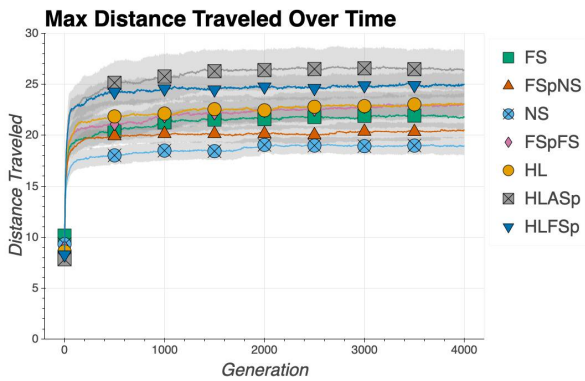


Figure 6: Maximum distance traveled across twenty replicates per treatment over evolutionary time for all treatments. Shaded areas represent the 95% confidence intervals.

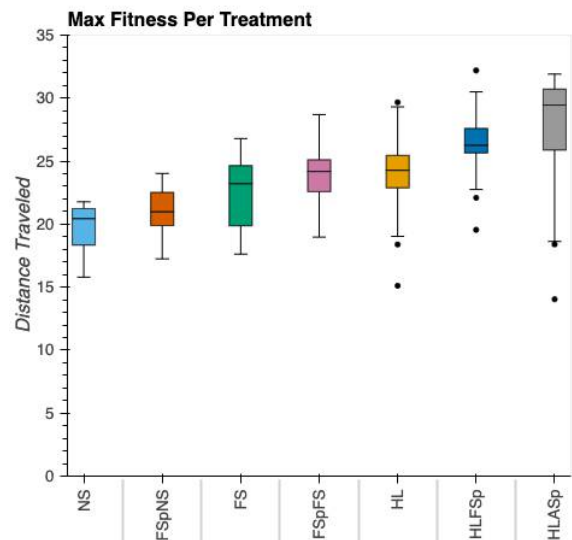


Figure 7: Distribution of the farthest traveling individual per replicate across the treatments. Adapted from Moore and Clark (2018).

ments. It appears that a flexible spine, compared to similar animats that have a rigid, or actively controlled spine, typically have lower efficiency, but this is only a significant difference in 1 out of 3 cases (*NS* vs. *FSpNS*, $p < 0.04$). When evaluating replicates on both efficiency and distance, we found that the single lowest performing individual is in the *HLASp* treatment. However, many of the highest performing individuals are also in the *HLASp* treatment. No clear advantage is apparent when passive flexibility is included in an animat in terms of efficiency.

Conclusions

In this paper, we investigated the effect of adding additional DOF to a rigid-body quadrupedal animat in terms of performance and efficiency. Adding additional DOF in the form of

passive flexibility or actively controlled hinge joints significantly increases performance in terms of distance traveled. However, efficiency remains unaffected when it is not directly included as a selective pressure during evolution.

Adding flexibility to the animat in the spine and lower sliders significantly increases distance traveled versus the base animat configuration. Animats with both spine and lower slider flexibility are the farthest traveling individuals among those with passive flexibility. This result supports those of earlier works (Seo and Sitti, 2013; Lessin et al., 2014; Clark et al., 2012) where flexibility aids in the per-

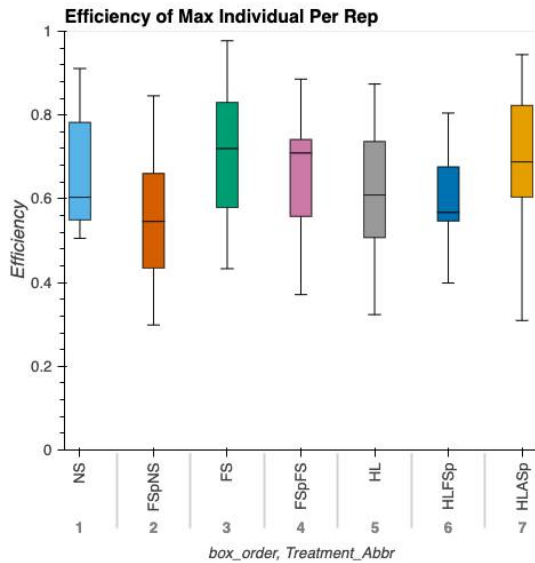


Figure 8: Efficiency of the best individual per replicate across treatments.

Table 3: Wilcoxon Rank-Sum Test comparing efficiency of the farthest traveling individual per replicate across treatments.

	NS	FSpNS	FS	FSpFS	HL	HLFSp	HLASp
NS	-	< 0.04	= 0.26	= 0.55	= 0.31	= 0.33	= 0.43
FSpNS		-	< 0.01	= 0.11	= 0.55	= 0.18	< 0.03
FS			-	= 0.19	< 0.02	< 0.05	= 0.60
FSpFS				-	= 0.37	= 0.28	= 0.09
HL					-	= 0.88	= 0.25
HLFSp						-	= 0.09
HLASp							-

formance of robotic systems. Thus, when building robotic systems, examining the incorporation of flexible components, such as springs, may be worthwhile to increase performance of a rigid-body robot. It also further clarifies and expands (Moore et al., 2015), in that flexibility is likely not the sole driver of performance increases, rather, both flexibility and an increase in the DOF positively impact performance.

Actively controlled joints lead to even higher performing individuals across all treatments. Replacing the lower sliding joints with actively controlled hinge joints results in the three highest performing treatments, out of the seven conducted. Adding spinal mobility in the form of a passively flexible spine, or actively controlled one, further increases distance traveled. This suggests that while passive flexibility improves performance over the base animat in this study, the increase is likely not due directly to including passive flexibility. Rather, the increase in DOF drives improvements in distance traveled.

In terms of robotics systems, incorporating flexibility could still be beneficial depending on the platform and problem constraints. Here, we find that including flexibility increases performance over a fully rigid-body robot with no flexibility. Although active control produces the highest performing individuals, it may be that a designer does not want further increases in control complexity nor the additional hardware (servos, wiring, batteries, etc) required to coordinate additional actively controlled actuators. Instead, a controller can use the dynamics of passive joints to improve performance as demonstrated here, and in other work mentioned previously. Furthermore, passive flexibility may reduce wear on other mechanical components by dampening locomotive forces.

Future extensions to this study will investigate how flexibility and active control affect performance in other animat platforms such as hexapods. We plan to introduce more complex high-level controllers such as artificial networks (ANN) to see how these features are integrated in control logic. Furthermore, we plan to expand the scope of evolvability in terms of morphological components to evolve, along with exploring multi-objective algorithms.

Source Code

The source code for running these experiments is provided at https://github.com/jaredmoore/Evo_Flex_Quadruped_Code.

Acknowledgements

The authors would like to thank our reviewers for their thoughtful commentary to improve this work as well as Philip K. McKinley, Craig P. McGowan, and Byron DeVries for their support in this work. Additional thanks to Grand Valley State University and Missouri State University for supporting this research through computing resources and financial support.

References

Ackerman, J. and Seipel, J. (2013). Energy efficiency of legged robot locomotion with elastically suspended loads. *IEEE Transactions on Robotics*, 29(2):321–330.

Alexander, R. and Vernon, A. (1975). The mechanics of hopping by kangaroos (macropodidae). *Journal of Zoology*, 177(2):265–303.

Alexander, R. M. (1984). Elastic energy stores in running vertebrates. *American Zoologist*, 24(1):85–94.

Alexander, R. M., Dimery, N. J., and Ker, R. F. (1985). Elastic structures in the back and their rôle in galloping in some mammals. *Journal of Zoology*, 207(4):467–482.

Auerbach, J. E. and Bongard, J. C. (2010). Dynamic resolution in the co-evolution of morphology and control.

- In *Proceedings of the Twelfth International Conference on Artificial Life*, pages 451–458, Odense, Denmark.
- Baudinette, R. V., Snyder, G. K., and Frappell, P. B. (1992). Energetic cost of locomotion in the tammar wallaby. *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology*, 262(5):R771–R778.
- Clark, A., Wang, J., Tan, X., and McKinley, P. (2014). Balancing performance and efficiency in a robotic fish with evolutionary multiobjective optimization. In *2014 IEEE International Conference on Evolvable Systems*, pages 227–234, Orlando, Florida, USA.
- Clark, A. J., Moore, J. M., Wang, J., Tan, X., and McKinley, P. K. (2012). Evolutionary design and experimental validation of a flexible caudal fin for robotic fish. In *Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, pages 325–332, East Lansing, Michigan, USA.
- Cully, A. and Mouret, J.-B. (2013). Behavioral repertoire learning in robotics. In *Proceedings of the 2013 ACM Genetic and Evolutionary Computing Conference*, pages 175–182, Amsterdam, Netherlands. ACM.
- De Rainville, F.-M., Fortin, F.-A., Gardner, M.-A., Parizeau, M., and Gagné, C. (2012). Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 85–92, Philadelphia, Pennsylvania, USA. ACM.
- Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. G. (2015). Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2(4).
- Fischer, V. and Hickinbotham, S. (2011). Evolving a metabolic subsumption architecture for cooperative control of the e-puck. *Memetic Computing*, 3(4):231–244.
- Funes, P. and Pollack, J. (1998). Evolutionary body building: Adaptive physical designs for robots. *Artificial Life*, 4(4):337–357.
- Lessin, D., Fussell, D., and Miikkulainen, R. (2014). Trading control intelligence for physical intelligence: Muscle drives in evolved virtual creatures. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, pages 705–712, Vancouver, BC, Canada. ACM.
- Li, X. and Miikkulainen, R. (2014). Evolving multimodal behavior through subtask and switch neural networks. In *Proceedings of The Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, New York, NY.
- Moore, J. M. and Clark, A. J. (2018). Bend and flex: Passive flexibility or active control in a quadruped animat. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 113–114, Kyoto, Japan. ACM.
- Moore, J. M., McGowan, C. P., and McKinley, P. K. (2015). Evaluating the effect of a flexible spine on the evolution of quadrupedal gaits. In *Proceedings of the 13th European Conference on Artificial Life*, pages 166–173, York, UK.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. The MIT Press.
- Rieffel, J. A., Valero-Cuevas, F. J., and Lipson, H. (2010). Morphological communication: Exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface*, 7(45):613–621.
- Ruina, A., Bertram, J. E. A., and Srinivasan, M. (2005). A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition. *Journal of Theoretical Biology*, 237(2):170–192.
- Seo, T. and Sitti, M. (2013). Tank-like module-based climbing robot using passive compliant joints. *IEEE/ASME Transactions on Mechatronics*, 18(1):397–408.
- Smith, R. (2013). Open Dynamics Engine, <http://www.ode.org/>.
- Stanton, A. and Channon, A. (2013). Heterogeneous complexification strategies robustly outperform homogeneous strategies for incremental evolution. In *Proceedings of the 12th European Conference on Artificial Life*, pages 973–980, Taormina, Italy.
- Valero-Cuevas, F., Yi, J.-W., Brown, D., McNamara, R., Paul, C., and Lipson, H. (2007). The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Transactions on Biomedical Engineering*, 54(6):1161–1166.
- Vanderborght, B., Albu-Schäffer, A., Bicchi, A., Burdet, E., Caldwell, D., Carloni, R., Catalano, M., Eiberger, O., Friedl, W., Ganesh, G., Garabini, M., Grebenstein, M., Grioli, G., Haddadin, S., Hoppner, H., Jafari, A., Laffranchi, M., Lefeber, D., Petit, F., Stramigioli, S., Tsagarakis, N., van Damme, M., ten Ham, R., Visser, L., and Wolf, S. (2013). Variable impedance actuators: a review. *Robotics and autonomous systems*, 61(12):1601–1614.

Ego-Noise Predictions for Echolocation in Wheeled Robots

Antonio Pico Villalpando¹, Guido Schillaci^{1,2}, Verena V. Hafner¹, Bruno Lara Guzmán³

¹*Adaptive Systems Group, Humboldt-Universität zu Berlin. Berlin, Germany*
{pivillaa, hafner}@informatik.hu-berlin.de

²*The BioRobotics Institute, Scuola Superiore Sant'Anna. Pisa, Italy*
guido.schillaci@santannapisa.it

³*Cognitive Robotics Group, Universidad Autónoma del Estado de Morelos. Cuernavaca, Mexico*
bruno.lara@uaem.mx

Abstract

Echolocation is the process in which an animal produces a sound and recognises characteristics of its surrounding - for instance, the location of surfaces, objects or prey - by listening to the echoes reflected by the environment. Studies on robot echolocation can be found in the literature. Such works adopt active sensors for emitting sounds, and the echoes reflected from the environment are thus analysed to build up a representation of the robot's surrounding. In this work, we address the usage of robot ego-noise for echolocation. By ego-noise, we mean the auditory noise (sound) that the robot *itself* is producing while moving due to the frictions in its gears and actuators. Ego-noise is a result not only of the morphological properties of the robot, but also of its interaction with the environment. We adopt a developmental approach in allowing a wheeled robot to learn how to *anticipate* characteristics of the environment *before* actually perceiving them. We programmed the robot to explore the environment in order to acquire the necessary sensorimotor information to learn the mapping between ego-noise, motor, and proximity data. Forward models trained with these data are used to anticipate proximity information and thus to classify whether a specific ego-noise is resulting from the robot being close to or distant from a wall. This experiment shows another promising application of predictive processes, that is for echolocation in mobile robots.

Introduction

Echolocation is the process in which an animal produces a sound and recognises characteristics of its surrounding - for instance, the location of surfaces, objects or prey - by listening to the echoes reflected by the environment. This capability is typical of animals such as bats, dolphins, and whales. Bats, for instance, emit sounds in the range of 10-150 kHz from the larynx and vocal tract, and they perceive the environment as acoustic images assembled from information received in echoes at the two ears (Simmons and Stein (1980)).

Similar to animal echolocation, blind and sighted people, after some training sessions, can become capable of echolocation by making sound emissions (Griffin (1944); Kolarik et al. (2014)). Many studies analyse the capability of blind individuals to produce sound emissions - for instance by tapping their canes, snapping their fingers or making clicking

noise with their mouths - and to use the returning echoes to get information about objects in their surroundings. Such a capability is known as human echolocation. Thaler et al. (2018) clearly demonstrated that blind and sighted people experienced in echolocation adjust their emissions - in terms of intensity and number of clicks - to situational demands, just like bats.

Echolocation has been studied in robotics as well. Steckel and Peremans (2013) presented a navigation model which solves a simultaneous localization and mapping (SLAM) task with a biomimetic sonar mounted on a mobile robot. In a more recent study, Eliakim et al. (2018) built a terrestrial robot that imitates bats' echolocation abilities. In particular, the proposed framework allowed the robot to navigate in a novel environment by means of a map created solely based on sound. Using the echoes reflected from the environment, the robot delineated the borders of the object it encountered in a greenhouse environment and classified them using an artificial neural network. The ANN was trained to distinguish between two object categories - plants and non-plants.

In this work, we address the usage of robot ego-noise for echolocation. By ego-noise, we mean the auditory noise (sound) that the robot *itself* is producing while moving due to the frictions in its gears and actuators. We have shown in previous work that such an ego-noise can be very informative in accomplishing specific tasks. In fact, robot ego-noise can carry out useful information about the movements that the robot is executing and about some of the characteristics of the external environment.

For instance, in Pico et al. (2016), we adopted forward models as computational tools for encoding the dynamics of the motor system of a wheeled robot and the effect of self-produced movements on the perceived ego-noise. In robotics and control theory, a forward model incorporates knowledge about sensory changes produced by self-generated actions, that is a mapping between actions and their consequences (Schillaci et al., 2016). In Pico et al. (2016), we showed how the auditory predictions provided by a set of trained forward models could be used for determining the velocity profile from its related auditory in-

put. Then, we showed how the auditory predictions provided by a forward model trained with data gathered using a self-exploration behaviour could be used to detect changes in the inclination of the surface on which the robot was moving.

In Pico et al. (2017), we extended the learning framework for the same robotic platform. In particular, we addressed the acquisition of the mapping between auditory ego-noise and the motor commands that generated it based on two behavioural and computational components: a self-exploration behaviour and an inverse model. We applied a convolutional autoencoder for a semi-supervised feature learning and dimensionality reduction of the auditory signals recorded from the robot microphones—whereas in the previous study, Mel-Frequency Cepstral Coefficients were used as features extracted from the auditory input. In the study, we demonstrated how the trained models could be used for imitating robot movements from listening to the noise they produced.

In Pico et al. (2018), we further extended the framework and demonstrated how predictive processes could be used to communicate motor information between robotic agents, through auditory means. In a simulated experiment, a robot generated a specific auditory feature vector from an intended sequence of actions and communicated it for reproduction to another robot, which consequently decoded it into motor commands, using the knowledge of its own motor system.

In the robot echolocation studies mentioned above, the robots emitted sounds using active sensors. In this work, instead, we use the ego-noise information produced by the robot movements - as in the studies reported in Pico et al. (2016, 2017, 2018) - in support of echolocation. In particular, we investigate the usage of forward models to anticipate proximity information and thus classify whether a specific ego-noise is resulting from the robot being close or distant from a wall. This experiment demonstrates another promising application of predictive processes, that is for echolocation in mobile robots.

Differently to the studies on echolocation mentioned above (Steckel and Peremans, 2013; Eliakim et al., 2018), we adopt a developmental approach in allowing the robot to learn how to *anticipate* characteristics of the environment *before* actually perceiving them. As it will be described in the following text, we programmed the robot to explore the environment in order to acquire the necessary sensorimotor information to learn the mapping between ego-noise, motor, and proximity data.

The rest of the paper is structured as follows. Section *Methodology* describes the robotic platform used in this study, the environmental setup, the data used for training the internal forward model and the model architecture. Section *Results* presents the results of the experiment and the quantitative analysis we carried out. Finally, we conclude the paper in section *Conclusions*.

Methodology

Robotic platform

In this experiment, we used a wheeled robot built at the Adaptive Systems Group at the Computer Science Department of the Humboldt-Universität zu Berlin (Figure 1). The robot is equipped with two DC gear motors placed in a differential configuration. Each motor has attached a quadrature magnetic encoder for speed measurement. We used the number of encoder counts made in a range of 100ms as a velocity measure. There are three distance sensors in front of the robot (only one is used in the experiment), as well as a microphone for audio recording. ROS (Robot Operating System) has been used for software development.

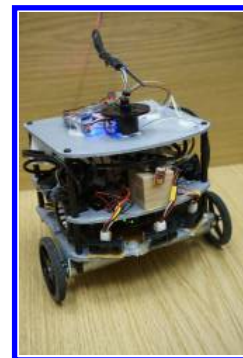


Figure 1: The two-wheeled robot with 2 speed sensors, 3 distance sensors and a microphone.

Environmental setup

The agent was moving in an environment made of a wooden floor and walls with dimensions of 92x56x38 cm (figure 2). The top part of the box was kept free to allow for free movement of the agent, since all power connections were supplied through a cable. Inside this arena, the robot could move up to 5 cm close to any wall, as is described in section (motor babbling).

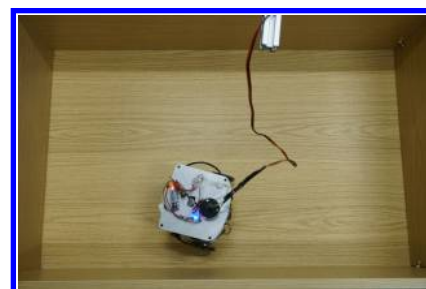


Figure 2: Top view of the arena used in the experiments.

Collected data

The following data was available for model building:

- *Auditory.* The Fast Fourier Transform (FFT) of 2048 samples of audio data at 22,050 kHz (about 100 milliseconds). The sound was recorded using the onboard microphone, which results in a vector of 1025 elements.
- *Motor speeds.* The mean of both quadrature magnetic encoders count in a range of 100 milliseconds.
- *Distance.* from infrared proximity sensors. Measurements equal or closer than 15 centimeters are labeled as “1”, longer distances are labeled as “0”
- *Motor commands.* Motor command applied to both motors at every time step in a range from 1 to 10.

All data was collected in time steps of 150 milliseconds interval, 100 milliseconds of data and a 50 milliseconds pause. Audio and motor speeds were recorded at the beginning of each time step. Distance measurements were taken after 100 milliseconds each time step (synchronized with the end of the audio and speed recordings).

Model Architecture

We adopted forward models (Wolpert et al. (2001)) as a computational tool for encoding the dynamics of the sensorimotor states of the robot. A forward model uses past and current sensorimotor states (in this work: sound, motor speed, and motor commands) to predict future sensory states (here, presence of a wall).

We designed two forward models for this experiment, implemented as deep neural networks: one taking the auditory signal recorded from the microphone as one of its inputs (model 2) and one without any auditory input (model 1). Model 1 served as a reference to assess if the robot ego-noise can have an effect in the wall detection task as we did not expect any relationship between its inputs (motor speed and motor command) and its output (distance to the wall). In other words, we assumed that the information contained in the motor speeds and motor commands should not be enough to detect the presence of a wall. On the other hand, we expected model 2 to encode useful information due to the presence of the generated robot ego-noise (and its interaction with the environment) as a model input.

The neural models were implemented with python 3.5 and keras 2.2.4, using tensorflow 1.11.0 as backend.

The forward model structures are defined as follows.

Model 1 Figure 3 illustrates the overall structure of this model. Its inputs are the motor command applied to the motors and the speed sensors’ measurements (mean of both motor speeds) of the last four time steps ($t - 3, t - 2, t - 1, t$), resulting in a 4×2 matrix. The output is a binary prediction at time $t + 1$ that indicates whether or not the robot is close to the wall (a threshold of 15cm has been utilised, to determine whether or not the robot is *close* to the wall).



Figure 3: Model 1 (no ego-noise) input-output structure

Figure 4 depicts in more detail the architecture. It consists of a long short-term memory (LSTM) recurrent layer at the beginning, followed by two dense layers (multi layer perceptrons). Dropout and batch normalization were applied at each layer. For the output layer we set a sigmoidal activation in the last layer.

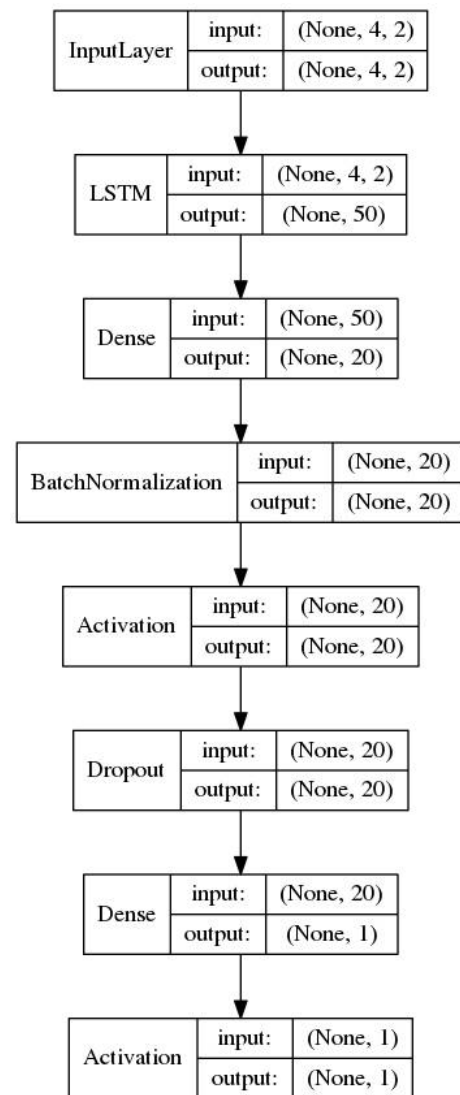


Figure 4: Model 1 (no ego-noise) neural network architecture

Model 2 This model, besides motor command and speeds, adds four time steps of ego-noise as input. Every time step consists of the Fast Fourier Transform (FFT) of a 100ms sound window (figure 5). Each window is a vector of dimension 1025.

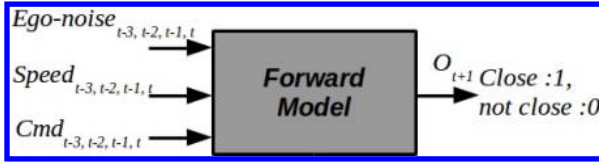


Figure 5: Model 1 (no ego-noise) input-output structure

Figure 6 shows the deep learning architecture, which consists of two different inputs: one that takes the ego-noise (4x1025 matrix) through three 1D convolutional layers and another one that takes the motor commands (4x2 matrix) and speeds through a dense layer. The outputs of both layers are then concatenated and fed to a LSTM layer, and then to 2 dense layers. The output of this model is the same as that of model 1, a binary prediction at time $t + 1$ indicating whether or not the robot is close to the wall. Dropout and batch normalization were applied at each layer, as well as a sigmoidal activation in the output layer.

Motor babbling

We collected the data from making the robot execute a random motor babbling behaviour inside the wooden arena. Ten possible motor commands were applied randomly to both motors every 4 time steps (every time step has a 150 milliseconds interval) so the robot moved in a straight line. At distances of about 5 centimeters from the wall, the robot activated an obstacle avoidance behaviour that made it turn and then reactivate the random behaviour. No data were collected during the obstacle avoidance behaviour.

Model training

We gathered 10,930 training samples and 2,800 validation samples, half of them when the robot was close to the wall and half when it was far from the wall. Both models were trained with the same data, ten times each, and tested with a new dataset of 4,172 samples. The models' loss function was binary cross entropy optimized with the Stochastic Gradient descent (SDG) algorithm.

Results

For each model we built ten repetitions to be able to assess the models' performance in the wall detection task. Due to the random initializations of the parameters, each of the repetitions performed slightly differently, as shown in figure 7.

Model 1 (no ego-noise input) had a mean accuracy of 57.15% on the validation dataset, while model 2 (ego-noise

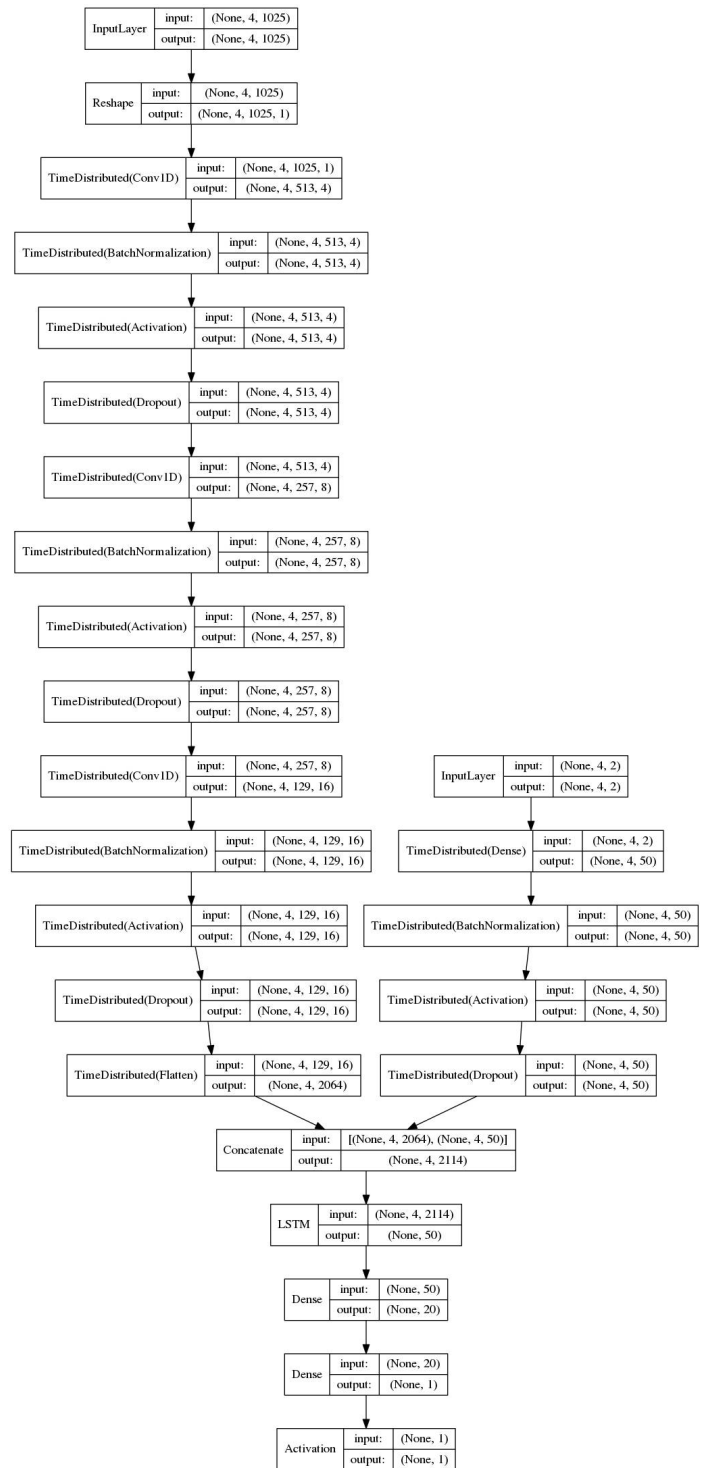


Figure 6: Model 2 (ego-noise) neural network architecture

input) had a mean accuracy of 82.39%. As it is shown, the model using the ego-noise as input had a better performance, which could indicate that there was a relationship between the generated robot ego-noise and its distance to the wall.

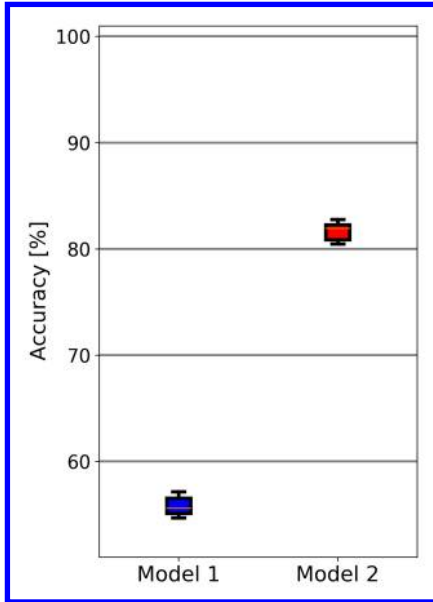


Figure 7: Accuracy stats of models in test set.

From these ten model candidates we selected the one with the best accuracy to carry out a classification task using the test dataset. Over the test dataset, model 1 had an accuracy of 58.05% and model 2 of 81.88%.

Figures 8 and 9 show the confusion matrices of this classification task. The rows indicate the targets (actual presence/absence of a wall) and the columns are the predictions made by the models. Therefore, a good model should have higher values over the matrix' main diagonal, as is the case with model 2 (including ego-noise as input). By contrast, model 1 was not able to make accurate predictions when the robot was close to a wall.

From these diagrams we see that the ego-noise information could be successfully incorporated into the models, thus improving the detection of the walls in the arena.

The confusion matrix in Fig. 9 grants an overview of the classification capabilities of the forward model 2, but it lacks specificity about its accuracy for each motor command issued. In order to explore this, we split the simulated outputs by motor command and show the results in Fig. 13. The motor commands in the x axis show increasing velocities from 1 to 10. It can be seen that the model's accuracy was very similar for almost all moving velocities, meaning that the agent was able to detect a wall using ego-noise information disregarding its displacement velocity. The exception was motor command 1, corresponding to the lowest moving velocity, in which case the model's accuracy was considerably

smaller.

The spectrogram in Fig. 11 also supports the idea that the slowest velocity was inadequate to train the forward model, as all other velocities show a frequency component not present in the slowest one (motor command 1). This can be seen for lower frequencies in figure 12 and for higher frequencies in figure 13. Since the energy in the frequency bins in the spectrogram was used as model input, this could explain the worse performance of the model at the lowest velocity. In other words, the ego-noise energy generated at the lowest speed seems to be insufficient. However, it remains unclear if another model trained only with slow velocities would be also able to map the relationship between motor command an expected sensory input. Further tests would be needed to address this problem and to establish whether the most significant information source is, in this case, only the motor velocity.

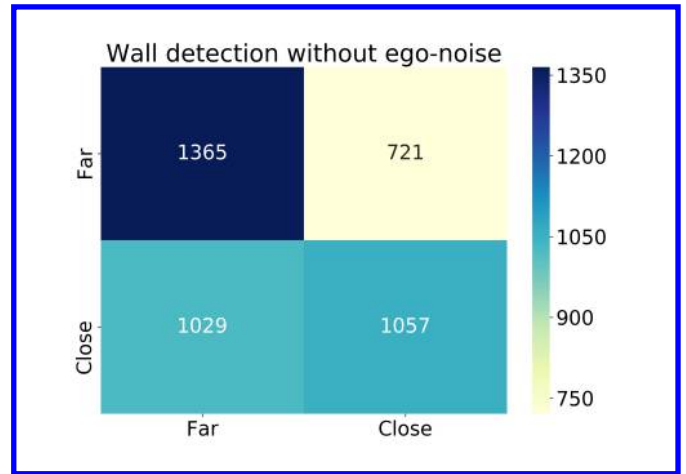


Figure 8: Model 1 test confusion matrix.

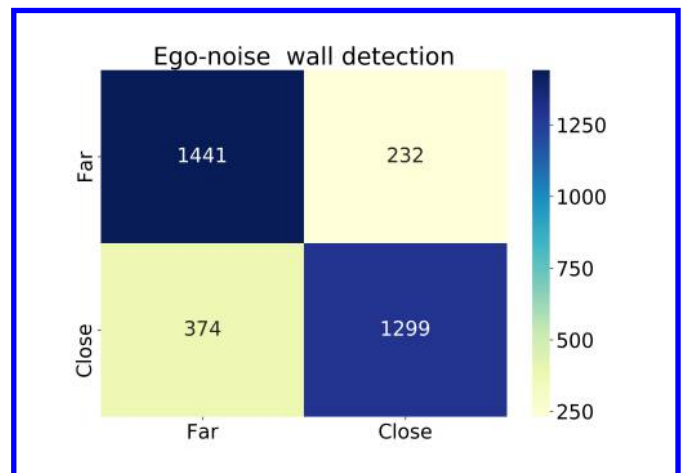


Figure 9: Model 2 test confusion matrix.

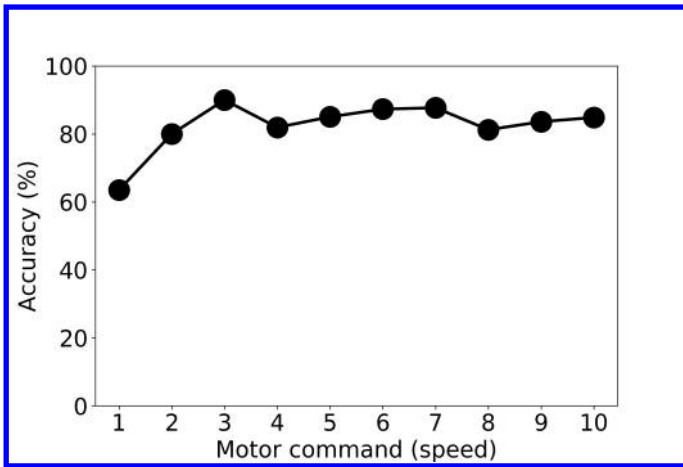


Figure 10: Accuracy stats of model 2 for each motor command in the test set.

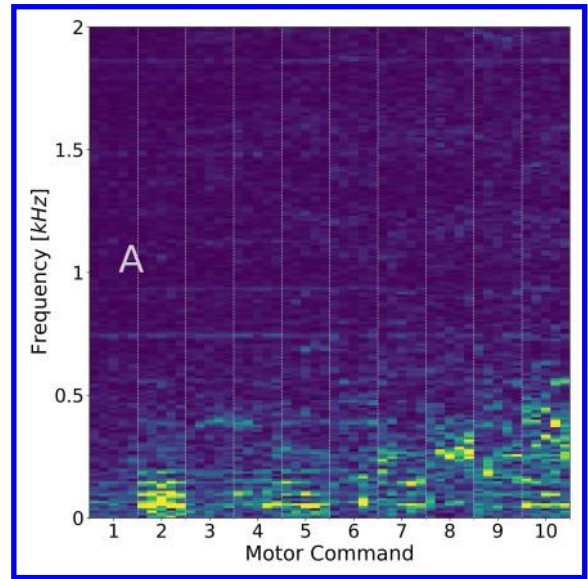


Figure 12: Spectrogram of ego-noise generated by each motor command at frequencies from 0 to 2 kHz.

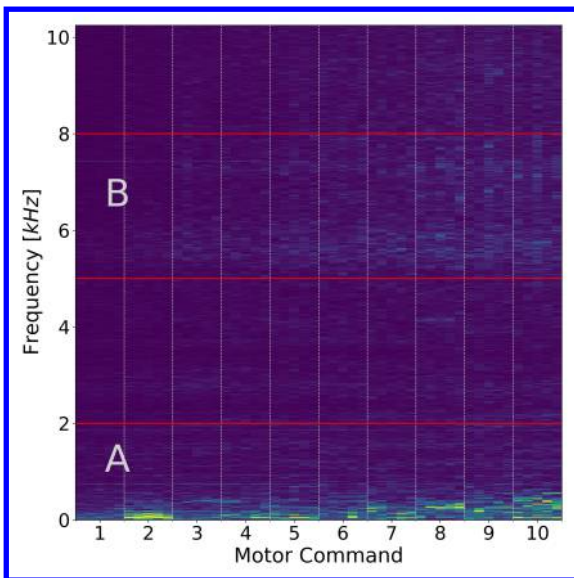


Figure 11: Spectrogram of ego-noise generated by each motor command at frequencies from 0 to 11 kHz.

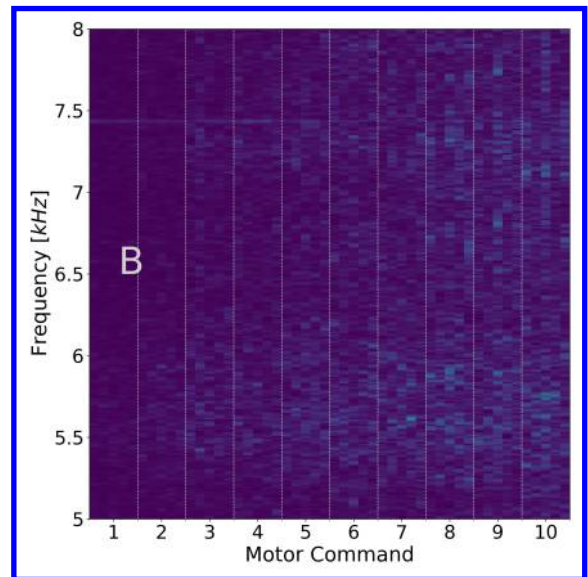


Figure 13: Spectrogram of ego-noise generated by each motor command at frequencies from 5 to 8 kHz.

Conclusions

This manuscript presented an experiment on ego-noise predictions for echolocation in a wheeled robot developed at the Humboldt-Universität zu Berlin. Echolocation is the process in which animals produce sounds and recognise characteristics of their surroundings. Typically, robotic studies addressing this utilise active sensors that emit sounds, and thus estimate the characteristics of the environment analysing the echoes that the surroundings reflect back to the robot sensors. In this study, we used the ego-noise produced by the movements of the robot itself as auditory information for echolocation. The experiment is preliminary, in the sense that only basic characteristics of the environment can be detected, namely whether or not the robot is close to an obstacle.

Moreover, we investigated the usage of predictive forward models for anticipating proximity information from motor and auditory ego-noise information, and for classifying whether a robot is near or far from walls in an experimental arena. In particular, the proposed framework - once having undergone a preliminary learning session - allows the robot to *anticipate* characteristics of the environment *before* actually perceiving them.

We adopted a developmental approach inspired by how infants acquire sensorimotor coordination, by means of self-exploration behaviours. During random motor exploration, the wheeled robot generated sensorimotor information consisting of motor information, speed detected using wheel encoders, proximity information and ego-noise recorded with the robot microphones. These data have been used to train a forward model, that encoded the mapping between the different modalities. The predictions of the forward models have been used to anticipate proximity information.

This experiment represents another promising application of predictive processes for the implementation of basic cognitive skills in artificial systems.

Robots actively emitting sounds through active sensors, as in bats echolocation, may surely outperform the proposed system. Nonetheless, the ego-noise that is in any case produced by the embodied interaction of the robot within its environment represents an additional source of information which is available *for free* to the system. The information carried out by the robot ego-noise may be still integrated with that of active sensors, and thus be exploited to increase the performance of the whole echolocation system.

The experiment presented in this work addresses only the learning of the presence or absence of a nearby wall in the proximity of the robot. Future work will address testing the algorithm in more complex environments.

Acknowledgment

This work has partially received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773875 (EU-H2020

ROMI, Robotics for Microfarms). The models proposed here will be adopted as a basis for sensorimotor learning experiments in the context of microfarming robots. The work of GS has partially received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 838861 (Predictive Robots).

References

- Eliakim, I., Cohen, Z., Kosa, G., and Yovel, Y. (2018). A fully autonomous terrestrial bat-like acoustic robot. *PLoS computational biology*, 14(9):e1006406.
- Griffin, D. R. (1944). Echolocation by blind men, bats and radar. *Science*, 100(2609):589–590.
- Kolarik, A. J., Cirstea, S., Pardhan, S., and Moore, B. C. (2014). A summary of research investigating echolocation abilities of blind and sighted humans. *Hearing research*, 310:60–68.
- Pico, A., Schillaci, G., and Hafner, V. V. (2018). Predictive models for robot ego-noise learning and imitation. In *2018 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 266–271.
- Pico, A., Schillaci, G., Hafner, V. V., and Lara, B. (2016). How do I sound like? Forward models for robot ego-noise prediction. In *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 246–251.
- Pico, A., Schillaci, G., Hafner, V. V., and Lara, B. (2017). On robots imitating movements through motor noise prediction. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 318–323.
- Schillaci, G., Hafner, V. V., and Lara, B. (2016). Exploration behaviors, body representations, and simulation processes for the development of cognition in artificial agents. *Frontiers in Robotics and AI*, 3:39.
- Simmons, J. A. and Stein, R. A. (1980). Acoustic imaging in bat sonar: echolocation signals and the evolution of echolocation. *Journal of Comparative Physiology*, 135(1):61–84.
- Steckel, J. and Peremans, H. (2013). Batslam: Simultaneous localization and mapping using biomimetic sonar. *PloS one*, 8(1):e54076.
- Thaler, L., De Vos, R., Kish, D., Antoniou, M., Baker, C., and Hornikx, M. (2018). Human echolocators adjust loudness and number of clicks for detection of reflectors at various azimuth angles. *Proceedings of the Royal Society B: Biological Sciences*, 285(1873):20172735.
- Wolpert, D. M., Ghahramani, Z., and Flanagan, J. R. (2001). Perspectives and problems in motor learning. *Trends in cognitive sciences*, 5(11):487–494.

Evolutionary Synthesis of Sensing Controllers for Voxel-based Soft Robots

Jacopo Talamini², Eric Medvet^{1,2}, Alberto Bartoli¹, and Andrea De Lorenzo¹

¹Machine Learning Lab, Department of Engineering and Architecture, Università di Trieste, Italy

²Evolutionary Robotics and Artificial Life Lab, Department of Engineering and Architecture, Università di Trieste, Italy
emedvet@units.it

Abstract

Soft robots allow for interesting morphological and behavioral designs because they exhibit more degrees of freedom than robots composed of rigid parts. In particular, *voxel-based soft robots (VSRs)*—aggregations of elastic cubic building blocks—have attracted the interest of Robotics and Artificial Life researchers. VSRs can be controlled by changing the volume of individual blocks: simple, yet effective controllers that do not exploit the feedback of the environment, have been automatically designed by means of Evolutionary Algorithms (EAs).

In this work we explore the possibility of evolving *sensing controllers* in the form of artificial neural networks: we hence allow the robot to sense the environment in which it moves. Although the search space for a sensing controller is larger than its non-sensing counterpart, we show that effective sensing controllers can be evolved which realize interesting locomotion behaviors. We also experimentally investigate the impact of the VSR morphology on the effectiveness of the search and verify that the sensing controllers are indeed able to exploit their sensing ability for better solving the locomotion task.

Introduction

Traditionally, robots have been made using rigid parts connected by joints. This allowed engineers to model robots behaviour and eased the design of body and controllers for the robots. On the other hand, creatures in nature are composed also, or mainly, of soft tissues and are quite effective in solving many complex tasks which are still utterly hard for robots (Kim et al., 2013). Inspired by nature (Lin et al., 2011), in the recent years many researchers focused on robots made on soft tissues, called *soft robots* (Rus and Tolley, 2015). The efforts concerned methods for the assisted or automated design of soft robot bodies (Cheney et al., 2013, 2014) and controllers (Braganza et al., 2007; Vaughan, 2018), often by means of simulation, and techniques for building actual soft robots (Iida and Laschi, 2011; Shepherd et al., 2011).

Voxel-based Soft Robots (VSRs) are a particular category of soft robots. They are aggregations of small elastic cubic building blocks called *voxels* (Hiller and Lipson, 2012).

VSRs have been important for the raise of the *embodied cognition paradigm* according to which the complexity of behavior of a (virtual) creature depends on both its brain and its body (Pfeifer and Bongard, 2006). According to this paradigm, a robot should be designed by considering brain and body together rather than by focusing only on its brain, i.e., on its controller. This research path has been particularly significant for VSRs, within a common framework in which the ability of the VSR to interact with the environment derived mainly from its body (Cheney et al., 2013, 2014).

In this paper, we explore the possibility of automatically synthesizing *sensing controllers* for simple VSRs, i.e., controllers which can sense the environment and exploit the gathered information for guiding the robot movements. We consider VSRs in which the sensing is distributed across the full body, i.e., on each voxel composing the VSR. In other words, we consider a VSR as an aggregation of simple parts that can be used both as actuators and as sensors.

We consider three different VSRs, i.e., with different bodies, and synthesize the corresponding controllers for solving a locomotion task. For each VSR we evolved a sensing controller and a more traditional, non-sensing controller. We represent sensing controllers as artificial neural networks (ANNs) whose topology is determined by the body of the robot, while for non-sensing controller we use a simpler representation which has already been successfully adopted (Kriegman et al., 2018). We synthesize both kinds of controllers with the same EA where, as we will show, the sensing controller corresponds to a larger search space than the non-sensing one, having more parameters. We evolved each VSR in two different environments, i.e. an even surface and an uneven surface.

Our experimental results, obtained by simulation, show that sensing controllers are always more effective than non-sensing ones, regardless of the body of the VSR and of the environment in which they evolved. Moreover, we also find that sensing controllers exhibit behaviors that are more heterogeneous than those of their non-sensing counterparts. Most importantly, we also assess the behavior of controllers in environments *different* from those in which they were

evolved and found that sensing controllers are more effective even in such scenarios. This result suggests that sensing controllers are indeed able to exploit their peculiar ability to sense the environment in which they are immersed.

Related work

The idea of evolving the body and the controller of simulated creatures dates back to '90s (Sims, 1994). In the cited work, the creatures body is modular, and the controller, in the form of an ANN, is distributed among their body components, capable of sensing the environment.

Other attempts to optimize ANNs controlling soft robots have been done later. For instance, Braganza et al. (2007) consider a tentacle-like manipulator which is controlled by an ANN, since the design of a traditional closed-loop controller for this specific robot was considered unfeasible. Another example is the optimization of a locomotion controller in the form of an ANN for a quadruped simulated creature (Vaughan, 2018).

On the other hand, control strategies different than ANNs have led to interesting results. Bruder et al. (2019), for example, have recently designed a linear dynamical model for controlling soft robots, based on a data-driven model: the authors claim that the proposed method, being more traditional and control-oriented, avoids issues of the ANNs acting as the black-boxes.

We remark that the works cited above face the problem of sensing, but are not based on VSRs. Research on VSRs focused more on how to design (often by means of evolutionary computation) the body of the robot: when the controller was of a non-trivial complexity, it had no sensing ability. Nevertheless, interesting behaviors have been found.

First attempts of morphological optimization of VSRs were done by Hiller and Lipson (2012) and, later, by Cheney et al. (2013). In the latter work, the novelty was mainly in the representation of the morphology and in the corresponding EA, both achieved with CPPN-NEAT (Neuroevolution of Augmented Topologies applied to Compositional Pattern-Producing Networks, Stanley (2007)): because of their ability to compactly describe patterns with repetitions and symmetries (which resemble nature), CPPN proved to be useful for evolving effective VSR morphologies. In that case, the task was locomotion and the controller was actually determined by the morphology, since different materials statically corresponded to different actuations. A similar approach has been applied later by Cheney et al. (2015) for evolving VSRs able to escape from a tight space.

A different kind of control of the VSR, but still not able to sense the environment, has been studied by Cheney et al. (2014). The authors proposed to define materials for the voxels in terms of their ability to propagate and react to an activation signal, inspired by properties of real, biological tissues. Morphologies were then evolved with CPPN-NEAT for the locomotion task.

Materials composing VSRs, in particular soft vs. stiff ones, are also the focus of (Bongard et al., 2016). The authors implemented a distributed growth mechanism, in place of actuation by oscillating global signals. The development of VSRs is allowed during their entire life span, acting at a lower time scale than the oscillation. The task is inspired by plants, and consists in growing towards static (possibly multiple) source of light in the environment, thus allowing the VSRs the ability to sense to a certain extent.

VSRs have been used as a case of study also for reasoning about the evolution in different environment (Corucci et al., 2018). The authors of the cited work evolved morphologies on a land environment in comparison with the ones in a water environment. Subsequently, they investigated the effects of an environmental transition, from land to water and the opposite, during the evolution, and they try to explain morphological results. To some degree, we too experiment with VSRs facing different environment: we assess their ability to move in environments which were not seen during the evolution and we show that sensing is beneficial in this scenario.

Scenario: controlling VSRs

Voxel-based soft robots (VSRs)

A *voxel-based soft robot* (VSR) is an assembly of one or more *voxels*, i.e., cubic building blocks, each linked to up to 6 neighbour voxels. Voxels are also elastic in the sense that their volume may either contract or expand with respect to the resting volume; the volume of each voxel may vary independently of the volume of any other voxel. We consider VSRs composed of a predefined number of voxels n . The *morphology* of a VSR is the way in which its voxels are linked.

We assume a discrete-time physics model in which scale values are set at regular intervals $t = k\Delta t$, $k \in \mathbb{N}$, where Δt is a parameter.

At any time, each voxel is defined by $s, \mathbf{x}, \mathbf{v}, \mathbf{v}'$, where: s is the *scale*, i.e., the ratio between the current and resting volume of the voxel; \mathbf{x} , \mathbf{v} , and \mathbf{v}' are the position, velocity, and acceleration of its center.

The behavior of the robot can be determined by imposing a value for the scale of each of its voxels. By varying the scale for each voxel over time, the corresponding positions, velocities and accelerations will vary over time as well depending on how voxels are linked together. In this work we use the physics model presented by Kriegman et al. (2017). The behavior of the VSR derives hence from the positions, velocities, and acceleration of its composing voxels, which themselves derive from the values imposed to the scale. We call *controller* of the VSR the way in which scale values are set over the time.

In general, a controller may set the values of the scale over the time basing on external input related to the interactions of the VSR with the environment; or, it may set the

scale regardless of those interactions. We call the two approaches *sensing* and *non-sensing* controllers, respectively. In the next sections we describe the two specific controllers that we consider in this work.

Non-sensing controller

We consider the simple non-sensing controller proposed by Kriegman et al. (2018) in which the scale s_i of the i -th voxel varies over time according to a sinusoidal signal, which determines the relative scale with respect to a resting value:

$$s_i(k) = s_i^0 + a \sin(2\pi f k \Delta t + \phi_i) \quad (1)$$

Frequency f and amplitude a are predefined and identical for all the voxels. Phase ϕ_i and resting value s_i^0 are instead defined separately for each voxel and constitute the parameters $\theta_{\text{NS}} = (s_1^0, \phi_1, \dots, s_n^0, \phi_n)$ of the controller. It can be seen, hence, that the number of parameters of this non-sensing controller, and therefore the size of the space of the corresponding controller instances, grows linearly with the number n of voxels in the VSR, i.e., $|\theta_{\text{NS}}| = 2n \sim O(n)$.

Sensing controller

We consider a sensing controller in which the VSR senses the environment in terms of the actual scale, velocity, and acceleration of each of its voxels: since these figures are determined also by how the VSR interacts with the environment, e.g., by pushing on the floor, they correspond to sensing the environment. These inputs, along with a single sinusoidal signal $\sin(2\pi f k \Delta t)$, are fed to a feed-forward ANN whose output layer determines the values of the scale to be set for each of the voxels.

More in detail, the ANN is composed of an input layer of $3n + 1$ neurons (the $+1$ being fed with the sinusoidal signal), an hidden layer of h neurons, and an output layer of n neurons. The activation function is the *Rectified Linear Unit* (ReLU). The input layer is fed with the values $s_1(k-1)$, $\|\mathbf{v}_1(k-1)\|$, $\|\mathbf{v}'_1(k-1)\|$ of each voxel. Each output neuron emits a value $o_i \in [-1, 1]$ which is then mapped to $[s^0 - \Delta s, s^0 + \Delta s]$, where s_0 and Δs are pre-defined values which are the same for all the voxels. The output of the i -th neuron at time $k\Delta t$ determines the scale $s_i(k)$ of the i -th voxel:

$$s_i(k) = s^0 + \Delta s o_i \quad (2)$$

$$o_i = \mathbf{f}_i(s_1(k-1), \|\mathbf{v}_1(k-1)\|, \|\mathbf{v}'_1(k-1)\|, \dots, s_n(k-1), \|\mathbf{v}_n(k-1)\|, \|\mathbf{v}'_n(k-1)\|; \theta_S) \quad (3)$$

where $\|\mathbf{v}_i(k-1)\|$ is the norm of the velocity of the i -th voxel at time $(k-1)\Delta t$, $\mathbf{f} : \mathbb{R}^{3n+1} \rightarrow [0, 1]^n$ represents the ANN, and θ_S are the ANN parameters (i.e., weights).

Concerning the number of neurons in the hidden layer, we set $h = 0.65n$. It can be seen that the number of parameters of this sensing controller grows with n^2 , i.e., $|\theta_S| = 3(n+1)h + hn \sim O(n^2)$.

Instantiating the controller

We instantiate the two controllers, i.e., we determine the values for their parameters θ_{NS} and θ_S , by means of evolutionary computation. To this end, we use for both controllers the Evolutionary Algorithm (EA) shown in Algorithm 1, already used by Kriegman et al. (2018) for evolving a non-sensing controller. This EA evolves a fixed size of n_{pop} individuals for n_{gen} generations, each individual being a vector θ of values ($\theta = \theta_{\text{NS}}$ and $\theta = \theta_S$ for the non-sensing and for the sensing controller, respectively). Only a unary genetic operator (mutation) is used: the mutation consists in perturbing each parameter in θ with probability p_{mut} , the amount of perturbation being with a random value randomly sampled from a normal distribution $N(0, \sigma_{\text{mut}})$. When evolving the non-sensing controller, we limit the values of each $s_i^0 \in \theta_{\text{NS}}$ parameter, after the mutation, to the interval $[s^0 - \Delta s, s^0 + \Delta s]$.

The generational model is a $n + m$ with overlapping and individuals are compared using Pareto dominance applied on their fitness and age: the age of the individual is incremented at each generation, whereas new individuals have the age set to 0. In case of tie in a selection (i.e., when one individual has to be selected from a set of individuals on the same Pareto front), individuals with the best fitness are preferred; in case of further tie, the individual is chosen at random. The same criterion is used to determine the *best individual* at the end of the evolution.

```

1  $P \leftarrow \emptyset$ 
2 foreach  $i \in \{1, \dots, n_{\text{pop}}\}$  do
3    $P \leftarrow P \cup (\text{random}(), 0)$ 
4 end
5 foreach  $i \in \{1, \dots, n_{\text{gen}}\}$  do
6    $P' \leftarrow \emptyset$ 
7   foreach  $(\theta, a) \in P$  do
8      $\theta' \leftarrow \text{mutate}(\theta)$ 
9      $P' \leftarrow P' \cup (\theta, a + 1)$ 
10     $P' \leftarrow P' \cup (\theta', a + 1)$ 
11   end
12    $P' \leftarrow P' \cup (\text{random}(), 0)$ 
13    $P \leftarrow \text{select}(P', n_{\text{pop}})$ 
14 end

```

Algorithm 1: The EA for evolving the controller.

The fitness of an individual θ , i.e., a controller for a VSR, measures its ability to perform a given task. In this work, we consider the locomotion task and set the fitness to the distance that the VSR corresponding to the individual travels along the x -axis during a simulation of a predefined amount of n_{sim} time steps. Despite its apparent simplicity, locomotion is considered a benchmark for VSRs (Cheney et al., 2013, 2014; Kriegman et al., 2018).

We remark that other techniques might be used for the purpose of instantiating a controller, given a morphology

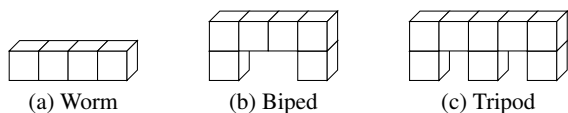


Figure 1: The three different VSR morphologies.

and a simulator. In particular, for learning the sensing-controller, which is based on ANN, EAs operating on ANNs might be used, e.g., NEAT (Stanley and Miikkulainen, 2002) or CPPN-NEAT (Stanley, 2007). Or, since the considered scenario consists in an autonomous agent that interacts with the environment trying to maximizing a reward (here, the traveled distance), Reinforcement Learning techniques might be used (Duan et al., 2016). However, we leave the exploration of these alternative options to future work, since here we are interested in comparing the nature of the controller, and the information it can exploit, rather than the learning technique.

Experiments and results

We performed an experimental evaluation aimed at investigating the effectiveness of the sensing controller with respect to the non-sensing one. In particular, we aimed at answering the following research questions: (RQ1) Is a sensing controller better than a non-sensing one? (RQ2) Does the larger size of the search space for the sensing controller affect the search effectiveness? (RQ3) Is a sensing controller actually able to exploit its ability to sense the environment? For answering these questions, we considered three different VSR *morphologies* and two different *environments*.

Morphologies are shown in Figure 1: we call the corresponding VSRs *worm*, *biped*, and *tripod*. They differ in the number of composing voxels ($n \in \{4, 6, 8\}$) and hence correspond to different numbers of parameters for defining the controllers.

Concerning the environment, we simulated the movement of the VSR on an *even* surface and on an *uneven* surface. In all cases, we performed 30 evolutionary runs (i.e., 30 independent executions of Algorithm 1) for each combination of morphology and environment. We used the implementation made available by Kriegman et al. (2018)¹, with the parameters of the physics model, morphologies, and EA shown in Table 1. We run the experiments using AWS EC2 on the c4.8xlarge EC2 instances, each equipped with 36 vCPU based on 2.9 GHz Intel Xeon E5-2666 and with 60 GB RAM; we distributed the fitness evaluation across the vCPUs and runs across instances.

In each run, the VSR was put in the environment with its main dimension laying on the x -axis, the same axis along

¹<https://github.com/skriegman/how-devo-can-guide-evo>

Param.	Value	Param.	Value
Δt	0.14 ms	n_{pop}	30
a	0.001 m ³	n_{gen}	200
f	40 Hz	p_{mut}	$\frac{1}{ \theta }$
s^0	0.01 m ³	σ_{mut}	1
Δs	0.001 m ³	n_{sim}	10 000

Table 1: Parameters of the physics model (top left), morphologies (bottom left), and EA (right) used in the experiments.

Table 2: Fitness (in mm, mean μ and standard deviation σ across the 30 runs) of the best individual at the end of the evolution in the environment with even surface. The p -value is computed with the Mann-Whitney U-test (see text).

Morph.	Non-sensing		Sensing		p -value [$\times 10^{-3}$]
	μ	σ	μ	σ	
Worm	146	8	3012	329	0.002
Biped	69	19	931	74	0.006
Tripod	550	26	636	76	0.024

which the traveled distance is measured for computing the fitness.

Environment: even surface

Table 2 presents the main results obtained in the environment with even surface, with the three morphologies. The table shows the mean μ and the standard deviation σ of the fitness of the best individual at the last generation across the 30 runs. The table also shows the p -values obtained with the Mann-Whitney U-test that we performed for each morphology in order to verify if the samples have the same median.

The foremost finding is that sensing controllers clearly outperform non-sensing ones. That is, a VSR controlled by a sensing controller is in general better in performing the locomotion task, regardless of the morphology. The difference is always statistically significant (with a significance level of $\alpha = 0.05$) and substantial in two on three cases, the worm and the biped.

Concerning the tripod, the sensing controller is still better, in terms of the final best fitness, than the non-sensing one, but the difference is lower (636 ± 76 vs. 550 ± 263) with respect to the worm and biped (for which traveled distance difference is of an order of magnitude). We interpret this finding as a consequence of the fact that the number of voxels in the tripod is larger: the complexity of the controller is $O(n)$ for the non-sensing case and $O(n^2)$ for the sensing case, and the same applies for the size of the search space. As a further evidence for this interpretation, we show in Figure 2 how the fitness of the best individual varies during the evolution (mean across the 30 runs) for the

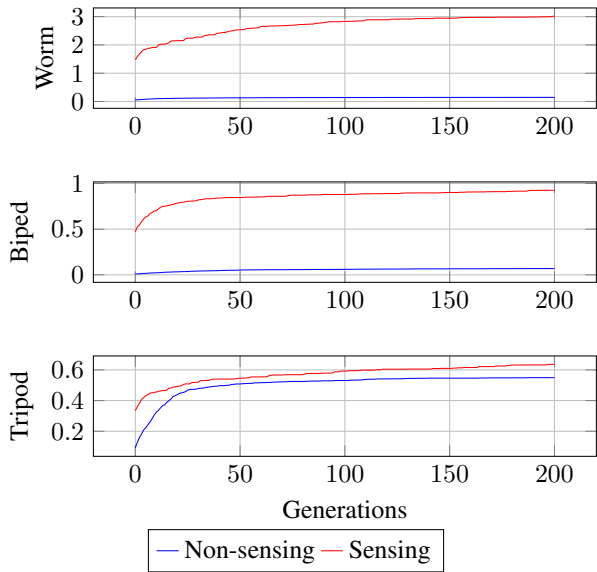


Figure 2: Fitness (in m, mean across the 30 runs) of the best individual during the evolution.

three morphologies. Beyond highlighting the lower difference for the tripod, Figure 2 suggests that the evolution of a sensing controller has not yet stopped at the end of the evolution (200-th generation), for this case; on the other end, this does not occur with the non-sensing controller. In other words, the larger search space makes finding the optimum harder. We remark, however, that other techniques exist for evolving ANNs which are suitable for scenarios like the one considered in this work. In particular, we argue that NEAT (or its recent variants as, e.g., the one of Silva et al. (2015)) might be a way to address the issue of the large search space, thanks to its ability to progressively increase the expressiveness of the representation—i.e., complexification.

Analysis of the behaviors In order to further investigate the differences between the sensing and non-sensing controllers, we observed the resulting behaviors during the simulations: i.e., we looked at the way best evolved controllers moved and drawn qualitative reasoning (see Figure 3). We found that sensing controllers resulted, in general, in a broader set of behaviors, the difference being more apparent for the worm. Interestingly, for this morphology the behaviors exhibited by the sensing controllers often visually resembled those of the real biological counterpart.

In an attempt of quantifying the result of this qualitative analysis, we devised a way of systematically capturing and describing the behaviors of the VSR—similar procedures have been already used for analyzing the behavior of robots with evolved controllers, e.g., in Silva et al. (2017). We proceeded as follows. (1) For each morphology, we considered all and only the 60 best controllers (sensing and non-sens-

ing) obtained at the last generation. (2) We considered the discrete signals corresponding to the position $\mathbf{x}_{\text{CM}}(k)$ of the center of mass of the VSR during fitness evaluation. Figure 4 shows an example trajectory of one of the best sensing controllers for the worm morphology. (3) We computed the discrete Fourier transform (DFT) coefficients \mathbf{d}_x and \mathbf{d}_y , with $\mathbf{d}_x, \mathbf{d}_y \in \mathbb{R}^{n_{\text{sim}}}$, of the x - and z -components of $\mathbf{x}_{\text{CM}}(k)$; we did not consider the y -component since VSRs do not move significantly along that axis (see Figure 4). (4) We concatenated \mathbf{d}_x and \mathbf{d}_y , hence obtaining a vector $\mathbf{d} \in \mathbb{R}^{2n_{\text{sim}}}$ for each observed behavior. (5) Finally, we mapped all the behaviors from $\mathbb{R}^{2n_{\text{sim}}}$ to \mathbb{R}^2 using Multidimensional Scaling (MDS) (Cox and Cox, 2000). We explored different dimensionality reduction techniques, e.g., t-SNE (Maaten and Hinton, 2008): the qualitative observations presented below did not change.

Figure 5 shows the results of the analysis of the behaviors: for each morphology, the figure includes a plot where each behavior is a marker positioned according to the first two MDS coordinates. Three observations can be done based Figure 5. First, for the two simplest morphologies (worm and biped) the behaviors obtained with sensing and non-sensing controllers look clearly dissimilar: the red cloud is far from the blue cloud. Second, non-sensing controllers result in more homogeneous behaviors than sensing controllers: the red cloud is in general larger than the blue cloud. Third, the tripod case is, consistently with the previous findings, different from the other two cases: the difference of behaviors is fuzzier and similar behaviors can be found which are obtained with different controllers. We think that the motivation for this finding is twofold. On one hand, the larger complexity of the morphology may result in a larger set of interactions between the VSR and the environment, that is, in a larger expressiveness. On the other hand, as already observed above, the larger search space of this case may take longer to converge to a good controller; i.e., from another point of view, within the same number of generations, different evolutionary runs may follow different paths in the search space which do not end in the same “point”.

Environment: uneven surface

For the purpose of answering (RQ3), we considered a second case in which some aspect of the environment changes over the time. Differently than in the environment with even surface, variable environmental conditions constitute an opportunity for the sensing controller to exploit its peculiar ability of sensing the environment: that ability is instead not available for VSRs with the non-sensing controller.

For easing the experimentation, we introduced the variable conditions as a varying vector for the gravity acceleration. In particular, we varied the direction of the gravity vector during the simulation and kept constant its norm $\|\mathbf{g}\| = 9.8 \text{ m s}^{-2}$. The condition can be expressed as a function describing the value of the x -component $g_x(k)$ of

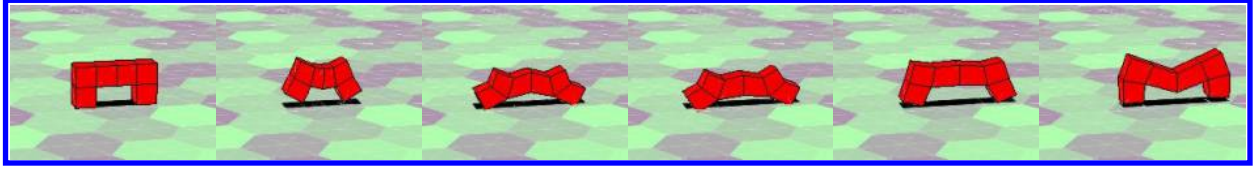


Figure 3: Frames capturing the behavior of a biped with one of the evolved sensing controller in the environment with even surface.

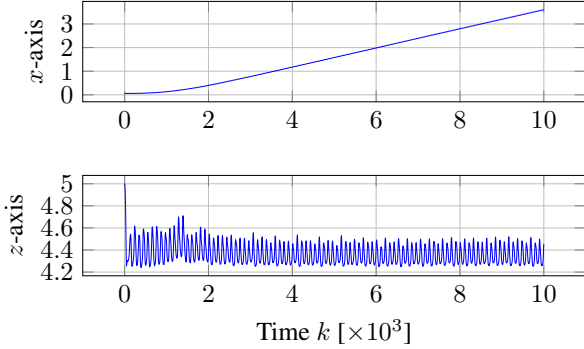


Figure 4: Trajectory $\mathbf{x}_{\text{CM}}(k)$ of the center of mass, shown separately for the two salient axes (scale is $\times 10^{-3}$ for the z -axis), of a worm with one of the evolved sensing controller in the environment with even surface.

the gravity vector \mathbf{g} over the time—assuming that the y -component is always equal to 0.

We proceeded as follows. First, we performed the evolutionary runs imposing a sinusoidal signal for the x -component of the gravity:

$$g_x^{\text{evo}} = \sin(2\pi f_{\mathbf{g}}^{\text{evo}} k \Delta t) \quad (4)$$

where $f_{\mathbf{g}}^{\text{evo}} = 2 \frac{1}{\Delta t n_{\text{sim}}} = 1.43$ Hz. Then, we assessed each evolved controller (i.e., the best individual at the last evolution) in three different *validation scenarios*:

$$g_x^{\text{flat}}(k) = 0 \quad (5)$$

$$g_x^{\text{step}}(k) = \begin{cases} 0 & \text{if } k \leq \frac{n_{\text{sim}}}{2} \\ 3 & \text{otherwise} \end{cases} \quad (6)$$

$$g_x^{\text{sin}}(k) = \sin(5\pi f_{\mathbf{g}}^{\text{evo}} k \Delta t) \quad (7)$$

By considering validation scenarios which are different from the one using during the evolution, we hence also assessed the generalization ability of the EA in evolving the VSR controllers. Note that varying the direction of the gravity vector basically corresponds to considering an uneven, instead of flat, surface on which the VSR moves.

Table 3 shows the results of the experiments in the uneven environment.

It can be seen that, also in this environment, the sensing controller is always more effective than the non-sensing one.

Table 3: Fitness of the best individual at the end of the evolution and its traveled distance in the validation scenarios (both in mm, mean μ and standard deviation σ across the 30 runs) in the uneven environment. ρ is the ratio between the traveled distance in the validation scenario and the fitness value.

		Non-sensing			Sensing		
Morph.		μ	σ	ρ	μ	σ	ρ
Fitness	Worm	120	8		3050	358	
	Biped	78	19		873	180	
	Tripod	556	60		620	378	
Flat	Worm	138	5	1.15	3132	498	1.02
	Biped	73	30	0.93	2228	187	2.55
	Tripod	528	43	0.95	727	100	1.17
Step	Worm	111	56	0.92	3543	412	1.16
	Biped	69	20	0.88	1010	226	1.15
	Tripod	539	160	0.96	870	213	1.40
Sin	Worm	104	7	0.86	3194	281	1.04
	Biped	77	17	0.99	446	151	0.51
	Tripod	505	48	0.91	512	175	0.82

VSRs moved by the former travel a longer distance in any condition: both when computing the fitness (i.e., with g_x^{evo}) and in the validation scenarios (i.e., with g_x^{flat} , g_x^{step} , and g_x^{sin}). As for the even environment, differences are in general less apparent for the tripod than for the other two morphologies. All the differences are statistically significant according to the Mann-Whitney U-test ($\alpha = 0.05$): we do not show the values in the table.

Of more interest are the findings concerning the comparison between the fitness of the best individual and its performance in the validation scenario. Table 3 captures the outcome of this comparison in the two ρ columns: for a given morphology, controller, and validation scenario, ρ is the ratio of the distance traveled in the validation scenario and the fitness value, i.e., the one traveled with g_x^{evo} .

The key finding is that ρ is lower than 1 in most cases (8 on 9) for the non-sensing controller and greater than 1 in most cases for the sensing controller (7 on 9). VSRs equipped with the sensing controllers are hence able to move well on scenarios different than the one used for their evolu-

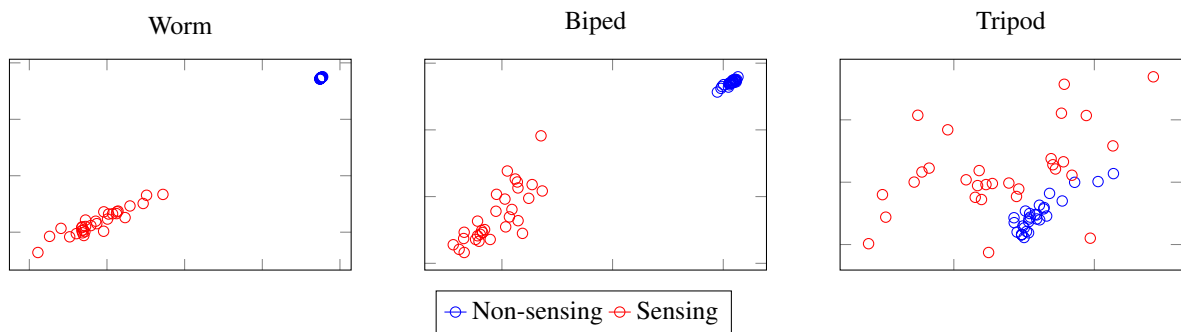


Figure 5: Behaviors resulting from the 60 best controllers evolved in the environment with even surface, with the three morphologies.

tion, whereas VSRs with non-sensing controller are not. We explain this clear difference with the fact that the sensing ability allows to react to an environment different from the one the controlled evolved and to adapt the VSR behavior.

Finally, Table 3 shows that, not surprisingly, the Sin validation scenario is the most difficult for all the VSRs: still, the worm equipped with a sensing controller is able to perform not worse on this scenario than on the one seen during the evolution ($\rho = 1.04$).

Analysis of the behaviors We performed the same analysis of the behaviors as for the environment with the even surface. The results are shown in Figure 6.

The findings are similar to the previous case. Sensing controllers exhibit, in general, more various behaviors and this difference is less apparent for the tripod than for the worm and the biped. However, Figure 6 also highlights that the behaviors resulting from sensing controllers differ among the three validation scenarios. The difference is more apparent for the biped. We motivate this latter finding with the fact that this morphology is a good trade-off in complexity: it is not too simple to prevent large variation in the behaviors (like the worm), nor too complex to make harder the evolution of controller able to exhibit a well-defined behavior (like the tripod).

Conclusions

Voxel-based soft robots are a promising framework in which the behavior of a robot is determined by both its brain, i.e., its controller, and its body. In this work we have explored a form of holistic design in which the controller is equipped with sensing capabilities distributed across the full body of the robot. We have considered a sensing controller represented as a neural network and have considered the problem of synthesizing such a controller automatically, by means of an Evolutionary Algorithm. We have exercised such an algorithm on three different bodies, each in two different environments, with the aim of solving a locomotion task. We have compared the resulting sensing controller to a more tra-

ditional one, also synthesized automatically with the same Evolutionary Algorithm, and we have found that the sensing controller is more effective than its non-sensing counterpart, also when immersed in an environment different from the one in which it evolved.

We believe these results are very promising and suggest that the shifting of complexity from the controller to the body intrinsic to voxel-based soft robots, should be carefully coupled with forms of distributed sensing. We intend to investigate the potential of sensing controllers on larger robots and more complex tasks. In order to cope with the resulting complexity of the search space, we plan to rely on a more efficient evolutionary framework, such as, e.g., CPPN-NEAT, as well as a modular design in which robots are assembled out of smaller (parts of) robots evolved separately.

Acknowledgments

The experimental evaluation of this work has been done on Amazon AWS within the “AWS Cloud Credits for Research” program. We thank Vittorio Casagrande and Erica Salvato for useful discussions about the design of the sensing controller.

References

- Bongard, J., Laschi, C., Lipson, H., Cheney, N., and Corucci, F. (2016). Material properties affect evolutions ability to exploit morphological computation in growing soft-bodied creatures. In *Artificial Life Conference Proceedings 13*, pages 234–241. MIT Press.
- Braganza, D., Dawson, D. M., Walker, I. D., and Nath, N. (2007). A neural network controller for continuum robots. *IEEE transactions on robotics*, 23(6):1270–1277.
- Bruder, D., Gillespie, B., Remy, C. D., and Vasudevan, R. (2019). Modeling and control of soft robots using the koopman operator and model predictive control. *arXiv preprint arXiv:1902.02827*.
- Cheney, N., Bongard, J., and Lipson, H. (2015). Evolving soft robots in tight spaces. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pages 935–942. ACM.

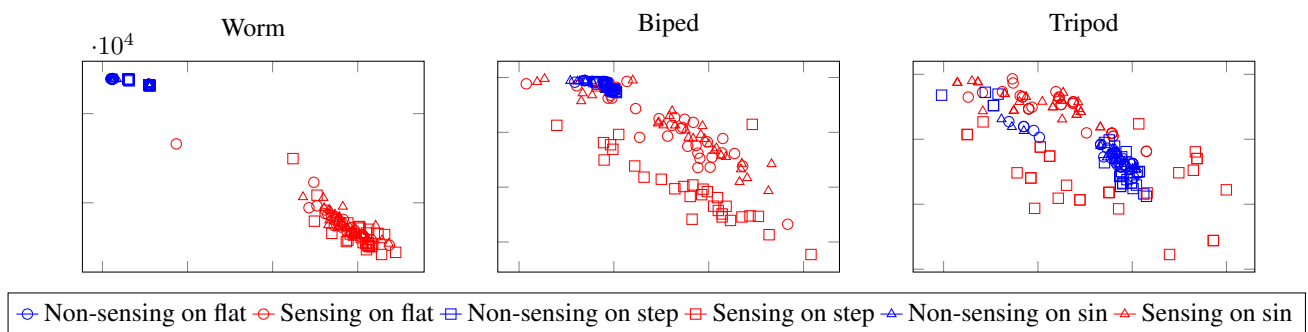


Figure 6: Behaviors resulting from the 60 best controllers evolved in the environment with uneven surface, with the three morphologies when testing them in the three validation scenarios.

- Cheney, N., Clune, J., and Lipson, H. (2014). Evolved electrophysiological soft robots. In *Artificial Life Conference Proceedings 14*, pages 222–229. MIT Press.
- Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2013). Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 167–174. ACM.
- Corucci, F., Cheney, N., Giorgio-Serchi, F., Bongard, J., and Laschi, C. (2018). Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics*, 5(4):475–495.
- Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.
- Duan, Y., Chen, X., Houthoofd, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338.
- Hiller, J. and Lipson, H. (2012). Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466.
- Iida, F. and Laschi, C. (2011). Soft robotics: challenges and perspectives. *Procedia Computer Science*, 7:99–102.
- Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294.
- Kriegman, S., Cappelle, C., Corucci, F., Bernatskiy, A., Cheney, N., and Bongard, J. C. (2017). Simulating the evolution of soft and rigid-body robots. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1117–1120. ACM.
- Kriegman, S., Cheney, N., and Bongard, J. (2018). How morphological development can guide evolution. *Scientific reports*, 8(1):13934.
- Lin, H.-T., Leisk, G. G., and Trimmer, B. (2011). Goqbot: a caterpillar-inspired soft-bodied rolling robot. *Bioinspiration & biomimetics*, 6(2):026007.
- Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Pfeifer, R. and Bongard, J. (2006). *How the body shapes the way we think: a new view of intelligence*. MIT press.
- Rus, D. and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature*, 521(7553):467.
- Shepherd, R. F., Ilievski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., Chen, X., Wang, M., and Whitesides, G. M. (2011). Multigait soft robot. *Proceedings of the national academy of sciences*, 108(51):20400–20403.
- Silva, F., Correia, L., and Christensen, A. L. (2017). Hyper-learning algorithms for online evolution of robot controllers. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 12(3):14.
- Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2015). odneat: An algorithm for decentralised online evolution of robotic controllers. *Evolutionary Computation*, 23(3):421–449.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.
- Vaughan, N. (2018). Evolution of neural networks for physically simulated evolved virtual quadruped creatures. In *Conference on Biomimetic and Biohybrid Systems*, pages 507–516. Springer.

Different Forms of Random Motor Activity Scaffold the Formation of Different Habits in a Simulated Robot

Mario Zarco¹ and Matthew Egbert^{1,2}

¹Department of Computer Science, University of Auckland, Auckland, NZ

²Te Ao Mārama — Centre for Fundamental Inquiry, University of Auckland, Auckland, NZ

zarco.mario.al@gmail.com

Abstract

A habit is formed through the repeated enactment of sensorimotor regularities created and maintained by means of plastic changes on the mechanism that brings them about. This precarious, self-maintaining sensorimotor organization is known as sensorimotor autonomy. One can imagine how some habits would be better suited to the maintenance of a biological individual. Evolution can bias the parameters of the plastic medium over which sensorimotor autonomy emerge so as to be beneficial to biological autonomy. In this work, we show that varying some parameters that bring about plastic changes in the behavior-generating medium, different sensorimotor individuals emerge. The simulation consists of a simple robot coupled with a habit-based controller with a random-based exploratory phase in a one-dimensional environment. The results show that, varying the parameters of such a phase, qualitative different habits emerge characterized by static, monotonic and oscillatory behaviors. Quantitative variations of the oscillatory behavior are also shown using the frequencies distribution obtained from the motor time series of the formed habits. The results are interpreted in terms of how the sensorimotor habitat could emerge from the random traversing of the sensorimotor environment. Finally, a comparison between this model and the skin brain thesis is presented.

Introduction

The enactive concept of autonomy, defined by concepts of precarious self-maintenance and operational closure, can be used in diverse contexts to identify an “individual”. This concept of autonomy first emerged as a way to identify and delimit biological individuals (autopoiesis) (Maturana and Varela, 1980). It has since been applied in a variety of other contexts, including sensorimotor dynamics, the organs and organization of multicellular life, etc. By applying this method in these different contexts and at different scales, diverse types of precarious individuality emerges. Of particular relevance to the remainder of this paper, is the relationship between an agent’s biological and sensorimotor autonomies, i.e. the relationship between its biological self-maintenance (persistence as an organism) and self-maintaining sensorimotor world interactions. The notion of autonomous sensorimotor dynamics is not one that is widespread, nor is it self-evident, so let us spend a little

bit of time expanding upon what is meant by this concept. It is clear that sensory input influence subsequent actions. The converse is also true: actions influence sensory input. You step forward and the image projected on to your retina changes in rule-like ways, where for instance objects that are closer to you will move more than those that are farther away. It is possible to construct artificial “brains” i.e. media that transform sensory input into motor action, such that (and this is the critical bit) one or more patterns of action maintain the conditions necessary for those patterns of action to persist. In this context, we can identify precarious autonomous sensorimotor individuals¹ — patterns of action that stabilize or otherwise maintain the conditions that are necessary for the mechanism that generates those patterns of action to persist.

A number of investigations have modeled this idea of sensorimotor autonomy (even if they have not always used this word to label it). Studies include (Di Paolo, 2003; Egbert and Barandiaran, 2014; Egbert and Canamero, 2014), and it has been related to Ashby’s notions of ultrastability (Ashby, 1952) and the apparent ultrastability of human perception, exemplified by our ability to adapt to diverse systematic perturbations to our sensory apparatus (e.g. Kohler, inversion of the visual field (Di Paolo, 2003)). The whole idea is the central topic of the recent book by Di Paolo et al. (2017).

The relationship between sensorimotor autonomous individuals and the biological autonomous individual is not one of simple subsumption or hierarchy (Barandiaran, 2008). An organism’s nervous system is part of the medium in which sensorimotor autonomous individuals emerge and persist (or not!). Sensorimotor individuals are thus fundamentally dependent upon the organisms biological body. It is also the case that (most) biological bodies essentially depend upon the nervous-system-mediated sensorimotor activities to persist. Each depends upon the other.

It is interesting, though, that these relationships neither require nor imply that sensorimotor individuals and the biological individual to have compatible norms or viability limits. Nevertheless many habits are beneficial or neutral

¹Other authors might refer to a “sensorimotor autonomy”

with regard to impact upon biological viability (Egbert and Canamero, 2014).. Why? How does this alignment of norms and viability conditions come to exist? How do the sensorimotor individual and the biological individual over which they operate come to cooperate? And just as interestingly: in what conditions does this cooperation fail to emerge/develop?

The properties of the nervous system play a key role in influencing which types of sensorimotor autonomous individuals will persist. One possible mechanism through which the biological and sensorimotor norms become aligned is via evolution. Evolution tunes the nervous system so as to produce behaviors that are beneficial to biological viability. This could happen in a variety of different ways. For instance, we can compare among three different possibilities. The first one is the evolution of a set of hard-wired stimulus-response reflexes. If the organism's behavior consisted of nothing but hard-wired responses, then the mechanism that produces the behavior would not depend upon the result of the behavior. That is, the hard-wired reflexes would not participate in maintaining the conditions necessary for such a mechanism to generate them. Therefore, the behavior could not be considered as a sensorimotor individual because these reflexes (and consequently the behaviour) would not cease to operate, unless the organism died. A hard-wired mechanism thus would in fact still depend upon the behavior in terms of biological viability. This behavior would then be part of the precarious biological autonomous individual, not a sensorimotor one.

Another possibility is one where reflexes play a less direct role of scaffolding the formation of sensorimotor individuals. The idea here is that evolution could result in a few hard-wired stimulus responses that influence which patterns of behavior are more likely to occur. Plastic mechanisms in the nervous system then support whichever self-sustaining sensorimotor individual emerge, but the reflexes bias the system to produce sensorimotor individuals that are conducive to biological survival.

No doubt there are hard-wired reflexes that guide some our behavior (and our sensorimotor development). It seems unlikely, though, that more than a few specific stimulus responses could be specified at the genetic level. The last possibility is one where genes can modulate more coarse features of nervous system topology, connectivity and interaction, and random motor babbling enables and biases the formation of different sensorimotor individuals. Plastic changes in the nervous system then support the self-maintaining sensorimotor individual that bring about such changes. This is the idea that we explore using a model in the next section. Specifically, we simulate a simple robot embedded in a 1D world. The robots "controller" is a sensorimotor medium which reinforces previous patterns of sensorimotor behavior. We vary the parameters of the random motor activity that is activated by the controller when it is

in sensorimotor conditions it has not previously experienced and show that just by modulating this basic parameter, it is possible to bias the system to produce particular qualitative behaviors (static, monotonic, oscillatory) and even to vary some of the quantitative aspects of those behaviors (oscillation frequency).

The concept of habit and its models are described in the next section. After that, the model and the experiments are explained. Finally, the results are reported and discussed.

The Concept of Habit

The concept of habit was one of the most important theoretical primitives for the study of mind before being usurped by the notion of mental representation (Barandiaran and Di Paolo, 2014). Currently, with the anti-representationalism trend in cognitive science due to the different embodied approaches to cognition, a reappraisal of the notion of habit has been required. Moreover, habits could be the building blocks for modeling the organization of behavior, and their ecology the core of *Mental Life* (Barandiaran, 2007). A very brief look to the re-definition of habit that inspired this work based is presented so as to compare modeling frameworks below. Di Paolo (2003) argued that habits are dynamic invariants obtained from a circular process between plasticity and behaviour. Habits can be understood as self-sustaining dynamic structures that are challenged when the behaviour is perturbed and adaptation is needed. After that, Barandiaran (2008) detailed the definition of habit as "a self-sustaining pattern of sensorimotor coordination that is formed when the stability of a particular mode of sensorimotor engagement is dynamically coupled with the stability of the mechanism generating it" (p. 281). Finally, Egbert and Barandiaran (2014) added the property of reinforcement by repetition to this concept of habit in order to have a working definition. A habit is then precarious in the sense that, if the sensorimotor patterns of behavior are not continuously re-enacted, the consequently plastic change is produced differently affecting the creation and maintenance of the sensorimotor correlations it contributes to cause.

Simulation models has been used as tools to obtain new insights through the exploration of new theoretical positions and, consequently, a possible conceptual reorganization (Di Paolo et al., 2000). On the one hand, Di Paolo (2003) explored an homeostatic neural controller using an evolutionary robotics approach to propose that true intentionality can be brought about investing a robot with a mechanism for acquiring a way of life: habits. In this case, a habit is grounded on the circular interaction between sensorimotor correlations and neurodynamic patterns, and the plastic reinforcing changes on the habit formation mechanism (Barandiaran, 2017). Barrett (2014) addressed important questions raised when the ecology of habits is presented as a dynamical system (e.g. models with recurrent neural

networks as controllers) challenging the conventional notion of habits and yielding a number of insight that must be treated in the future. Another modelling approach was taken by Egbert and Barandiaran (2014) who instantiated the concept of habit explained above using a novel habit-based controller. In this framework, a habit is formed through the plastic changes resulted by the continuous reenacting of the self-sustaining sensorimotor state trajectory emerging from the sensorimotor coupling between the robot and the environment. One important different between the two controllers is the scale in which they are modeled: the former is modeled in the neuro-dynamic level and the latter is modeled at the level of the sensorimotor dynamic.

Model

The robot controller proposed by Egbert and Barandiaran (2014) is a dynamical construct called the Iterant Deformable Sensorimotor Medium (IDSM). The IDSM coupled to robot sensors and motors influences the rate of change of the motors as a function of the current sensorimotor state, the rate of sensorimotor state change, and the current state of mapping as time passes (see Equation 6). The influence can then be understood as a "continuous transfer function that can be depicted as a multidimensional vector space (the dimension been the sensory and motor variables)" (Barandiaran, 2017, p. 421). The controller was designed such that current sensorimotor patterns of activity increases the likelihood of repeating similar sensorimotor patterns in the future. Broadly, as a robot embedded with an IDSM moves through sensorimotor state trajectories, a set of records of the sensorimotor dynamics, known as nodes, are created and modified. Formally, each node is a tuple of two vectors and a scalar in a normalized sensorimotor space, $N = \langle N_{\mathbf{p}}, N_{\mathbf{v}}, N_{\omega} \rangle$, where the elements indicate the sensorimotor state associated with the node or "position", the rate of sensorimotor change or "velocity", and the "weight" of the node, respectively. The normalized sensorimotor space is defined such that the range of all sensor and motor values are linearly scaled to lie in $[0, 1]$. A new node is created when the weighted density of nodes, $\phi(\mathbf{x})$, near the current sensorimotor state, \mathbf{x} , meets the condition expressed by Equation 1. This density function can be understood as a measure of how similar the current sensorimotor state is to the set of nodes created previously. It is calculated using the distance from every node to the current sensorimotor state bounded by the sigmoidal function in Equation 3, and whose result is scaled by the sigmoidal function of the weights defined in Equation 2.

$$\phi(\mathbf{x}) = \sum_N \omega(N_w) \cdot d(N_{\mathbf{p}}, \mathbf{x}) < k_t \quad (1)$$

$$\omega(N_w) = \frac{2}{1 + \exp(-k_w N_w)} \quad (2)$$

$$d(N_{\mathbf{p}}, \mathbf{x}) = \frac{2}{1 + \exp(k_d \|N_{\mathbf{p}} - \mathbf{x}\|^2)} \quad (3)$$

A weight represents the overall influence of the node in the rate of change of the motor output. After a node is created, its weight changes according to Equation 4. k_{dec} is a decay term which allows a steady decrease of the influence of the node when sensorimotor trajectories near that node are not re-enacted by the robot. k_{rein} is a reinforcing term which allows the increase of the influence of the node when its position is closed to the current sensorimotor state.

$$\frac{dN_w}{dt} = -k_{dec} + k_{rein} d(N_{\mathbf{p}}, \mathbf{x}) \quad (4)$$

A node is activated t_{act} units of time after creation, that is, the IDSM output is not influenced by the node during such a time, although its weight is updated. Equations 6 and 6 describes the so-called "influence" of the IDSM which in previous works has been the mapping function from the current sensorimotor state to a rate of motor change. The influence is composed by an "velocity" factor and a "attraction" factor. The former is the motor components of $N_{\mathbf{v}}$, and the latter is a force that cause the system to visit regions of the sensorimotor space with a higher density of nodes so that changes in the environment and perturbations can be compensated. Note that only the motor component of the "attraction" factor is used. The velocity factor pushes the sensorimotor dynamic away from the node while the attraction factor draws the sensorimotor dynamic toward the node (see Egbert and Barandiaran (2014) for more details). As can be seen, both factors are scaled by the distance function and the weight function defined by Equation 3 and 2, respectively. Once summed and scaled, the result is also scaled by the density of the nodes at the current sensorimotor state defined by Equation 1.

$$\mathbf{I} = \frac{1}{\phi(\mathbf{x})} \sum_N \omega(N_w) \cdot d(N_{\mathbf{p}}, \mathbf{x}) \cdot (Vel + Att)^\mu \quad (5)$$

$Vel : N_{\mathbf{v}}$
 $Att : A(N_{\mathbf{p}} - \mathbf{x}, N_{\mathbf{v}})$

$$A(\mathbf{a}, N_{\mathbf{v}}) = \mathbf{a} - \left(\mathbf{a} \bullet \frac{N_{\mathbf{v}}}{\|N_{\mathbf{v}}\|} \right) \frac{N_{\mathbf{v}}}{\|N_{\mathbf{v}}\|} \quad (6)$$

The controller was originally designed such that $\frac{d\mu}{dt} = \mathbf{I}$, where μ represents when motor components are only used, but the Equation 7 presents another alternative. As stated by Egbert (2018), the motor output is determined either by the influence function of the IDSM (\mathbf{I}) or by a random process (\mathbf{R}). The sigmoidal function, $s(\psi)$, is defined by the Equation 8 where ψ determines the local density of the nodes.

Therefore, if the robot visits unfamiliar sensorimotor states, the motor activity is to be random. 100 times per unit of time, \mathbf{R} has a p chance of being set to a random vector drawn from a Normal Distribution with mean 0 and variance r , that is, $\mathcal{N}(\mu = 0, \sigma^2 = r)$.

$$\frac{d\mu}{dt} = (1 - s)\mathbf{I} + s\mathbf{R} \quad (7)$$

$$s(\psi) = \frac{1}{1 + \exp(k_s\psi - k_s)}; \psi = \sum_N d(N\mathbf{p}, \mathbf{x}) \quad (8)$$

The IDSM can be used to train functional habits. Egbert and Barandiaran (2014) presented a two-phase experiment: (1) in the training phase, a robot executes a task using a Braitenberg vehicle-inspired controller, whilst the IDSM tracks the sensorimotor state trajectories; (2) in the free-action phase, the motor activity is only determined by the influence of the IDSM. If the robot is trained from different initial conditions, the robot continues performing similar patterns of behavior after training regardless the initial conditions.

Recently, Egbert (2018) investigated agency and norm-driven behavior using the IDSM as a framework to model a sensorimotor individual. The experiment consisted of a robot coupled with an IDSM with a random-based exploratory phase, that is, the motor activity was determined by Equation 7 in order to perform a task. In this case, neither a training phase nor a random initialization of the IDSM was needed but a good tuning of the parameters.

A remarkable difference between these approaches is that, whilst Egbert and Barandiaran (2014) used the IDSM to “exploit” the regularities among the sensorimotor contingencies, body and environment, Egbert (2018) used an augmented controller to also “explore” the sensorimotor space. Therefore, whilst only the re-enacting of sensorimotor trajectories is needed in the former for habit formation, random behavior is also needed in the latter for the emergence of precarious but stable sensorimotor patterns of behavior.

Other efforts have been recently done to understand how the habits formed using a simplification of the IDSM is biased by the number of nodes, nodes position and velocity (Woolford and Egbert, 2019). In the current work, an experiment is performed to show how different forms of the random motor activity (\mathbf{R}) employed in unfamiliar sensorimotor states bias the formation of habits so as to take different qualitative forms with different quantitative properties. The details are described in the next section.

Experiment

The experiment consists of varying the parameters of the random process, that is: p (i.e. the probability of setting a new random vector R in every time unit) and r (i.e.

the variance of the normal distribution from where R is drawn), such that $p \in \{0.0, 0.01, 0.02, \dots, 0.39, 0.40\}$ and $r \in \{0, 1, 2, \dots, 9, 10\}$. For each condition in $r \times p$, the formation of self-organized habits is allowed 100 times, randomizing the robot initial conditions in each run. We simulate an IDSM with a random-based exploratory phase coupled with a simple robot situated in a one-dimensional periodic environment. The robot has one motor which determines its velocity $m = \frac{dx}{dt}$, where x is the robot position, and the sensor activity is activated according to $\frac{1}{1+x^2}$. The motor activity is bounded such that $m \in [-1, 1]$, and the environment dimension is 4, so that the periodic variable $x \in [-2, 2]$. Table 1 lists the value of the controller parameters used during the experiment.

k_t	k_ω	k_d	k_{dec}	k_{rein}	k_s	t_{act}
1	0.0025	1000	-1	10	20	1

Table 1: Values of the controller parameters.

Each run lasts 200 units of time. Habit formation is allowed during all the time such that the position and velocity of each created node are recorded. Motor activity, robot position, and sensorimotor trajectory are only recorded from time 100 onward. The frequency and amplitude of the motor time series is then computed using the Fast Fourier Transform. Motor activity mean and standard deviation are also calculated. The processed data is used to train a Decision Tree Classifier with 2255 self-organized habits classified by hand. The model is trained splitting the data into two data sets whose elements are randomly chosen. The training data set constituted by 90% of the data, and the testing data set constituted by 10% of the data. The accuracy of the Decision Tree classifier is 0.9956.

Results

The self-reinforcing sensorimotor patterns can be distinguished by the sensorimotor trajectories in the sensorimotor space which produce qualitatively different behaviors in the environment. We have chosen to classify habits into three categories based on the similarities among their sensorimotor trajectories and the patterns generated in the environment. To understand the qualitative features, some examples are presented.

Examples

The figures of the following examples were created choosing arbitrary values for p and r . Each Figure consists of the sensorimotor space, the time series of the motor activity and robot position. Blue circles depict node position and white arrows show the direction of node velocity. The color intensity represents the value of the node weight, i.e. the bluer the node, the higher its weight. Gray dots are plotted when

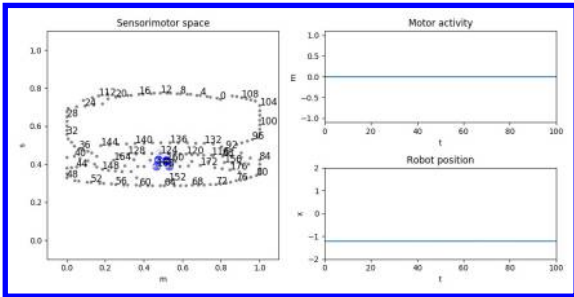


Figure 1: Example of a Static behavior obtained using $p = 0.1$ and $r = 3$.

weights have been degraded so that those nodes are imperceptible. The numbers located beside some nodes are useful to visualize node creation over time.

Static Behavior: This behavior is characterized by a point-like sensorimotor trajectory and zero motor activity. This stationary sensorimotor trajectory is usually formed after randomly driven changes in motor activity. Figure 1 shows how those changes end up in this kind of behavior. The initial possible sensorimotor trajectory runs from node 0 to 108. New nodes were created after the random-based exploratory phase was applied and the current sensorimotor state was different enough from the ones previously experienced such that the condition in Equation 1 was met. For instance, nodes 116 and 132. Over time, these nodes formed new narrow sensorimotor trajectories. The weights of distant nodes from such new trajectories were degraded such that the motor activity was determined by the influence of the newest nodes. At some time, the robot reaches an equilibrium when the velocities of the latest created nodes were opposite as can be seen for the directions of the white arrows.

Monotonic Behavior: This behavior is characterized by a linear sensorimotor trajectory whose motor value is constant, either -1 or 1 , and sensor value varies along the sensor range. Figure 2 depicts how this kind of behavior emerged after large random changes in motor activity. Many nodes were created over time, but the nodes located around the sensorimotor trajectory were reinforced sufficiently to overcome other possible trajectories (for example, the sensorimotor trajectory possibly defined by the nodes around $m = 1$). As can be seen, the nodes influencing the sensorimotor trajectory were not generated successively but they were created and reinforced in different moments.

This type of behavior also emerged due to other reasons. Figure 3a shows that this sensorimotor pattern can emerge when the random vector is drawn from normal distribution with variance 0 such that the motor-components of the terms *Velocity* and *Attraction* in Equation 6 are always zero. Here, we have a first examples of how the parameters of the random motor process can bias habit formation. As we shall

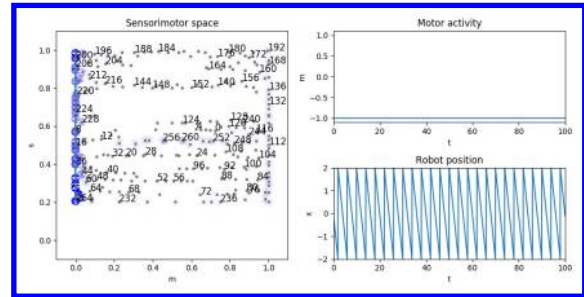


Figure 2: Example of a monotonic behavior obtained using $p = 0.38$ and $r = 9$.

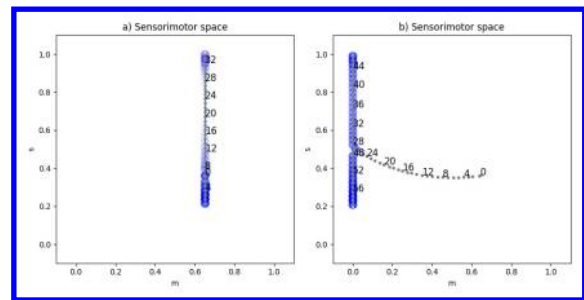


Figure 3: Examples of a monotonic behavior obtained using a) $p = 0.01$ and $r = 0$, and b) $p = 0.0$ and $r = 1$

see below this bias does not only hold for $p = 0$, but also for small values of p . Therefore, the randomly-set initial condition of the motor value remains constant over time. Figure 3b shows that this behavior can also emerge when the probability of changing vector \mathbf{R} is zero such that either the highest or the lowest motor value is reached. Therefore, new nodes are created and reinforced along the sensor range.

Oscillatory Behavior: This behavior is characterized by a closed sensorimotor trajectory. This trajectory produces an oscillatory motor activity and, therefore, an oscillatory robot displacement as can be seen in Figures 4 and 5. Both figures were produced with the same values of p and r but different random seed. The nodes defining the sensorimotor trajectory in Figure 4 were created almost successively over time, while the latest created nodes in Figure 5 constrained the formation of other possible trajectories due to the opposite directions of nodes velocities (see nodes 132 – 184). The random motor activity biased the final sensorimotor pattern either defining slightly changes in motor activity (for instance, nodes 99 – 108 in Figure 4) or driving the node creation to specific regions of the sensorimotor space (for instance, nodes 84 – 88 in Figure 5).

The class of some trajectories could not be so clear. For example, a sensorimotor trajectory could be formed with most of the nodes located around either $m = -1$ or $m = 1$, and just a few nodes around another motor value whose

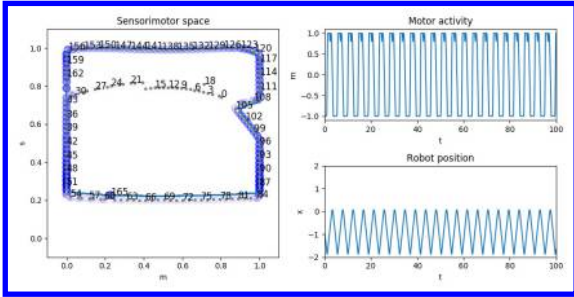


Figure 4: Example of an oscillatory behavior obtained using $p = 0.02$ and $r = 2$.

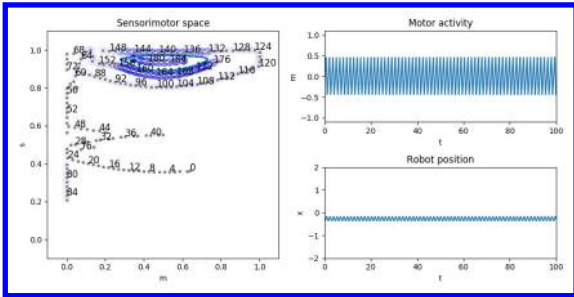


Figure 5: Example of an oscillatory behavior obtained using $p = 0.02$ and $r = 2$.

weights influence the motor activity. This behavior might be wrongly classified as an oscillatory behavior if only the sensorimotor trajectory is used as the reference. However, the robot would be performing a monotonic behavior because the influence of the smaller set of nodes is usually only enough to slightly increase or decrease robot speed.

The values of p or r used to generate the figures do not entail that qualitatively similar behaviors are more likely to emerge under these conditions. We present the results of modulating both parameters in order to know what kind of behaviors emerge regardless robot initial conditions.

Statistical Results

Static Behaviors: Figure 6 shows that static behaviors can emerge neither when $r = 0$ and p varies nor when r varies and $p = 0.0$. Note that the higher the value of p and the lower the value of r , the more likely is the emergence of these kinds of habits. Therefore, continuous changes of the components of \mathbf{R} drawn from a normal distribution with a small standard deviation, σ , result in fast small steps in the sensorimotor space that avoids the formation of wide sensorimotor trajectories.

Monotonic Behaviors: Figure 7 shows that only monotonic behaviors emerge when $r = 0$ and p varies, and when r varies and $p = 0.0$. In the first case, $r = 0$ implies that R is a vector constituted by zeros. The random-based exploratory

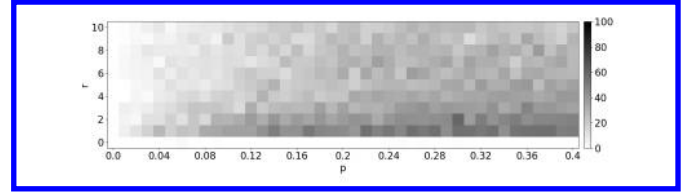


Figure 6: Grayscale representation of the number of static behaviors emerged per each couple of p and r .

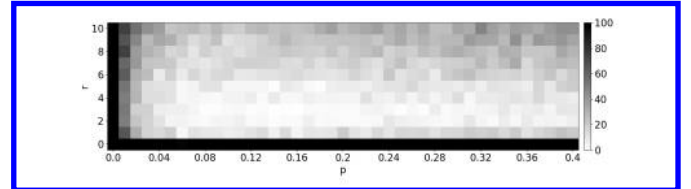


Figure 7: Grayscale representation of the number of monotonic behaviors emerged per each couple of p and r .

phase tries to influence motor activity at the beginning of the time because $s(\phi) \approx 1$ and then $\frac{d\mu}{dt} = \mathbf{R}$. The motor activity is constant with the same value of the initial condition. At some time, $s(\phi) \approx 0$ and then $\frac{d\mu}{dt} = \mathbf{I}$; however, the motor-components of $N_{\mathbf{V}}$ are zero because those components of the created nodes are zero too. Therefore, motor activity remains constant. New nodes could be created along the sensor value range but the IDSM do not take over motor activity at any time (see Figure 3a). On the other hand, in the second case, $p = 0.0$ implies that R is constant over time, although was defined randomly. Therefore, the only possible sensorimotor trajectory is formed when the motor value reaches one limit of the motor range after a transient (see Figure 3b). Notice that behaviors formed when $r = 0$ and p keeps fixed motor state over time, whilst behaviors formed when r varies and $p = 0.0$ maximize motor state during a transient. Finally, monotonic behaviors are also considerably likely when continuous changes of the components of \mathbf{R} drawn from a normal distribution with a high standard deviation result in fast large steps in the sensorimotor space that allows the reinforcing of the nodes in the limits of the sensorimotor space.

Oscillatory Behaviors: Figure 8 shows that this behavior is highly probable to emerge for any couple of parameters, except when $r = 0$ and p varies or when r varies and $p = 0.0$. However, oscillatory behaviors are more likely when sporadic changes (small values of p ; see from 0.02 to 0.12, for instance) of the components of \mathbf{R} drawn from a normal distribution, whose variance can be chosen along the set but zero, are performed. These changes result in slow small steps in the sensorimotor space allowing the formation of closed sensorimotor trajectories. As can be seen in Figure 9, outliers or not-formed habits, usually characterized by

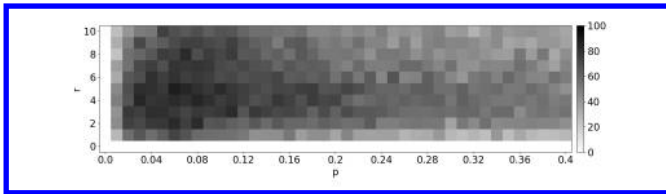


Figure 8: Grayscale representation of the number of oscillatory behaviors emerged per each couple of p and r .

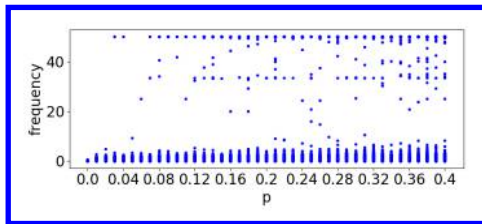


Figure 9: Frequencies distribution considering all the values of r per each value of p .

high frequencies, are less likely for small values of p .

Discussion

We have shown how simply by changing the parameters of the random motor activity that is engaged when the sensorimotor state is unfamiliar, it is possible to bias qualitative and quantitative properties of the patterns of sensorimotor activity that become ingrained as habits in the IDSM. For low values of p , monotonic behaviors, where motor values are maximized, are the most likely to emerge. When $r = 0$, other monotonic behaviors are the most likely, now where a random motor velocity is maintained. Other values of p and r produce more subtle variations in the types of emergent habits, with regions that are more likely to produce oscillatory behaviors. These behaviors produce motor activities at different frequencies. Outliers and not-formed habits characterized commonly by high frequencies are more likely for high values of p . The random-based exploratory phase produces changes such that the final behavior could be driven to a specific region of the sensorimotor space, or it could contribute directly to the shape of the such a behavior. For instance, the behaviors depicted in Figure 4 and Figure 5, respectively.

We can interpret this model in term of how random motor babbling makes certain kind of sensorimotor organization available for the agent. First, notice that sensorimotor contingencies can be described as lawful regularities of sensory stimulation, internal activity, and motor activity. However, according to Buhrmann et al. (2013), defining formally this concept has resulted problematic due to the range of useful interpretations. They introduced the operational definition of four kinds of sensorimotor contingencies by consid-

ering how the sensorimotor flow changes in relation to the body and environment, internal activity, functional behavior, and normative dimension. The four distinct notions are briefly summarized (Buhrmann et al., 2013): (1) Sensorimotor environment: the set of all possible sensor states obtained as function of given motor command varying freely, i.e. without considering internal activity, for a particular embodiment and external environment. (2) Sensorimotor habitat: the set of all actual sensorimotor trajectories that can be generated considering internal activity, i.e. closing the loop, given a range of boundary conditions and parameters (3) Sensorimotor coordination: any reliable sensorimotor pattern (e.g. stable trajectories, transients) that contribute functionally to the goals of an agent. (4) Sensorimotor strategies: organization of sensorimotor coordination patterns regularly used by the agent because it has been previously chosen as consequence of having been normatively evaluated.

Now, consider what kind of sensorimotor contingencies are available for the agent when the sensorimotor trajectories are formed. In the random, exploratory phase, the available sensorimotor contingencies can be described as the immediate sensory consequences of random motor changes (comparable to the sensorimotor-environment as described by Buhrmann et al. (2013)). Thus, the agent is randomly “navigating” the sensorimotor environment to generating a history of sensorimotor states that bias the formation of the possible sensorimotor trajectories due to the history-dependent plasticity of the behavior-generating medium. In the non-stochastic phase, the sensorimotor dynamics and thus the experienced sensorimotor contingencies are available only through specific patterns of agent-environment interaction driven by the influence of the controller. Therefore, the results shown in Figure 6, 7 and 8 could be considered as the sensorimotor habitat since they depict the actual sensorimotor trajectories generated by the closed-loop system for a range of two relevant initial parameters: p and r . Notice, then, that the model can be seen as capturing the development of a sensorimotor habitat from the random traversing of the sensorimotor environment.

In this preliminary model, even though we did not include any simulation of biological autonomy—no biological essential variables or viability limits—it is easy to imagine how some of these behaviors would be better suited to the survival of the biological individual. As example, consider the skin brain thesis which states that the early nervous system and sensorimotor organization of the organisms emerge from the evolution of a transverse net organization extended across the body that produced coordinated patterns of contraction and extension (Keijzer, 2015). This thesis first considers a contractile tissue extended across the body of the organisms. This surface is an excitable medium across which excitatory and inhibitory activity can produce self-organized patterns of contraction-extension. These patterns are produced by means of a multicellular organization

composed by cells with neural-like features (synaptic signaling to neighboring cells and, later in evolution, to non-neighboring cells) and external individual sensitivity with internal local feedback. The morphology of a multicellular body allows the organism to be sensitive to spatiotemporal dynamics of its own body and, therefore, to spatiotemporal structures of an environment. Consequently, functional patterns of contraction-extension can emerge due to a close organizational loop “that is sensitive to the ongoing dynamical contraction changes across this body and influencing the processes that generate and maintain these same patterns of behavior” (Keijzer, 2015, p. 325). Notice that, even though patterns of contraction and extension are not random as in the IDSM, the thesis stresses the internal spontaneous activity which, firstly, allows the organism to generate patterns of contraction-extension. These patterns enable the organization of an early nervous system which in turn generates them. This self-induced movement generates specific patterns in the sensory surface which in turn influences the early nervous system organization and, thus, the pattern generation such that the organism can cope with external disturbances.

We can compare the skin brain thesis and the model described in this work. In both cases there are many possible sensorimotor individuals that could inhabit a given sensorimotor environment. In the case of an organism with the organization described by the skin brain thesis, a sensorimotor individual thus would emerge from the reenactment of the sensorimotor contingencies generated first by the self-initiated motility, and later by the nervous system dynamic (which depends upon the sensorimotor contingencies it brings about). Notice that any sensorimotor trajectory is biased by the preliminary self-induced motility of the organism that allows the “navigation” of the sensorimotor environment. Similarly, in our experiments, a sensorimotor individual results from the reenactment of the sensorimotor contingencies generated by the random and the node-driven motor activity (the latter depends upon the history-dependent plasticity that the own activity brings about). The formation of any sensorimotor trajectory is biased by the random traversing of the sensorimotor environment. Even though the three qualitative categories of behaviour (static, monotonic and oscillatory) described in the results can be considered trivial, the results shed light on how evolution can bias the parameters of a plastic medium over which sensorimotor individuals emerge, so as to encourage the formation of sensorimotor individuals that are conducive or beneficial to biological survival. This form of evolutionary coupling of biological individual and sensorimotor individual norms underdetermines the patterns of behavior that emerge — unlike hardwired, purely reflexive behavior, the patterns of behavior in the sensorimotor individual remain precarious autonomous entities.

References

- Ashby, W. R. (1952). *Design for a Brain: The Origin of Adaptive Behaviour*. Wiley.
- Barandiaran, X. E. (2007). *Mental Life: conceptual models and synthetic methodologies for a post-cognitivist psychology*, pages 49–90. Imprint Academic.
- Barandiaran, X. E. (2008). *Mental Life: A Naturalized Approach to the Autonomy of Cognitive Agents*. PhD thesis, University of the Basque Country.
- Barandiaran, X. E. (2017). Autonomy and enactivism: Towards a theory of sensorimotor autonomous agency. *Topoi*, 36(3):409–430.
- Barandiaran, X. E. and Di Paolo, E. A. (2014). A genealogical map of the concept of habit. *Frontiers in Human Neuroscience*, 8:522.
- Barrett, N. (2014). A dynamic systems view of habits. *Frontiers in Human Neuroscience*, 8:682.
- Buhrmann, T., Di Paolo, E., and Barandiaran, X. (2013). A dynamical systems account of sensorimotor contingencies. *Frontiers in Psychology*, 4:285.
- Di Paolo, E. A. (2003). *Organismically-Inspired Robotics: Homeostatic Adaptation and Teleology Beyond the Closed Sensorimotor Loop*, pages 19–42. Advanced Knowledge International.
- Di Paolo, E. A., Buhrmann, T., and Barandiaran, X. (2017). *Sensorimotor Life: An enactive proposal*. Oxford University Press.
- Di Paolo, E. A., Noble, J., and Bullock, S. (2000). Simulation models as opaque thought experiments. In Bedau, M. A., McCaskill, J. S., Packard, N., and Rasmussen, S., editors, *Seventh International Conference on Artificial Life*, pages 497–506. MIT Press, Cambridge, MA.
- Egbert, M. D. (2018). Investigations of an adaptive and autonomous sensorimotor individual. In *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 343–350.
- Egbert, M. D. and Barandiaran, X. E. (2014). Modeling habits as self-sustaining patterns of sensorimotor behavior. *Frontiers in Human Neuroscience*, 8:590.
- Egbert, M. D. and Canamero, L. (2014). Habit-based regulation of essential variables. In Sayama, H., Rieffel, J., Risi, S., Doursat, R., and Lipson, H., editors, *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*.
- Keijzer, F. (2015). Moving and sensing without input and output: early nervous systems and the origins of the animal sensorimotor organization. *Biology & Philosophy*, 30(3):311–331.
- Maturana, H. R. and Varela, F. J. (1980). *Autopoiesis and cognition: the realization of the living*. D. Reidel Publishing Company.
- Woolford, F. M. G. and Egbert, M. D. (2019). Behavioral variety of a node-based sensorimotor-to-motor map. *Adaptive Behavior*.

Guiding aggregation dynamics in a swarm of agents via informed individuals: an analytical study

Yannick Gillet¹, Eliseo Ferrante², Ziya Firat¹ and Elio Tuci¹

¹University of Namur, Namur, Belgium

²University of Birmingham, Birmingham, UK
elio.tuci@unamur.be

Abstract

Self-organised aggregation, the formation of large clusters of independent agents, is an important process in swarm robotics systems since it is the prerequisite for more complex collective behaviours. Previous work on self-organised aggregation focused on the study of the individual mechanisms required to allow a swarm to form a single aggregate. In this paper, we discuss an analytical model which looks at the possibility to use the concept of informed individuals to allow the swarm to distribute on different aggregation sites according to proportions of individuals at each site arbitrarily chosen by the designer. Informed individuals are opinionated agents that selectively prefer an aggregation site and avoid to rest on the non-preferred sites. We study environments with two aggregation sites, and consider two different scenarios: one in which the informed individuals are equally distributed in numbers between the two sites; and one in which informed individuals for one type of site are three times more numerous than those on the other site. Our objective is to find out whether and for what range of model parameters the swarm distributes between the two sites according to the relative distribution of informed agents among the two sites. The analysis of the model shows that the designer capability to exploit informed individuals to control how the swarm aggregates depends on the environmental conditions. For intermediate values of the site carrying capacity, a small minority of informed individuals is able to guide the dynamics as desired by the designer. We also show that the larger the site carrying capacity the larger the total proportion of informed individuals required to lead the swarm to the desired distribution of individuals between the two sites.

Introduction

The field of swarm robotics aims at studying and designing self-organising collective behaviours for large groups of relatively simple individuals (Dorigo and Şahin, 2004). Swarm robotics takes inspiration from nature, whereby groups of social insects or other animals rely on proximate mechanisms (simple cognitive heuristics plus local interaction rules) that allow them to exhibit complex collective patterns which tend to be functional to achieve certain tasks (Cazemine et al., 2001). Examples include collective migration, site selection, pattern formation, and task specialization. In biological systems, natural evolution shapes the in-

dividual rules of action underpinning the group collective response. In artificial swarms, such as swarms of robots, artificial evolution can potentially be used to mimic natural evolution in order to automate the design of individual mechanisms (see Harvey et al., 2005; Tuci and Rabérin, 2015; Tuci et al., 2018). Alternatively, the designer can program the behaviour of each individual robot, and evaluate the performance of the collective behaviour at the group (macroscopic) level (Brambilla et al., 2013). However, given the difficulties in predicting the individual actions that result in the desired self-organising collective behaviour, the designer is required to program and evaluate multiple individual controllers before finding the one that underpins the desired group level response. Thus, this approach can be time consuming and largely dependent on the designer's intuitions on what is required to move from the individual to the group level desired behaviour.

A relatively recent idea to increase the degree of controllability of artificial swarms consists in introducing a small proportion of informed individuals which can be used to bias the collective behaviours in the direction specified by the designer (Ferrante, 2013). Informed individuals are opinionated agents that tend to bias any decision making process toward their preferred option. The effect of informed individuals on the groups dynamics have been originally discussed in biological models of collective motion, where a minority of individuals determined to move in a given direction induces the rest of the swarm to opt for their direction of motion (Couzin et al., 2005; Stroeymeyt et al., 2011; Krause and Ruxton, 2011). Informed individuals have been subsequently exploited in artificial swarms mainly as a means to control the system during collective motion (Çelikkanat and Şahin, 2010; Ferrante et al., 2012, 2014). We study the effects of informed individuals in a larger spectrum of self-organised collective behaviour. In particular, in this paper we further explore the effects of informed individuals in the context of self-organised aggregation (Firat et al., 2018).

Generally speaking, in aggregation tasks, individuals have to aggregate on a common location in the environment (Garner et al., 2005, 2008; Bayindir and Şahin, 2009; Correll

and Martinoli, 2011; Gauci et al., 2014). Swarm robotics studies have shown that robot’s controllers in which the individual probability to join and to leave an aggregation site depends on the number of robots perceived by an individual at the site, lead to the emergence of a single aggregate at one site among those available in the environments (Garnier et al., 2009; Campo et al., 2010). In (Halloy et al., 2007), robots controlled by similar principles influence the aggregation dynamics of cockroaches in mixed robot-animal groups. In particular, the robots programmed to preferentially rest on the lighter (rather than on the darker) shelter, induce cockroaches to behave similarly even if the animals would preferentially aggregate on the darker shelter in the absence of robots. The idea that some individuals could influence the aggregation dynamics of a group of autonomous agents, originally discussed in (Halloy et al., 2007) in the context of the robots-cockroaches interaction, has been recently explored in (Firat et al., 2018, 2019) in the context of swarm robotics systems. The authors in (Firat et al., 2018, 2019) have extended the analysis of the aggregation process in a two-site scenario as illustrated in (Garnier et al., 2009; Campo et al., 2010), to the case in which the swarm is characterised by the presence of informed individuals. In (Firat et al., 2018, 2019) the sites have distinctive features that allow the agents to discriminate between the two of them. Informed individuals are programmed to selectively avoid to rest on one of the two sites. Non-informed individuals rest with equal probabilities on both sites. These studies show that with a small proportion of informed individual it is possible to selectively drive the aggregation dynamics on a designer preferred aggregation site.

In this paper, we discuss the results of a mathematical model that looks at aggregation dynamics in a swarm of agents with different proportions of informed and non-informed individuals. Mathematical models are quite frequently used in the study of collective behaviour in artificial swarms to avoid the time and computational costs that robotics and others agent-based models undergo to explore the effects of a wide range of experimental conditions (Brambilla et al., 2013). Mathematical models of self-organised aggregation include geometric models (Bayindir and Şahin, 2009) and Markov chains (Soysal and Şahin, 2007). To study other collective behaviours, common approaches to modelling include ordinary differential equations (Montes de Oca et al., 2011; Valentini et al., 2015), stochastic differential equations such as rate equations (Lerman and Galstyan, 2002), chemical reaction networks (Valentini et al., 2015), Fokker-Planck and Langevin equations (Hamann and Wörn, 2008), and control theory (Hsieh et al., 2008), among others. Our model uses a system of ordinary differential equations to study how informed individuals can be used in the context of aggregation to distribute the agents of a swarm between two distinctive aggregation sites (one perceived by the individuals

as black and the other as white) according to two arbitrary rules specified by the designer. There are two types of informed individuals in our model: the “informed for black” individuals which rest only on the black site, and the “informed for white” individuals which rest only on the white site. Excluding the informed individuals of any type, the rest of the swarm is made of non-informed individuals, that is agents that rest on both aggregation sites with equal probabilities. Both informed and non-informed individuals leave an aggregation site with a probability given by a non-linear function of the density of individuals at the site.

Our objective is to find out whether and eventually for which parameter range the swarm distributes between the two sites according to the relative proportion of one type of informed individuals with respect to the other type, by keeping the total proportion of informed individuals as small as possible. We analyse the system for different total percentage of informed individuals in the swarm, from 0% to 100% informed individuals. For each percentage of informed individuals, we systematically vary the relative proportion of informed individuals of one type with respect to the proportion of individuals of the other type. In this paper, we report the results of two representative scenarios: one in which informed individuals are equally distributed in numbers between the two sites; and one in which the informed individuals for one type of site are three times more than the informed individuals for the other type of site. The first scenario has been chosen to represent the designer aims to induce the agents to aggregate in equal proportion on both sites. The second scenario has been arbitrarily chosen among those representative of the designer intention to induce the agents to aggregate in different proportions on each site. For each of the two scenarios illustrated in this paper, we vary the total proportion of informed individuals from 0% to 100% of the swarm population size. Moreover, we analyse the systems for different values of the site carrying capacity, that is the total number of individuals that can simultaneously rest on a site. We are interested in identifying the conditions whereby agents equally split on the two aggregation sites when both types of informed individuals are equally represented in the swarm, and the conditions whereby aggregation dynamics see agents aggregated 75% on a site and 25% on the other site, when one type of informed individuals is three times more represented than the other type. The results of this study shows that there are parameters’ values for which the distribution of individuals between the two sites matches the relative proportion of one type of informed individuals with respect to the other type. In particular circumstances, the desired aggregation dynamics can be observed with a small minority of informed individuals in the swarm. In other words, the analysis of the mathematical model indicates that informed individuals are a potentially effective means to control the aggregation dynamics in swarms of autonomous agents. In section Con-

clusions, we discuss the significance of our results for the swarm robotics community, and we explain how we intend to use these finding in our future research works.

Methods

In this section, we describe the system of Ordinary Differential Equations (ODEs) used to investigate the effects of different proportions of two different types of informed individuals on the aggregation dynamics in a scenario with two sites, a black and a white site. We draw inspiration from another ODEs system originally discussed in (Amé et al., 2004), and subsequently extended in (Amé et al., 2006) to model the aggregation dynamics observed in cockroaches. The distinctive feature of both the above cited models is that the individual probability of leaving a site is a non-linear function of the number of individuals currently resting at that site. Our departure point is the Amé et al. (2006)'s model, where two aggregation sites, with same characteristics, are symmetrical locations for aggregation for a group of N equal type individuals. The Amé et al. (2006)'s model is the following:

$$\dot{N}_i = -N_i \lambda_i + \mu \left(1 - \frac{N_i}{S}\right) N_{ext}; \quad (1)$$

with

$$\lambda_i = \frac{\epsilon}{1 + \gamma \left(\frac{N_i}{S}\right)^2}; \quad N_{ext} = \left(N - \sum_{i=1}^p N_i\right); \quad (2)$$

where N_i is the number of individual resting on site i , λ_i is the individual probability to leave site i , the parameter $\epsilon = 0.01s^{-1}$, the parameter $\gamma = 1667$, S is the maximum number of individuals that a site can host (i.e. the site carrying capacity), $\mu = 0.001s^{-1}$ is the rate of entering a site, N_{ext} is the number of individuals outside the sites, and $p = 2$ is the number of sites. The analysis of this model shows that the agents form a single aggregate only when each aggregation site can host more that the totality of the swarm's individuals. The model also predicts how the agents distribute in different environments varying for the number of aggregation sites and the diameter of each site bearing upon the site capacity to host individuals (see Amé et al., 2006).

We modified the system in Eq. 1 to take into account two novel features that distinctively characterised our study: that is, the differences between the two sites, one of which is perceived by the individuals as black, and the other as white, and the presence of two different types of informed individuals. With the introduction of colour differences between the two sites, the total number of individuals in a group N is given by $N = N_b + N_w + N_{ext}$, with N_b and N_w being the number of individuals resting on the black and on the white site, respectively.

Defining $\sigma = S/N$, $x_b = N_b/N$ and $x_w = N_w/N$, with $N_{ext} = N - N_b - N_w$, leads us to the following system:

$$\begin{cases} \dot{x}_b = -x_b \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) (1 - x_b - x_w) \\ \dot{x}_w = -x_w \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) (1 - x_b - x_w) \end{cases} \quad (3)$$

with

$$\lambda_b = \frac{\epsilon}{1 + \gamma \left(\frac{x_b}{\sigma}\right)^2}; \quad \lambda_w = \frac{\epsilon}{1 + \gamma \left(\frac{x_w}{\sigma}\right)^2}; \quad (4)$$

where λ_b and λ_w refer to the probability of leaving the black and the white site, respectively. As shown in Eq. 3, the system is independent of N and depends only on the fraction of individuals on the two sites.

The distinction between informed and non-informed individuals is introduced into the system with the notation i_w (informed for white) for informed individuals that do not rest on the black site, i_b (informed for black) for informed individuals that do not rest on the white site, and ni (non-informed) for non-informed individuals, who can potentially rest on both sites. With this distinction in place, ρ_{i_b} and ρ_{i_w} are the proportion of informed individuals of type i_b and i_w , respectively. $x_b^{i_b}$ refers to the fraction of individuals on the black site that are of type i_b ; $x_w^{i_w}$ refers to the fraction of individuals on the white site that are of type i_w ; x_b^{ni} refers to the fraction of individuals on the black site that are of type ni ; and x_w^{ni} refers to the fraction of individuals on the white site that are of type ni . The fraction of individual on the black site (x_b) and on the white site (x_w) is then written as:

$$\begin{cases} x_b = x_b^{i_b} + x_b^{ni} \\ x_w = x_w^{i_w} + x_w^{ni} \end{cases} \quad (5)$$

since, by definition, informed individuals of type i_w never rest on the black site, and informed individuals of type i_b never rest on the white site.

Generalising Eq. 3 to the case with informed and non-informed individuals gives

$$\begin{cases} \dot{x}_b^{i_b} = -x_b^{i_b} \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) x_{ext}^{i_b} \\ \dot{x}_b^{ni} = -x_b^{ni} \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) x_{ext}^{ni} \\ \dot{x}_w^{i_w} = -x_w^{i_w} \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) x_{ext}^{i_w} \\ \dot{x}_w^{ni} = -x_w^{ni} \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) x_{ext}^{ni} \end{cases} \quad (6)$$

where $x_{ext}^{i_b}$, $x_{ext}^{i_w}$, and x_{ext}^{ni} are the fraction of individuals of type i_b , i_w , and ni that are outside the two sites. These fractions can be expressed in the following way

$$\begin{cases} x_{ext} = 1 - x_b - x_w = 1 - x_b^{ni} - x_b^{i_b} - x_w^{ni} - x_w^{i_w} \\ x_{ext}^{i_b} = \rho_{i_b} - x_b^{i_b} \\ x_{ext}^{i_w} = \rho_{i_w} - x_w^{i_w} \\ x_{ext}^{ni} = x_{ext} - x_b^{i_b} - x_w^{i_w} \\ \quad = (1 - \rho_{i_b} - \rho_{i_w}) - x_b^{ni} - x_w^{ni} \end{cases} \quad (7)$$

Finally, substituting Eq. 7 into Eq. 6 we obtain the following system:

$$\begin{cases} \dot{x}_b^{i_b} = -x_b^{i_b} \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) (\rho_{i_b} - x_b^{i_b}) \\ \dot{x}_b^{i_w} = -x_b^{i_w} \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) ((1 - \rho_{i_b} - \rho_{i_w}) - x_b^{i_w} - x_w^{i_w}) \\ \dot{x}_w^{i_b} = -x_w^{i_b} \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) (\rho_{i_w} - x_w^{i_b}) \\ \dot{x}_w^{i_w} = -x_w^{i_w} \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) ((1 - \rho_{i_b} - \rho_{i_w}) - x_w^{i_w} - x_b^{i_w}) \end{cases} \quad (8)$$

In the particular case when all the individuals of the group are informed (i.e. $\rho_{i_b} + \rho_{i_w} = 1$), this set of equations reduces to

$$\begin{cases} \dot{x}_b = -x_b \lambda_b + \mu \left(1 - \frac{x_b}{\sigma}\right) (\rho_{i_b} - x_b) \\ \dot{x}_w = -x_w \lambda_w + \mu \left(1 - \frac{x_w}{\sigma}\right) (\rho_{i_w} - x_w) \end{cases} \quad (9)$$

The set of equations illustrated in Eq. 8, is solved numerically to find equilibrium states (i.e., when $\dot{x} = 0$). Equilibrium states are studied with respect to the key parameters σ , ρ_{i_b} and, ρ_{i_w} . The results are discussed in next section.

Results

In this section, we show the results of our analysis, by discussing the equilibrium states of Eq. 8, for different sets of values for the parameters σ , ρ_{i_b} , and ρ_{i_w} . We remind the reader that the parameter σ is the ratio between the site carrying capacity S and the swarm size N . When $\sigma = 1$ each aggregation site can host as many individuals as the swarm size; when $\sigma < 1$, each aggregation site can host fewer individuals than the swarm size; when $\sigma > 1$, each site can host more individuals than the swarm size. ρ_{i_b} and ρ_{i_w} refer to the proportion of individuals of type i_b (informed for black) and i_w (informed for white), respectively. We also remind the reader that our objective is to find out the set of parameters for which the individuals distribute between the two sites according to the relative proportion of one type of informed individuals with respect to the other type. We are also particularly interested in finding what is the critical value of $\rho_i = \rho_{i_b} + \rho_{i_w}$ (i.e. the proportion of informed individuals) above which this objective is realized, and how this changes with respect to σ . For example, when $\rho_{i_b} = 0.3$ and $\rho_{i_w} = 0.3$ we expect 50% of the individuals on the white site and 50% of the individuals on the black site, and we would like to know how much we can decrease both ρ_{i_b} and ρ_{i_w} and still maintain this allocation.

When there are no informed individuals in the swarm ($\rho_{i_b} = 0$ and $\rho_{i_w} = 0$), our model reduces to the original (Amé et al., 2006)'s model. As in (Amé et al., 2006), we also find out that for $\sigma < 1$, the swarm equally distribute between the two sites. However, when $\sigma > 1$ the individuals are able to make a collective decision and to aggregate either on the black or on the white site.

When the entire swarm is composed of informed individuals (i.e. $\rho_{i_b} + \rho_{i_w} = 1$, see also Eq. 9), the fraction of individuals aggregated on the black site (i.e., x_b) is shown in

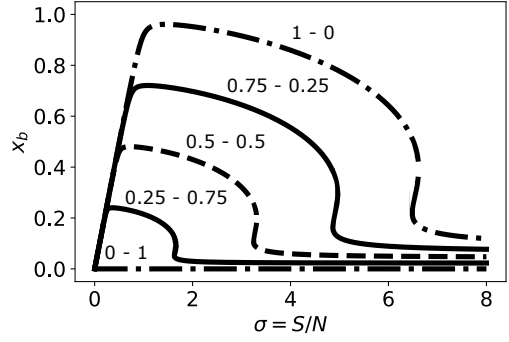


Figure 1: Graph showing the steady state for x_b when σ varies from 0 to 8 for different values of the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$ when the swarm is made of only informed individuals ($\rho_{i_b} + \rho_{i_w} = 1$). The numbers above each line indicate the fraction of informed individuals of types i_b and i_w . Dashed line: $\frac{\rho_{i_b}}{\rho_{i_w}} = 1$. Continuous lines: $\frac{\rho_{i_b}}{\rho_{i_w}} = 3$ or $1/3$. Dashed-dotted lines: $\rho_{i_b} = 0$ or $\rho_{i_w} = 0$.

Figure 1. This graph represents the steady state for x_b when σ varies from 0 to 8, and for different values of the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$.

In the low σ range, when each site is not big enough to host all the corresponding informed individuals (the black site for individuals of type i_b , and the white site for individuals of type i_w), the individuals allocate themselves to both sites until they reach the site carrying capacity. This trend does not depend on the relative ratio between $\frac{\rho_{i_b}}{\rho_{i_w}}$, therefore for low values of sigma informed agents are not able to influence the aggregation dynamics. When σ surpasses a critical value that depends on $\frac{\rho_{i_b}}{\rho_{i_w}}$, each site becomes big enough to host all the corresponding informed individuals. Steady-state dynamic for different $\frac{\rho_{i_b}}{\rho_{i_w}}$ are qualitatively different but follow a similar trend. Up to a another critical value of σ , again dependent on $\frac{\rho_{i_b}}{\rho_{i_w}}$, most individuals simply aggregate on the site they prefer. This is the regime in which informed agents have a maximal influence on the dynamics. However, above this new critical σ , individuals are no longer able to aggregate at all. This analysis reveals that, as it happened for the original model discussed in (Amé et al., 2006), and regardless of the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$, environmental parameters such as the site carrying capacity strongly influences the aggregation dynamics and that informed agents can guide self-organisation only in a limited range of this parameter. For example, when the aggregation site becomes too large, the probability to aggregate on a site, which depends on the site current density, tends to remain too low to trigger the aggregation process. In other words, the density of individuals on each site never reaches a critical value to induce the individuals to aggregate on a site. Thus, the individuals tend to disperse rather than aggregate. For each site, the transition between the two regimes illustrated above is determined by the number of informed individuals that are attracted by

that particular colour: the higher the number of informed individuals of each type, the higher the size of the site required to trigger the regime change. For each ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$, there exists a maximum size σ_{max} corresponding to the point of regime change. For example, when 75% of individuals are of type i_b and 25% are of type i_w , in order to have all of them aggregated on the corresponding preferred site, σ has to be smaller than 1.6 (see Figure 1, continuous lines). When $1.6 < \sigma < 4.8$, the white site becomes too large to trigger aggregation for the individuals of type i_w , while the individuals of type i_b are enough to cope with the dimension of their corresponding aggregation site. When $\sigma > 4.8$, even the black site becomes too large to trigger aggregation.

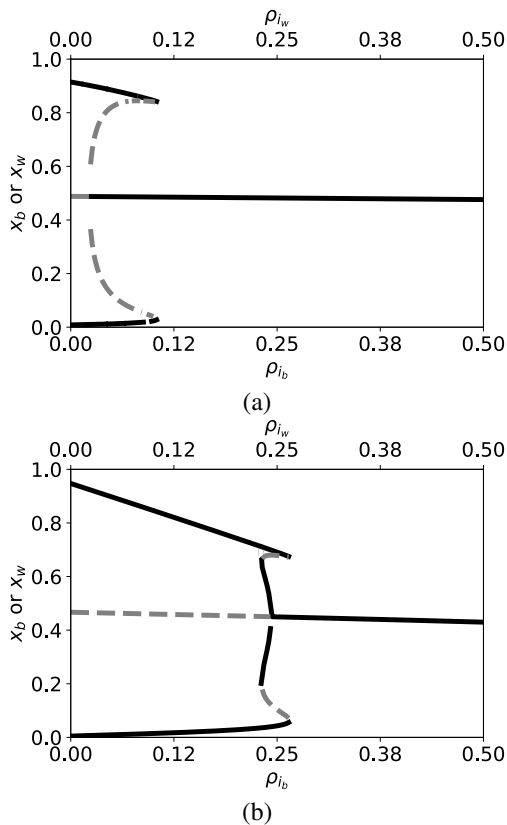


Figure 2: (a) Fraction of individuals on the black and on the white site, when $\frac{\rho_{i_b}}{\rho_{i_w}} = 1$ for (a) $\sigma = 1$ and (b) $\sigma = 2$. Black continuous lines: stable solutions. Dashed grey lines: unstable solutions.

The analysis carried out so far tell us that the most interesting regime is the one in which σ is in a range (dependent on $\frac{\rho_{i_b}}{\rho_{i_w}}$) that allows the individuals to aggregate in their respective preferred sites. In this range, we ask ourselves whether we can now have a hybrid swarm composed of informed and non-informed individuals, and whether informed individuals can still guide the dynamics in a similar way as when the swarm was only composed of informed

individuals. We thus proceed by analysing the system for $\sigma = 1$ and $\sigma = 2$ for different proportions of informed individuals in the swarm, and for two different values of the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$. In all the figures that will follow, we will report stable equilibria with continuous black lines, and unstable equilibria with dashed grey lines.

Figure 2a (resp. Figure 2b) reports results with the ratio $\frac{\rho_{i_b}}{\rho_{i_w}} = 1$ for $\sigma = 1$ (resp. $\sigma = 2$), that is for each proportion of informed individuals in the swarm, 50% of them are of type i_b and 50% are of type i_w . The graph shows that the individuals aggregate on one site only (i.e. either the black or the white site), until a critical value for the total proportion of informed individuals ρ_i (about 24% of the swarm for $\sigma = 1$ and about 50% of the swarm for $\sigma = 2$), above which individuals are able to aggregate in equal numbers on both sites. Therefore, informed agents are able to guide self-organised aggregation only above a critical proportion of informed individuals, which increases with increasing σ , which therefore suggest that larger aggregation sites have again a counter-intuitive negative effect on the controllability of this self-organised behaviour.

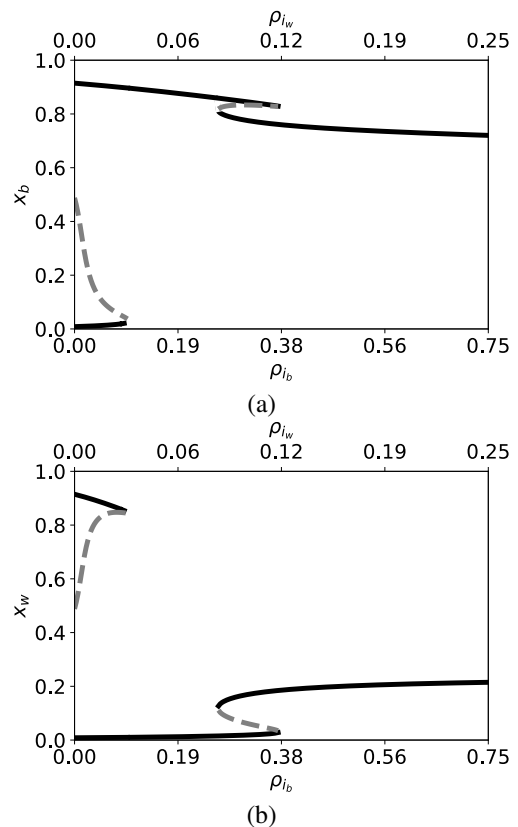


Figure 3: (a) Fraction of individuals, when $\frac{\rho_{i_b}}{\rho_{i_w}} = 3$ and $\sigma = 1$ for (a) the black site and (b) the white site. Black continuous lines: stable solutions. Dashed grey lines: unstable solutions.

In Figure 3, the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$ is set to 3, that is for each proportion of informed individuals in the swarm, 75% of them are of type i_b and 25% are of type i_w , and $\sigma = 1$. The graphs in Figure 3a and 3b can be globally understood as follows: below a given threshold of about 10% of informed individuals of type i_b , one of two things can happen: either informed individuals of type i_b and non-informed individuals aggregate on the black site, and informed individuals of type i_w do not aggregate on any site; or informed individuals of type i_w and non-informed individuals aggregate on the white site, and informed individuals of type i_b do not aggregate on any site. Under this condition, the behaviour of informed individuals that do not aggregate on their preferred site can be explained by observing that the dimension of the site is too large relative to their number to trigger any aggregation process. Beyond 30% of informed individuals of type i_b , 75% of the swarm aggregates on the black site and 25% of the swarm aggregate on the white site. This is the regime where informed agents are able to guide self-organised dynamics.

Mathematically, when the total proportion of informed individuals in the swarm is low, the following approximations hold:

$$\left\{ \begin{array}{l} x_b^{i_b} \approx \rho_{i_b} \\ x_b^{n_i} \approx 1 - \rho_{i_b} - \rho_{i_w} \\ x_w^{i_w} \approx 0 \\ x_w^{n_i} \approx 0 \end{array} \right. \text{ OR } \left\{ \begin{array}{l} x_b^{i_b} \approx 0 \\ x_b^{n_i} \approx 0 \\ x_w^{i_w} \approx \rho_{i_w} \\ x_w^{n_i} \approx 1 - \rho_{i_b} - \rho_{i_w} \end{array} \right. \quad (10)$$

In such a case, the swarm aggregates only on one site, with the informed individuals that prefer the other site do not join the aggregate and they do not aggregate on their preferred site. When the total proportion of informed individuals in the swarm is high, the following approximations hold:

$$\left\{ \begin{array}{l} x_b^{i_b} \approx \rho_{i_b} \\ x_b^{n_i} \approx R_b - \rho_{i_b} \\ x_w^{i_w} \approx \rho_{i_w} \\ x_w^{n_i} \approx R_w - \rho_{i_w} \end{array} \right. \quad (11)$$

where R_b and R_w is the ratio of informed individuals of type i_b and i_w over the total number of informed individuals, respectively. These results are valid only when the critical value of σ_{max} is not reached for the specific values of R_b and R_w , as discussed previously.

Figure 4 reports results of an analysis similar to the one reported in Figure 3 but with $\sigma = 2$ instead than $\sigma = 1$, with the ratio $\frac{\rho_{i_b}}{\rho_{i_w}}$ still set to 3. The graphs in Figure 4a and 4b show that below a given threshold of about 20% of informed individuals of type i_b , the same behaviour is observed as in Figure 3a and 3b: informed individuals of type i_b and non-informed individuals aggregate on the black site, and informed individuals of type i_w do not aggregate on any site; or informed individuals of type i_w and non-informed individuals aggregate on the white site, and informed individuals of type i_b do not aggregate on any site. Beyond 20%

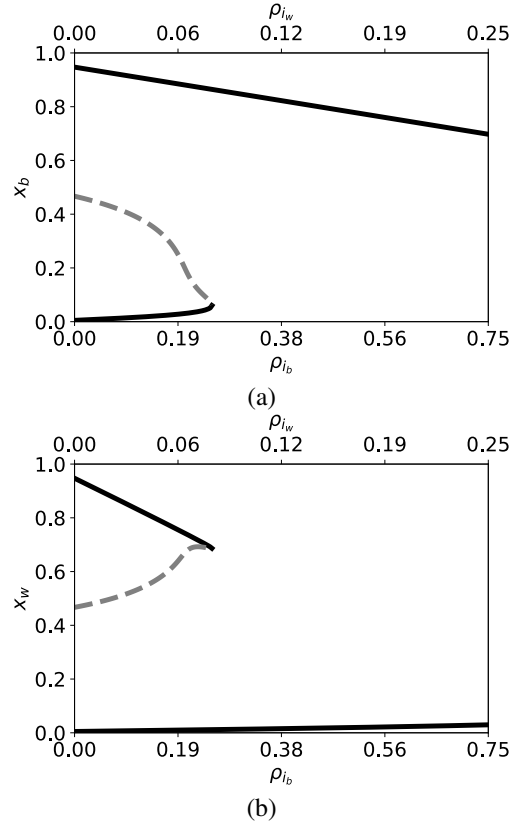


Figure 4: (a) Fraction of individuals, when $\frac{\rho_{i_b}}{\rho_{i_w}} = 3$ and $\sigma = 2$ for (a) the black site and (b) the white site. Black continuous lines: stable solutions. Dashed grey lines: unstable solutions.

of informed individuals of type i_b , 75% of the swarm aggregates on the black site but no individuals aggregate on the white site, since the dimension of the site is too large to trigger any aggregation process. In other words, with $\sigma = 2$, informed agents are never able to guide the aggregation dynamics. Indeed, when $\frac{\rho_{i_b}}{\rho_{i_w}} = 3$, in order to induce the individuals of type i_w to aggregate on their white site we need $\sigma < \sigma_{max} \approx 1.6$.

To complete the discussion, analysis has been performed for values of $\sigma < 1$. As shown in Figure 5, when the fraction of informed individuals is low, the swarm behaves as predicted by (Amé et al., 2006). That is, individuals distribute equally among the two sites. The distribution of individuals then changes continuously up to the desired distribution when all the individuals are informed. Note that when the site carrying capacity is not large enough to contain the corresponding informed individuals, the amount of individuals on the site is limited by this capacity and therefore never reaches the desired fraction of individuals.

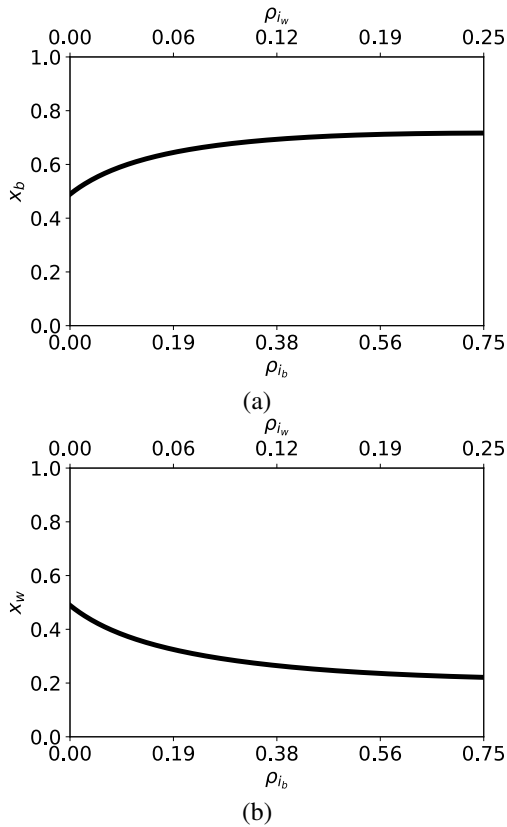


Figure 5: (a) Fraction of individuals, when $\frac{\rho_{i_b}}{\rho_{i_w}} = 3$ and $\sigma = 0.9$ for (a) the black site and (b) the white site. Black continuous lines: stable solutions.

Conclusions

In this paper, we introduced a mathematical ordinary differential equations model that is inspired by the one proposed by Amé et al. (2006). We performed an analytical study of self-organised aggregation in presence of two distinctive aggregation sites, one black and one white. We consider a swarm of agents characterised by the presence of informed individuals, that is agents that are able to recognise the colour and therefore discriminate between the two sites. Our model considers sub-populations of informed individuals, distinguishing between those that prefer the white and those that prefer the black site. Each type of informed individuals never rests on the non-preferred site. From an engineering perspective, when designing self-organised systems engaged in aggregation tasks, we would like to use informed individuals to guide the self-organised aggregation dynamics. In particular, we would like to correlate the relative proportion of one type of informed individuals with respect to the other type, with the total proportion of individuals aggregated in each site.

We analysed the equilibria of the model with respect to the site carrying capacity and to the proportion of informed

individuals that prefer the white or the black site. Results show that, as in Amé et al. (2006), dynamics are strongly dependent on the environmental conditions. For intermediate values of the site carrying capacity, the informed individuals are able to guide the dynamics. And within this range, the critical mass of informed individuals needed to guide the dynamics is positively correlated with the site carrying capacity, meaning that larger sites make the collective dynamics more difficult to be guided by informed individuals. Finally, to perform a non-even allocation among the two sites, the range of the carrying capacity parameter that allows informed individuals to guide these dynamics is even more narrow compared to the case of even allocation.

This paper has based its analysis on a seminal and important model of self-organised aggregation which was derived after experiments performed with real cockroaches (see Amé et al., 2006). However, experimental results we performed in (Firat et al., 2019) have already given us insight that, by having more control on the microscopic self-organised model of aggregation, it is possible to have informed individuals guiding the dynamics in a wider range of environmental conditions. In future work, we would like to focus our efforts in two directions. First, we would like to propose a macroscopic ODE model that more closely capture the microscopic design method discussed in (Firat et al., 2019) rather than the behaviour of natural cockroaches. Both in (see Amé et al., 2006) and in (Firat et al., 2019), the individual probability of leaving a site is a non-linear function of the density of individuals at a site. However, we believe that the specific non-linear dependency can be tuned in a way to make dynamics less dependent on environmental conditions when informed individuals are introduced. Secondly, the almost totality of self-organised models of aggregation in swarm of agents are engineered in order to amplify the effect of positive feedback, which is the prime mechanism responsible for aggregation. This is because the focus of all these studies is in achieving a single aggregate. Our approach to the problem differs from previous research works since we aim to use informed individuals to distribute the swarm on two or more aggregation sites (rather than concentrate it on a single site) according to the relative proportions of different types of informed individuals present in the swarm. Building upon the encouraging results of this macroscopic model, we are currently working at a new microscopic model that is able to regulate the effect of positive feedback in a way that is different between informed and non-informed individuals. This work aims to design individual controllers for swarm of robots that allow the designer to chose between inducing the swarm to aggregation on one site, and allocation of individuals to different sites in proportion to the relative frequency of informed robots present in the swarm.

References

- Amé, J., Rivault, C., and Deneubourg, J. (2004). Cockroach aggregation based on strain odour recognition. *Animal Behaviour*.
- Amé, J.-M., Halloy, J., Rivault, C., Detrain, C., and Deneubourg, J. L. (2006). Collegial decision making based on social amplification leads to optimal group formation. *Proceedings of the National Academy of Sciences*, 103(15):5835–5840.
- Bayindir, L. and Şahin, E. (2009). Modeling self-organized aggregation in swarm robotic systems. In *IEEE Swarm Intelligence Symposium, SIS'09*, pages 88–95. IEEE.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ.
- Campo, A., Garnier, S., Dédriche, O., Zekkri, M., and Dorigo, M. (2010). Self-organized discrimination of resources. *PLoS ONE*, 6(5):e19888.
- Çelikkanat, H. and Şahin, E. (2010). Steering self-organized robot flocks through externally guided individuals. *Neural Computing and Applications*, 19(6):849–865.
- Correll, N. and Martinoli, A. (2011). Modeling and designing self-organized aggregation in a swarm of miniature robots. *The International Journal of Robotics Research*, 30(5):615–626.
- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516.
- Dorigo, M. and Şahin, E. (2004). Guest editorial. Special issue: Swarm robotics. *Aut. Rob.*, 17(2–3):111–113.
- Ferrante, E. (2013). Information transfer in a flocking robot swarm. Ph.D thesis - Université Libre de Bruxelles.
- Ferrante, E., Turgut, A., Huepe, C., Stranieri, A., Pinciroli, C., and Dorigo, M. (2012). Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adaptive Behavior*, 20(6):460–477.
- Ferrante, E., Turgut, A., Stranieri, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2014). A self-adaptive communication strategy for flocking in stationary and non-stationary environments. *Natural Computing*, 13(2):225–245.
- Firat, Z., Ferrante, E., Cambier, N., and Tuci, E. (2018). Self-organised aggregation in swarms of robots with informed robots. In Fagan, D., Martín-Vide, C., M. O’Neill, and M.A. Vega-Rodríguez, editors, *Theory and Practice of Natural Computing*, pages 49–60. Springer.
- Firat, Z., Ferrante, E., Gillet, Y., and Tuci, E. (2019). On self-organised aggregation dynamics in swarms of robots with informed robots. Available at <http://arxiv.org/abs/1903.03841>.
- Garnier, S., Gautrais, J., Asadpour, M., Jost, C., and Theraulaz, G. (2009). Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behavior*, 17(2):109–133.
- Garnier, S., Jost, C., Gautrais, J., Asadpour, M., Caprari, G., Jeanson, R., Grimal, A., and Theraulaz, G. (2008). The embodiment of cockroach aggregation behavior in a group of micro-robots. *Artificial life*, 14(4):387–408.
- Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., and Theraulaz, G. (2005). Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. In *European Conference on Artificial Life*, pages 169–178. Springer.
- Gauci, M., Chen, J., Li, W., Dodd, T., and Groß, R. (2014). Self-organized aggregation without computation. *The International Journal of Robotics Research*, 33(8):1145–1161.
- Halloy, J., Sempo, G., Rivault, C., Asadpour, M., Tâche, F., Saïd, I., Durier, V., Canonge, S., Amé, J., Detrain, C., Correll, N., Martinoli, A., Mondada, F., Siegwart, R., and Deneubourg, J. (2007). Social integration of robots into groups of cockroaches to control self-organised choices. *Science*, 318(5853):1155–1158.
- Hamann, H. and Wörn, H. (2008). A framework of space–time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2):209–239.
- Harvey, I., Di Paolo, E., Wood, R., Quinn, M., and Tuci, E. (2005). Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11(1-2):79–98.
- Hsieh, M. A., Halász, Á., Berman, S., and Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2):121–141.
- Krause, J. and Ruxton, G. (2011). The dynamics of collective human behaviour. *Lancet*, 377(9769):903–904.
- Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Auton. Robots*, 13(2):127–141.
- Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3):305–327.
- Soysal, O. and Şahin, E. (2007). A macroscopic model for self-organized aggregation in swarm robotic systems. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics*, pages 27–42. Berlin, Heidelberg. Springer Berlin Heidelberg.
- Stroeymeyt, N., Franks, N. R., and Giurfa, M. (2011). Knowledgeable individuals lead collective decisions in ants. *Journal of Experimental Biology*, 214(18):3046–3054.
- Tuci, E., Alkilabi, M., and Akanyety, O. (2018). Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers in Robotics and AI*, 5:1–15.
- Tuci, E. and Rabérin, A. (2015). On the design of generalist strategies for swarms of simulated robots engaged in a task-allocation scenario. *Swarm Intelligence*, 9(4):267–290.
- Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2015). Collective decision with 100 kilobots: speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multi-Agent Systems*, pages 1–28.

Emergent Escape-based Flocking Behavior using Multi-Agent Reinforcement Learning

Carsten Hahn¹, Thomy Phan¹, Thomas Gabor¹, Lenz Belzner² and Claudia Linnhoff-Popien¹

¹Mobile and Distributed Systems Group, LMU Munich, Munich, Germany

²MaibornWolff, Munich, Germany
carsten.hahn@ifi.lmu.de

Abstract

In nature, flocking or swarm behavior is observed in many species as it has beneficial properties like reducing the probability of being caught by a predator. In this paper, we propose SELFish (Swarm Emergent Learning Fish), an approach with multiple autonomous agents which can freely move in a continuous space with the objective to avoid being caught by a present predator. The predator has the property that it might get distracted by multiple possible preys in its vicinity. We show that this property in interaction with self-interested agents which are trained with reinforcement learning solely to survive as long as possible leads to flocking behavior similar to Boids, a common simulation for flocking behavior. Furthermore we present interesting insights into the swarming behavior and into the process of agents being caught in our modeled environment.

Introduction

Flocking or swarm behavior is observed in many species in nature. A prominent example is fish schooling, where multiple fishes do not only stay close to each other for social reasons but coordinate their actions collectively. That means that an individual fish aligns its direction in regard to fishes that are close to it, while maintaining a certain cohesion of the group and still avoiding collisions with other individuals.

However, flocking behavior does not only exist as an end in itself. In nature, a schooling fish benefits from schooling in multiple ways: The swarm increases its hydrodynamic efficiency or mating chances. Also, flocking enhances foraging success as collaborative observation is superior to a single individual's. The same is true for predator detection. Even further, the probability of being caught decreases for an individual with regard to certain predator behaviors.

Reynolds (1987) showed that algorithmically implementing the three rules of alignment, cohesion and separation leads to flocking behavior while an individual only needs local knowledge about its surrounding neighbors (called Boids). In order to overcome these static flocking rules Morihiro et al. (2008) used reinforcement learning to train an individual to justify the rules stated above in order to form a swarm. This was done by shaping the reward sig-

nal according to distances between the individuals and limiting their actions to be attracted to another fish, be repulsed from another fish and move parallel in the same or opposite direction of another fish, respectively.

With SELFish we investigate the case that an individual tries to optimize its behavior w.r.t. the objective of surviving as long as possible in the presence of a predator (which might get distracted by multiple preys). We show that this simple objective leads to emergent flocking behavior (similar to Boids) in a multi-agent reinforcement learning setting, without the need to explicitly enforce it. Note that we use reinforcement learning merely as a tool to investigate the optimality of behavior, not analogous to nature (where swarms usually emerge genetically not due to individual training).

Reinforcement Learning

Reinforcement Learning denotes a machine learning paradigm in which an agent interacts with its environment and receives a certain reward for its action accompanied with an observation of the new state of the environment. Such scenarios are usually modeled as Markov Decision Processes (MDPs), where \mathcal{S} denotes the set of states of the environment, \mathcal{A} denotes the set of actions an agent can take and $r(s_t, a_t)$ is the intermediate reward received after action a_t was taken in state s_t at time step t . Also, the process moves to a new state s_{t+1} influenced by the action a_t , with the Markov property being that the new probability of transitioning into state s_{t+1} only depends on state s_t and the chosen action a_t : $\mathcal{P}(s_{t+1}|s_t, a_t)$. The goal is to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ which maximizes the accumulated reward $R_t = \sum_{i=t}^T \gamma^{i-t} r(s_i, a_i)$ from time step t to the simulation horizon T with a discounting factor $\gamma \in [0, 1]$.

In SELFish the state is partially observable, which means that instead of using the full state description s_t to determine the action $a_t = \pi(s_t)$, the agent only uses an observation $o_t \in \mathcal{O}$ (where \mathcal{O} is the space of all possible observations) as input to a policy function $\pi : \mathcal{O} \rightarrow \mathcal{A}$ to compute the action $a_t = \pi(o_t)$. Furthermore the observation may be different for every agent. However, we focus on a deterministic domain, so $\mathcal{P}(s_{t+1}|s_t, a_t) \in \{0, 1\}$.

Deep Learning

In Reinforcement Learning the policy or intermediate functions, which help to derive it, are usually expressed as deep artificial neural networks. Neural networks can be viewed as a directed graph of nodes, called neurons, which are interconnected by weighted edges. A neuron receives inputs over its ingoing edges, usually computes the weighted sum of the inputs, applies a non-linear function to this weighted sum and forwards its output to subsequent neurons via its outgoing edges. The neurons are usually arranged in layers, where layers between the input layer and the output layer of the network are referred to as hidden layers. Networks with multiple hidden layers are called deep neural networks.

Artificial neural networks serve as biologically inspired function approximators which can be trained by example to approximate a function f mapping an input vector $x \in \mathbb{R}^n$ to an output vector $y \in \mathbb{R}^m$ depending on the weights of the edges θ . The goal in training a neural network is to minimize the error between the networks' output $y' = f(x; \theta)$ and the known desired (example) output y by adjusting the weights θ accordingly. This can be done with the Backpropagation method combined with a gradient descent strategy.

Deep Q-Learning (DQN)

Q-Learning is a value-based approach named after the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, which describes the expected accumulated reward $Q^\pi(s_t, a_t)$ after taking action a_t in state s_t and following the policy π in all subsequent states. The goal is to find an optimal action-value function Q^* , which yields the highest accumulated reward. Q^* can be approximated through Bellman's principle based on the intuition that for an optimal policy, independently of the initial state and initial decision, all remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman (1957)). Starting from an initial guess for Q , it can be iteratively updated via

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

where the learning rate $\alpha \in (0, 1)$ is a parameter to be specified. The learned action-value function Q converges to Q^* , from which an optimal policy can be derived via $\pi^*(s_t) = \arg \max_a Q(s_t, a)$.

In Deep Q-Learning (DQN) (Mnih et al. (2013)) an artificial neural network is used to represent the action-value function Q . Also, to minimize correlations between samples and to alleviate non-stationary distributions an experience replay mechanism is used (Mnih et al. (2013)) which randomly samples previous state action transitions to train the neural network.

Deep Deterministic Policy Gradient (DDPG)

To overcome the limitation of Q-Learning, which cannot directly be applied to continuous action spaces, efforts were

made to learn the policy $\mu(s|\theta^\mu)$ directly with a parameterized objective function $J(\theta)$ (Silver et al. (2014); Lillicrap et al. (2015)). In addition it was proposed to split the learning process in two components to reduce the gradient variance, called actor-critic approach. The critic learns the action-value function $Q(s, a)$ using the Bellman equation as in Q-learning. The actor then updates the policy parameters θ^μ in the direction suggested by the critic:

$$\nabla_{\theta^\mu} J = \mathbb{E}_{s_t} [\nabla_a Q(s, a | \theta^Q) |_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s | \theta^\mu) |_{s=s_t}]$$

Multi-Agent Case

Many approaches have been suggested for the case that there are multiple agents present which are either self-interested or have to work together to achieve a cooperative goal. A straightforward idea in the case that there are multiple agents that act in their self-interest, which means that they only maximize their own accumulated reward, is deploying a standard reinforcement learning algorithm (as in the single-agent case) in each individual agent in the multi-agent setting and let all agents learn simultaneously. This straightforward approach bears the problem of non-stationarity in the state transitions. As one agent tries to adapt its actions in certain states, other agents, which are considered as part of the environment for the first agent, do so as well. This makes it difficult to learn a policy depending on the observed state, which no longer satisfies the Markov property.

Egorov (2016) approaches a pursuit-evasion game with reinforcement learning. There are multiple pursuers and multiple evaders. Only one agent of each kind is trained through Q-Learning at a time while the policies of the other agents are fixed. After a number of iterations the policy of the learning agent is distributed to all other agents of the same type. Through this process the policy of one set of agents is improved incrementally over time.

This mitigates the problem of non-stationarity. Furthermore it seems reasonable to copy the policy of one agent throughout multiple homogenous agents as all are alike and pursue the same self-interested goal. This observation is also relevant for flocking or swarm behavior of multiple agents as we will demonstrate below.

Swarm Behavior

In 1987, Craig Reynolds (Reynolds (1987)) described three basic rules through which flocking behavior can be modeled. For these rules an individual only needs local knowledge about its neighbors within a certain distance. These rules are:

- **Alignment:** Steer towards the average heading direction of local flockmates
- **Cohesion:** Steer towards the average position (center of mass) of local flockmates
- **Separation:** Steer to avoid crowding local flockmates

If each individual (called *Boids* by Reynolds as he thought of bird-like creatures) follows these rules, a swarm formation emerges. In an implementation, they can be expressed as physical forces which act upon an individual. Supplementary forces can be introduced, which repel an individual from an enemy or from obstacles, for example.

To overcome these static rules definitions, Morihiro et al. (2008) used Reinforcement Learning, particularly Q-Learning, to train agents to follow these rules. In their model the agents iteratively learn while at every time step an agent i only considers one other agent j . Agent i receives the euclidean distance to j as observation and can choose among four actions to execute. These actions are to move towards agent j , away from agent j or parallel to agent j either in the same or opposite direction. The reward agent i receives for an action depends on the previously mentioned distance to agent j and is shaped in a way that it intuitively represents the cohesion and separation rule. In this regard agent i receives a positive reward if it steers so to keep its distance to j within predefined boundaries.

While the previously mentioned approaches lead to flocking behavior, they neglect the beneficial properties flocking behavior might have for the individuals. One of those benefits could be the increased likelihood to survive in the presence of predators, as they might get distracted by the sheer amount of possible targets. The question arises whether flocking behavior occurs in a scenario with such properties where agents solely try to maximize their survival time. In contrast to Morihiro et al. (2008), we pursue a scenario in which agents are trained with reinforcement learning solely on the objective to survive, without explicitly enforcing swarm behavior. Additionally, we demonstrate that SELFish also works for a continuous action space of the agents.

Emergent Swarm Behavior

In order to investigate whether the objective to survive in the presence of a predator would lead to flocking behavior in a multi-agent setting, we created a model that facilitates such a behavior. In the following the properties of the environment will be explained. This is followed by a description of the action and observation space as well as the reward structure which was used to train the agents.

Environment

The agents, which are the prey in this scenario, can freely move in a continuous two-dimensional space, visualized as a square with predefined edge lengths (see Figure 1). An agent itself is represented as a circle with a surface substantially smaller than the space it is moving in. There are neither obstacles nor walls in the environment. Furthermore agents do not collide with each other. To ease free roaming of the agents, the space has the special characteristic that it wraps around at the edges forming a torus. That means that if an agent leaves the square visualization to the right, it



Figure 1: Example of the space with 60 agents (green) and one predator (orange).

will immediately enter it again from the left (same for the other direction or around top and bottom).

Together with the agents there also exists a predator in the environment. The predator is also represented as a circle. The goal of the predator is to catch the agents by moving to their position. As soon as the predator collides with an agent, the execution of the concerning agent will end and a new agent is spawned immediately at a random position to keep the number of agents in the system constant. If there are multiple agents within a certain distance around the predator, it will choose one for a target at random (otherwise it will move to the closest agent's direction). This means that the predator can be distracted by multiple agents in its proximity. Thus it might be beneficial for an agent to move towards other agents as the predator might get distracted, which is essential for flocking behavior. However, to prevent the predator from constantly changing targets it will follow a chosen target for a certain time before a new target will be chosen. By default, the agents and the predator move at the same speed. This would allow an agent to turn in the opposite direction of the predator and move away without the predator having a chance to catch up. That is why the predator will accelerate occasionally for a short amount of time, which simulates a leap forward to catch the prey it is following. The policy of the predator is static and does not change over time.

Objective of an Agent

The goal of the agents is not to collide with the predator. For this they receive a reward of **+1** for each step/frame they live and **-1000** for the collision with the predator which ends their life. With this reward structure the objective of the agents can be viewed as “surviving as long as possible”. As there are no obstacles in the environment and the agents do not collide with each other, there are no other rewards/penalties.

Action Space

The action space of the agents only comprises of the angle they want to turn each time step. The movement speed of the agents is constant and cannot be altered by them for now.

The action a , which represents the turning angle that can be chosen from discrete steps or out of a continuous interval by the agent, depends on the reinforcement learning strategy which is used later on. In the case that DQN is used, the actions an agent can choose from comprise five discrete degree values $\{-90^\circ, -45^\circ, 0^\circ, +45^\circ, +90^\circ\}$. The agent can choose any real-valued degree as turning angle in the case of DDPG.

As a side node, the predator can only take limited real-valued turns $\{x \in \mathbb{R} \mid -45^\circ \leq x \leq 45^\circ\}$ at every step with the goal to give the agents a higher maneuverability than the predator.

Observation Space

In order to facilitate the scalability to many autonomous agents, one agent cannot observe the full state of the environment; instead its observation is limited to itself, the predator and the n nearest neighboring agents. This approach can be explained biologically, where, for example, a fish in a swarm cannot observe the whole swarm but only its local neighbors. But it is also in line with related work, for example Boids, where also only local neighborhoods between agents are regarded. Furthermore it eases computation and has the nice property that the observation vector, which is forwarded through the reinforcement learning algorithm in order to obtain an action, has a constant length (cf. the following section).

For every observable entity e , the agent receives a 3-tuple which contains the euclidean distance between the entity and the agent, the angle the agent would have to turn to face towards the observed entity and the absolute orientation of the entity in the environment: $(dist_e, direction_e, orientation_e)$. As the environment is a torus, the distances are also calculated around the edges of the visualized square, with the shorter distance being taken (with the $direction_e$ corresponding to this). The absolute orientation of an entity is measured in degrees $[0^\circ, 360^\circ)$, where facing east corresponds to 0° , measuring the angle counter-clockwise. The angle an agent would have to turn to face towards another entity is measured in degrees in the range of $(-180^\circ, 180^\circ]$.

Accordingly, an agent receives the following observation for the predator, itself and the n nearest neighboring agents, in which the n neighbors are ordered by their distance.

$$\begin{bmatrix} dist_{predator} & direction_{predator} & orientation_{predator} \\ 0 & 0 & orientation_{self} \\ dist_{neighbor_1} & direction_{neighbor_1} & orientation_{neighbor_1} \\ dist_{neighbor_2} & direction_{neighbor_2} & orientation_{neighbor_2} \\ \vdots & \vdots & \vdots \\ dist_{neighbor_n} & direction_{neighbor_n} & orientation_{neighbor_n} \end{bmatrix}$$

Hyperparameter	DQN	DDPG
Training Steps	500,000	500,000
Hidden Layer	10	5
Neurons in Layers	16	Actor: 16 Critic: 32
Hidden Layer Activation	relu	relu
Last Layer Activation	linear	linear
γ	0.999999	0.999999
Optimizer	Adam	Adam
Learning Rate	0.001	0.001
Replay Buffer Size	50,000	100,000
Batch Size	64	512
Exploration	ϵ -Greedy $\epsilon = 0.1$	Ornstein Uhlenbeck $\theta = 0.15,$ $\mu = 0.0,$ $\sigma = 0.3$
Observable neighboring agents	5	1

Table 1: Hyperparameters for Reinforcement Learning

Training

As mentioned before, a valid way for training multiple homogeneous agents through reinforcement learning is to train only one instance and then to copy the learned policy to all instances of the homogeneous group (Egorov (2016)). This also resembles nature, where for example multiple schooling fish follow the same behavioral policy.

For this purpose, the DQN and DDPG implementations of Keras-RL (cf. Plappert (2016)) were used. Keras-RL is originally developed for OpenAI Gym Environments (Plappert (2016)), in which only single agents interact with these environments through a $step(action)$ -method, which is given an action and returns an observation, a reward and a done flag, indicating whether the current episode is finished. This interface was also used in the proposed swarm environment to train a single agent to avoid the present predator with the previously mentioned rewards, action and observation spaces. During the training of one agent, the other agents are present as well, onto which the policy (i.e. the neural network) of the learning agent is copied after each episode. An episode ends if the learning agent is caught by the predator or 10,000 steps (frames) were executed.

During training, the edge lengths of the space were 40×40 pixels, although it wraps around at the edges. Please note that the agents and the predator could be positioned at any real value in the interval $[0, 40]$. However, the values in the 3-tuples of the observation were normalized to $[0, 1]$ anyway. The agents and the predator were represented by circles of radius 1, with an agent being caught if the distance of its position and the position of the predator is below 2. Also, during training only 10 agents were present.

In order to find a good configuration for the parameters

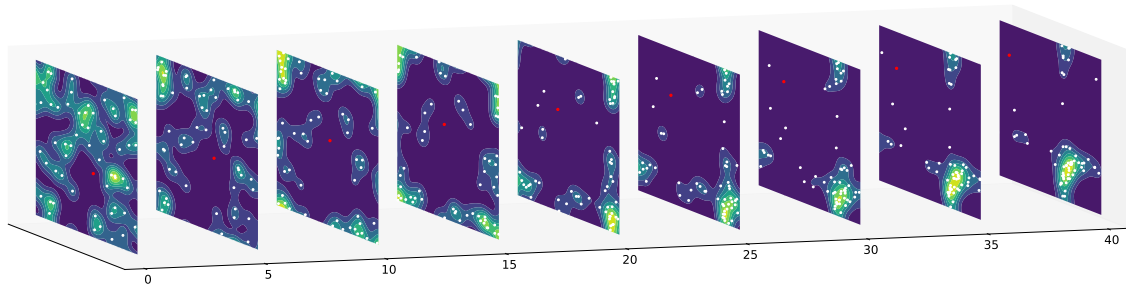


Figure 2: Swarm formation in the first 40 frames of an episode of $SELFish_{DQN}$. Agents (white) and predator (red) were randomly initialized. Kernel Density Estimation Phillips et al. (2006) was used to highlight the dense regions of the multi-agent swarm. Note that the space wraps around the edges.

of the reinforcement learning algorithms, many runs were executed. The quality of the parameter configuration of the training run was assessed during a test phase based on the cumulative reward the learning agent could acquire, which essentially equals the number of time steps it could survive. The number of neighboring agents that could be observed was also varied as parameter. See Table 1 for the best parameters found.

Even for the small number of agents which were present during the training, a swarming behavior could be observed when the learned behavior of one agent was transferred to the others. Since the observation of an agent is partial and thus limited to the 3-tuple $(dist_{neighbor_i}, direction_{neighbor_i}, orientation_{neighbor_i})$ for the n nearest neighbors, the number of agents as well as the size of the space can be increased without breaking the learned policy. With this even better swarming behavior can be observed, which shall be further evaluated in the next section.

Simulations and Results

First we want to give an impression of the swarms that are forming from reinforcement learning. See Figure 2 for the formation of a swarm in the first 40 frames of a test episode of $SELFish_{DQN}$. With a continuous action space, $SELFish_{DDPG}$, exhibits similar behavior although the swarm tends to be more dense. The swarm presumably forms because one agent learns that the predator might get distracted from it if it stays close to other agents which prolongs its life and thereby its accumulated reward.

Boids enforces the alignment, cohesion and separation of neighboring agents. This can be expressed by vector calculations together with weights which set these three rules in context. To make the scenario more similar to the reinforcement learning setting, another force which pushes the Boids away from the predator was added (altogether with a weight for this behavior which sets it in context to the other rules). To find a good configuration for the alignment, cohesion, separation and predator avoidance weight, multiple

runs with different parameter setting were executed. Again, the quality of a setting was evaluated based on the number of time steps a certain boid could survive.

If it is only about the survival of an agent, a simple strategy one could think of is to simply turn in the opposite direction of the predator and to move away from it regardless of the surrounding agents. This policy will be called *TurnAway* in the following and will be given for comparison¹.

Alignment and Cohesion

As Boids enforces the alignment and the cohesion of the agents, we want to compare the swarms resulting from predator avoidance through reinforcement learning to Boids by these means. As the orientation of an agent is measured as angle in $[0^\circ, 360^\circ)$ (facing east corresponds to 0°), the alignment of the agents can be measured as deviation from a mean angle of a group (see Figure 3). The absolute deviation of each agent from this mean angle was summed and averaged over the number of agents. To measure the cohesion of the swarm, the average distance between the agents was calculated. For this the distance between all agents i and j was summed and averaged by the number of pairs of agents.

Considering that the agents flee from a predator and the space wraps around at the edges, multiple flocks with different orientations, depending on their position in regard to the predator, might form, as it is already evident from the Figures 1 and 2. That is why it did not seem sensible to calculate alignment and cohesion over all agents in the space. To counter this, the density-based clustering method DBSCAN (Ester et al. (1996)) and particularly its scikit-learn implementation (Pedregosa et al. (2011)) was used beforehand and the average deviation from the mean angle and the average distance between two agents was only calculated for agents in a specific cluster (see Figure 4 for an example). The measurements over all agents are given for comparison.

¹For a short video showing all implemented policies please refer to <https://youtu.be/SY59CYaqWpE>

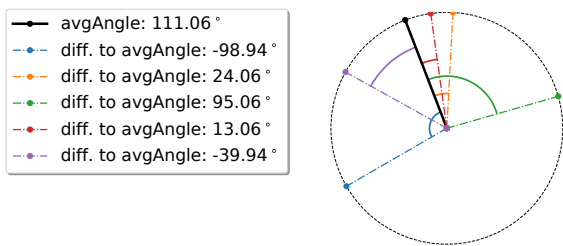


Figure 3: Considering the orientation of five agents in space, a mean angle (black) and the deviation from this in $[-180^\circ, 180^\circ]$ can be computed (Watson (1983)).

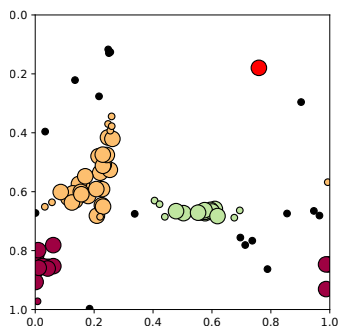


Figure 4: Example Clustering for SELFish_{DQN} with 40 agents (predator as red dot).

Figure 5 shows the number of agents in a specific cluster, when 40 agents were present in a space of 40×40 pixels. It is visible that the TurnAway strategy produces many noise points on average. The clusters that are found for TurnAway are mostly due to the agents moving in the same direction to avoid the predator and also overlapping when wrapping around the edges of the space. Boids and the two reinforcement learning approaches used in SELFish, DQN and DDPG, produce rather similar cluster numbers and sizes on average, with DDPG having a tendency to form one large cluster.

By looking at the average deviation from the mean orientation angle of the agents inside clusters (see Figure 6)

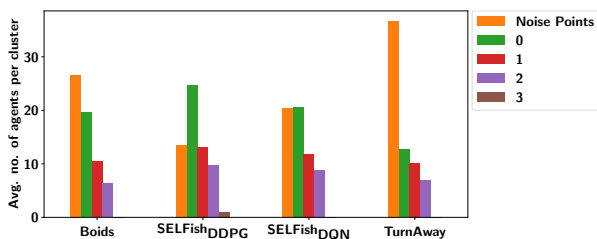


Figure 5: Average number of agents in a respective cluster (cluster ID given) with noise points being agents that could not be assigned to a specific cluster.

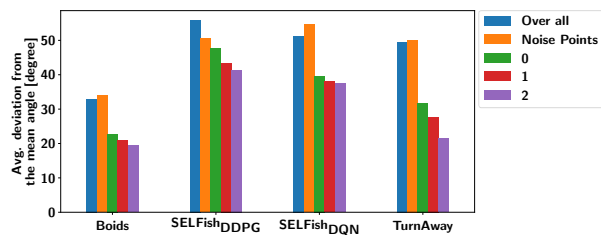


Figure 6: Average deviation from the mean orientation angle of the agents over clusters.

one can see that Boids produces the most aligned groups of agents which generally move in the same direction. SELFish_{DQN} and SELFish_{DDPG} are deviating more, presumably because agents following these policies tend to kind of quiver. Also these agents show the behavior of creating a line at the point at which they would again move towards the predator because of the torus environment. At these lines the agents circulate until the predator moves into their direction. For TurnAway only groups of agents moving in the same direction are detected anyway, with the average angle deviation being distorted by agents coming from the other side of the space and moving in the opposite direction. One might question whether the swarms (respectively clusters) found for SELFish_{DQN} or SELFish_{DDPG} also solely result from the fact that the agents learned to turn away from the predator and thereby move in the same direction. This can be countered by the observation that if the predator is pinned down at a fixed position (it cannot be removed completely as it is part of the agents' observation), the learning agents still form a swarm at the greatest possible distance from the predator where they circulate around each other. Figure 7 shows the average pairwise distance between agents either inside clusters, between noise points or between all agents, which is homogeneous over all four agent policies, with only SELFish_{DDPG} tending to produce somewhat denser agent groups. The homogeneity between the behavioral strategies with regard to the average pairwise distance also results from the DBSCAN clustering.

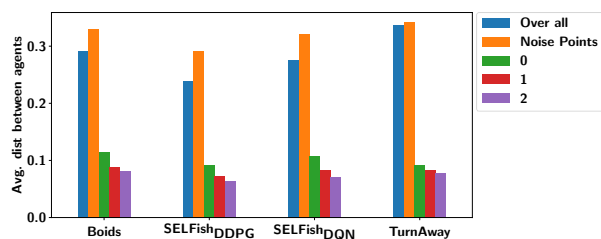


Figure 7: Average pairwise distance between agents either inside clusters, between noise points or over all. Edge lengths of the space normalized to 1 for distance calculation.

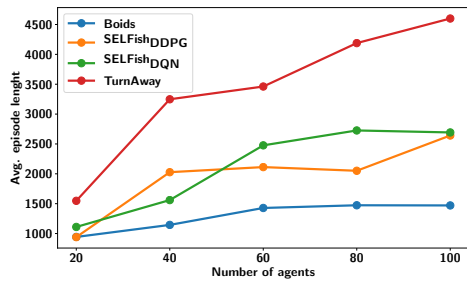


Figure 8: Average episode length for each of the behavior strategies with varying number of agents in the environment.

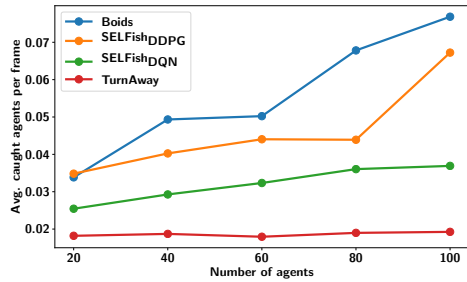


Figure 9: Number of caught agents divided by the time it took with varying number of agents in the environment.

Agent Survival

For the reinforcement learning algorithms the reward was defined such that the single learning agent received +1 for every step and -1000 for being caught. The maximization of the accumulated reward should encourage it to stay alive as long as possible. After the end of an episode, which ended when the learning agent was caught or 10,000 steps passed, the learned policy was copied to all other agents. Figure 8 shows the mean episode length for the different policies, which essentially corresponds to the mean accumulated reward of the learning agents. For the static policies, Boids and TurnAway, it corresponds to the time it took until a certain agent was caught. Note that although the number of agents in the environment is varied, the parameter for Boids or the policies for SELFish_{DQN/DDPG} are still those that were determined in smaller settings with only 10 agents.

It turns out that when evaluating the actual survival rate of every single agent, the best strategy to survive is to simply turn away from the predator. This is also true considering the whole swarm, i.e. all agents. In Figure 9, the absolute number of caught agents in an episode was divided by the length of the episode (reduced by a transient phase of 100 frames for swarm formation). These measurements were then again averaged over multiple episodes and runs (with different seeds).

This raises the question why this behavior was not found by the reinforcement learning algorithms. The outcome of

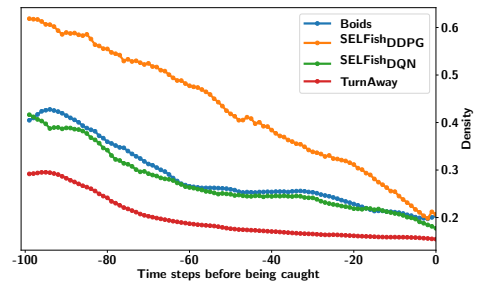


Figure 10: Density of an agent in accordance to the Kernel Density Estimation in the last 100 time steps before it is caught (mean for multiple agents).

the reinforcement learning could potentially be explained considering the Prisoner’s Dilemma (Poundstone (1992)). In this game-theoretical example, prisoners *A* and *B* are kept in arrest without means to communicate. Simultaneously, both are given the opportunity either to betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent with the respective pay-offs shown in Table 2. The only Nash equilibrium (Nash (1951)) is that both prisoners defect as this yields less charge for each of them than if one stays silent while the other prisoner keeps its strategy unchanged and testifies that the other committed the crime (betrays). The dilemma is that mutual cooperation yields a better outcome although it is not rational from a self-interested perspective. For our reinforcement learning setting it could be the case that the TurnAway strategy was not found because the learning process got stuck in the Nash equilibrium of staying with the swarm (analogous to the mutual defection in the Prisoner’s Dilemma). If all agents keep their policy of staying close to each other, the one agent deviating has a higher chance of being chosen as prey. Our learning procedure is in conformity with this as one learning agent adjusts its policy in such a way that it obtains the highest reward while the policies of the other agents stay unchanged (during an episode). This assumption is also supported by looking at the procedure how agents are caught (see Figure 11): When the predator moves in the direction of the swarm, it collaboratively moves away, with a few agents being left behind. The community of the agents gets smaller and smaller as some sheer off until one is separated and picked as prey. This is also evident considering the

		B	
		<i>B</i> stays silent (cooperates)	<i>B</i> betrays (defects)
A	<i>A</i> stays silent (cooperates)	-1 / -1	-3 / 0
	<i>A</i> betrays (defects)	0 / -3	-2 / -2

Table 2: Prisoner’s dilemma payoff matrix

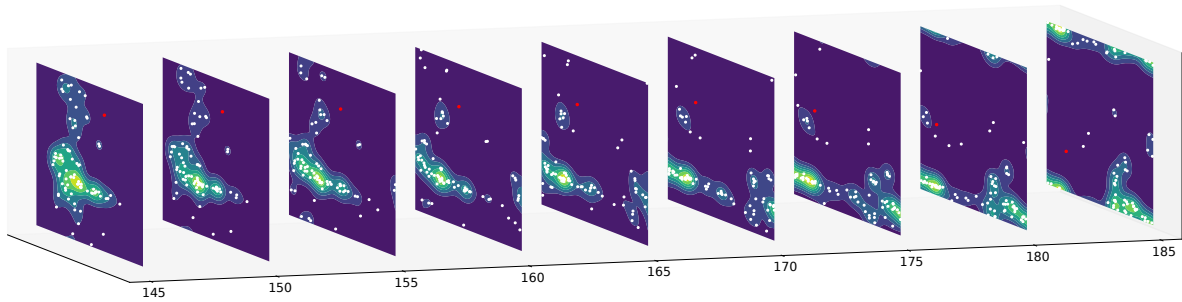


Figure 11: Separation of agents from the swarm before being caught.

density measurements of agents in the time steps before it is being caught. Figure 10 shows the density around an agent in accordance to the Kernel Density Estimation (cf. Figure 2 and 11) in the last 100 time steps of its life.

Conclusion and Future Work

With SELFish we showed that flocking behavior can emerge solely from self-interested agents. They are trained by multi-agent reinforcement learning to avoid being caught by a predator, who is distracted by flocks. Only one agent was trained at a time with a reward structure that encourages to avoid being caught as long as possible. After each episode the learning policy was copied onto all other agents. The results for SELFish_{DQN} and SELFish_{DDPG} concerning the alignment and cohesion but also with regard to the survival chances of the agents were compared with Boids, a common approach for algorithmic flocking simulations. Our results show that the measurements for the swarm are quite similar to Boids. In contrast to Boids, however, we never explicitly provided the agents with any model allowing to recognize or form a swarm. Our experiments thus show that it is possible to program swarms in a purely emergent way. Considering the survival of an agent, surprisingly, the reinforcement learning algorithms did not find the policy of simply turning away from the predator (without caring about flocking) although it yields higher accumulated rewards w.r.t. our reward structure. We propose that staying in the swarm is a Nash equilibrium (comparable to defecting in the Prisoner's dilemma), which would imply that swarms can form even when they are not the globally optimal behavior. Also, we would like to examine if other beneficial properties of a swarm, like increased hydrodynamic efficiency or easier search for food, which were not modeled by us, also lead to flocking behavior in a reinforcement scenario. This would probably also facilitate the steering of the swarm. Co-evolution of the behavior of the predator and its prey through reinforcement learning could be further investigated in our continuous environment. In our setting, agents could freely roam in a torus-like environment without obstacles or collisions. Naturally, there are enhancements to this like adding walls, obstacles and collisions between the agents.

References

- Bellman, R. (1957). Dynamic programming. *Princeton Press*.
- Egorov, M. (2016). Multi-agent deep reinforcement learning. *CS231n: Convolutional Neural Networks for Visual Recognition*.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Morihiro, K., Nishimura, H., Isokawa, T., and Matsui, N. (2008). Learning grouping and anti-predator behaviors for multi-agent systems. In *Int'l Conf. on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer.
- Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.
- Phillips, S. J., Anderson, R. P., and Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological modelling*, 190(3-4).
- Plappert, M. (2016). keras-rl. <https://github.com/keras-rl/keras-rl>.
- Poundstone, W. (1992). *Prisoner's Dilemma*. Doubleday.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH computer graphics*, volume 21. ACM.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *ICML*.
- Watson, G. (1983). *Statistics on spheres*. University of Arkansas lecture notes in the mathematical sciences. Wiley.

Engineering Application of Non-Reciprocal-Interaction-Based (NRIB) Model: Swarm Robotic System That Can Perform Spatially Distributed Tasks in Parallel

Takeshi Kano¹, Eiichi Naito², Takenobu Aoshima², and Akio Ishiguro¹

¹Research Institute of Electrical Communication, Tohoku University, Sendai, Miyagi, Japan 980-8577

²Business Innovation Division, Panasonic Corporation, Osaka, Japan

Correspondence: tkano@riec.tohoku.ac.jp

Abstract

We propose a simple decentralized control scheme for swarm robots that can perform spatially distributed tasks in parallel, drawing inspiration from the non-reciprocal-interaction-based (NRIB) model we proposed previously. Each agent has an internal state called “workload.” Each agent first moves randomly to find a task. When it finds a task, its workload increases, and then it attracts its neighboring agents to ask for their help. Simulation was used to demonstrate the validity of the proposed control scheme.

Recently, we have proposed an extremely simple model of collective behavior based on non-reciprocal interactions by drawing inspiration from friendship formation in human society (Kano et al. (2017a); Kano et al. (2017b)). It was demonstrated via simulations that various patterns emerge by changing the parameters (http://www.riec.tohoku.ac.jp/~tkano/ECAL_Movie1.mp4). Although this model (hereafter, we refer to Non-Reciprocal-Interaction-Based (NRIB) model) is highly abstract and it is difficult to conclude that it exactly mimics real social phenomena, the NRIB model can potentially contribute to various issues such as understanding the essence of collective behaviors of animals and active matters (Tanaka et al. (2017)) and designing swarm robotic systems.

This study focuses on the application of the NRIB model to the control of swarm robotic systems. Specifically, we extend the NRIB model to provide new insights into the problem of controlling swarm robots that can perform spatially distributed tasks in parallel (*e.g.*, swarm robots that can split into several groups to efficiently clean a room in which dust is spatially distributed) (Aşık and Akın (2017); Claes et al. (2015); Ducatelle et al. (2009)). We show via simulations that the robots driven by the proposed control scheme move and perform spatially distributed tasks in a quite reasonable manner.

We explain the proposed model hereafter. Agents exist on a two dimensional field wherein tasks are spatially distributed. Each agent has an internal state referred to “workload.” The position and the workload of agent i ($i = 1, 2, \dots, N$) are denoted by \mathbf{r}_i and C_i , respectively. The

amount of tasks at the position \mathbf{r} is denoted by $V(\mathbf{r})$. Each agent can detect the position and the workload of agents that exist in the area S_i , which is defined as the area within the circle whose radius and origin are r_{th} and \mathbf{r}_i , respectively.

The basic concept of the proposed control scheme is as follows. Each agent first moves at random to search for a task. Once it finds a task, *i.e.*, it enters an area where $V(\mathbf{r})$ is large, its motion slows down to perform the task. Concurrently, its workload increases, which works to attract its neighboring agents. As a consequence, the attracted agents can perform the task cooperatively. After the task is finished, the workload of the attracted agents decreases. Then, they repel each other and search for another task again.

Thus, the time evolutions of the workload of agent i , C_i , is given by the following equation:

$$\tau \dot{C}_i = \alpha V(\mathbf{r}_i) - C_i, \quad (1)$$

where α and τ are positive constants. Note that C_i is reset to 1 when it exceeds 1. Equation (1) means that C_i becomes large when agent i remains in the area where $V(\mathbf{r})$ is large.

The amount of task $V(\mathbf{r})$ decrements by ϵ in each time step at points where agents exist, because tasks are performed by the agents. Note that $V(\mathbf{r})$ is reset to zero when it becomes negative. Hence, according to Eq. (1), C_i decreases when the task is finished, *i.e.*, $V(\mathbf{r}_i)$ becomes zero.

The time evolution of the position of agent i , \mathbf{r}_i , is given by

$$\dot{\mathbf{r}}_i = (1 - C_i) \left[\sum_{j \in S_i} \{(aC_j - b)r_{ij}^{-1} - r_{ij}^{-2}\} \mathbf{e}_{ij} + \mathbf{n}_i \right], \quad (2)$$

where r_{ij} denotes the distance between agents i and j , \mathbf{e}_{ij} denotes the unit vector pointing from \mathbf{r}_i to \mathbf{r}_j , and a and b are constants satisfying $a > b > 0$. The components of the vector \mathbf{n}_i changes randomly at a certain time interval, which enables random walk of the robot. The term $(aC_j - b)r_{ij}^{-1} - r_{ij}^{-2}$ in Eq. (2) means that agent i approaches and repels from agent j when the workload of agent j , C_j , is high and low, respectively. Thus, agents tend to aggregate in areas where $V(\mathbf{r})$ is large. The term $(1 - C_i)$ denotes

the mobility of agent i . Specifically, agent i slows down as C_i increases and stops completely when $C_i = 1$. Hence, the robot performs tasks without moving until the $V(\mathbf{r})$ decreases to some extent.

Simulations were conducted to demonstrate the validity of the proposed control scheme. The periodic boundary condition is employed. Areas where $V(\mathbf{r})$ is large are initially distributed, and the initial positions of the agents are set to be random (Fig. 1(a)). The initial workload value is set to be 0.2 for all agents. Parameter values, which were determined by trial-and-error, are as follows: $N = 30$, $a = 2.0$, $b = 0.5$, $\alpha = 3.0$, $\tau = 2.0$, $\epsilon = 0.0005$, $r_{th} = 10$. The time step is 0.002. The x and y components of the vector \mathbf{n}_i are set to random values within the range of $[-0.25, 0.25]$, and are updated every 20000 time steps.

The result is shown in Fig. 1 (the movie can be downloaded from <http://www.riec.tohoku.ac.jp/~tkano/data.mpeg>). Agents aggregate in several areas where $V(\mathbf{r})$ is large, and their workload becomes large (arrows in Fig. 1(b)). They remain in these areas for a while, and then $V(\mathbf{r})$ decreases, *i.e.*, the tasks are performed (arrow in Fig. 1(c)). When $V(\mathbf{r})$ becomes almost zero, the workload becomes small and the agents begin to distribute while moving randomly (Fig. 1(d)). Then, when one of the agents enters an area where $V(\mathbf{r})$ is large, its workload increases and its neighboring agents aggregate in the area to perform the task (arrow in Fig. 1(e)). This process is repeated until the tasks are almost finished (Fig. 1(f)). In sum, agents are autonomously divided into several groups to perform spatially distributed tasks in parallel by using the proposed control scheme.

In conclusion, we proposed a decentralized control scheme for swarm robots, inspired by the NRIB model. In spite of the simplicity of the proposed control scheme, it was demonstrated via simulation that spatially distributed tasks were performed in parallel by the agents. Although comparison with other methods and solving several technical issues, *e.g.*, how to measure $V(\mathbf{r})$, r_{ij}^{-1} and \mathbf{e}_{ij} in real situations are still needed, we believe that the proposed control scheme could be used for various practical applications in which several spatially distributed tasks need to be performed efficiently. Moreover, it may be applicable when chasing moving targets, *e.g.* capturing fish and chasing criminals.

In future, we would like to examine the applicability of the proposed control scheme on an unstructured environment, *e.g.*, a field on which several obstacles exist. We would also like to develop real robots to validate the proposed control scheme in the real world.

Acknowledgements

The authors would like to thank Prof. Ken Sugawara of Tohoku Gakuin University and Naoki Matsui of Tohoku University for their insightful suggestions.

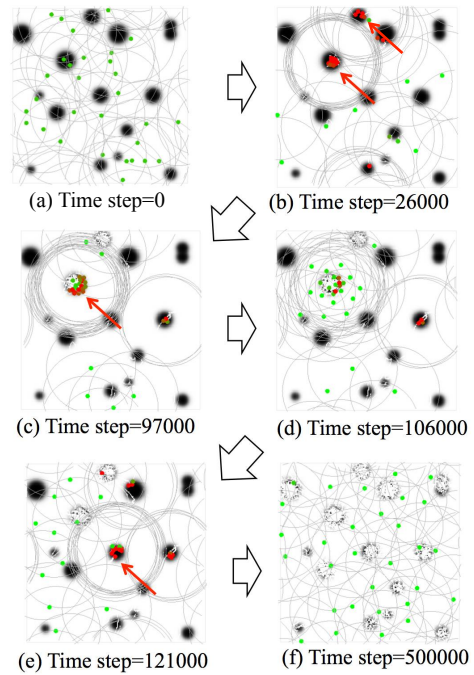


Figure 1: Simulation result. Areas in which $V(\mathbf{r}_i)$ is large are denoted by a dense color. Agents with high and low workload are colored by red and green, respectively. Gray circles denote the sensor range of the agents. The meanings of the red arrows are explained in the main text.

References

- Aşık, O. and Akın, H. (2017). Effective multi-robot spatial task allocation using model approximations. In Behnke, S., Sheh, R., Sariel, S., and Lee, D., editors, *RoboCup 2016: Robot World Cup XX. RoboCup 2016. Lecture Notes in Computer Science*, volume 9776, pages 243–255.
- Claes, D., Robbel, P., Oliehoek, F., Tuyls, K., Hennes, D., and Hoek, W. (2015). Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 881–890.
- Ducatelle, F., Förster, A., Di Caro, G., and Gambardella, L. (2009). New task allocation methods for robotic swarms. In *Proceedings of 9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions*.
- Kano, T., Osuka, K., Kawakatsu, T., and Ishiguro, A. (2017a). Mathematical analysis for non-reciprocal-interaction-based model of collective behavior. *Journal of the Physical Society of Japan*, 86:124004.
- Kano, T., Osuka, K., Kawakatsu, T., Matusi, N., and Ishiguro (2017b). A minimal model of collective behaviour based on non-reciprocal interactions. In *Proceedings of 14th European Conference on Artificial Life*, pages 237–244. Cambridge, MA: The MIT Press.
- Tanaka, S., Nakata, S., and Kano, T. (2017). Dynamic ordering in a swarm of floating droplets driven by solutal marangoni effect. *Journal of the Physical Society of Japan*, 86:101004.

Extended Artificial Pheromone System for Swarm Robotic Applications

Seongin Na¹, Mohsen Raoufi², Ali Emre Turgut³, Tomáš Krajník⁴, and Farshad Arvin¹

¹School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, Manchester, UK
seongin.na@student.manchester.ac.uk , farshad.arvin@manchester.ac.uk

² Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran

³Mechanical Engineering Department, Middle East Technical University, 06800 Ankara, Turkey

⁴Artificial Intelligence Centre, Faculty of Electrical Engineering, Czech Technical University, Prague, Czechia

Abstract

This paper proposes an artificial pheromone communication system inspired by social insects. The proposed model is an extension of the previously developed pheromone communication system, COS- Φ . The new model increases COS- Φ flexibility by adding two new features, namely, *diffusion* and *advection*. The proposed system consists of an LCD flat screen that is placed horizontally, overhead digital camera to track mobile robots, which move on the screen, and a computer, which simulates the pheromone behaviour and visualises its spatial distribution on the LCD. To investigate the feasibility of the proposed pheromone system, real micro-robots, *Colias*, were deployed which mimicked insects' role in tracking the pheromone sources. The results showed that, unlike the COS- Φ , the proposed system can simulate the impact of environmental characteristics, such as temperature, atmospheric pressure or wind, on the spatio-temporal distribution of the pheromone. Thus, the system allows studying behaviours of pheromone-based robotic swarms in various real-world conditions.

Introduction

Social insects are known for conducting complex tasks with coordination in highly effective ways. They carry out complicated tasks such as food foraging, aggregating, mating, etc. using limited perception and memory capabilities (Schmickl et al., 2009; Jackson and Morgan, 1993; Agosta, 1992; Schmickl et al., 2016; Michener and Press, 1974). Those complex tasks require effective communication mechanisms within a group of insects. As a key to achieve the effective communication, several social insects use pheromones which is a medium for the stigmergic behaviours. Stigmergy is an indirect coordination mechanism using a shared communication medium (Theraulaz and Bonabeau, 1999). A medium created by an agent in the environment actuates the other agents to perform certain actions without any direct communication between them (Heylighen, 2016; Marsh and Onof, 2008). As an example of how stigmergy is used, an ant releases trail pheromone when it detects food and other ants detect the trail and follow it (Jackson and Ratnieks, 2006; Wyatt, 2003; Sumpter and Beekman, 2003).



Figure 1: Artificial pheromone system including a horizontally placed LCD screen, overhead camera for tracking system and a computer.

This coordination mechanism has remarkable features compared with other traditional methods such as direct communication. First of all, it achieves optimisation through positive and negative feedback (Jackson and Ratnieks, 2006; Sumpter and Beekman, 2003; Theraulaz and Bonabeau, 1999; Heylighen, 2016). By reinforcing or suppressing the medium at a position depending on how close it is to the goal, the agents can carry out the tasks in an optimal manner, e.g. the shortest path for food foraging is created after multiple iterations with reinforcement and suppression on the pheromone trail. Secondly, it does not require complex functionality for each agent (Heylighen, 2016). Compared to the conventional methods, stigmergy demands significantly lower capability for each agent. For instance, an agent does not need to save information into its memory because it is saved in and read from the media. As another example, an agent does not have planning or anticipation ability since it performs tasks only based on the present media. The advantages of stigmergy have inspired researchers who study swarm robotics especially adopting pheromone (Fossum et al., 2014; Purnamadajaja and Russell,

2007; Font Llenas et al., 2018). Researchers have developed and implemented artificial pheromone systems with different means from chemical substances and RFID chips to light-based means and virtualization system (Arvin et al., 2018b; Fujisawa et al., 2014; Herianto and Kurabayashi, 2009; Arvin et al., 2015; Valentini et al., 2018; Beckers et al., 2000). The methods using chemical means duplicate evaporation, diffusion, locality and reactivity which are the characteristics of pheromone in the real world (Fujisawa et al., 2014). Besides, different kinds of chemical substances can be used as different kinds of pheromone which have distinct functionality (Purnamadajaja and Russell, 2007). Although their similarities to the pheromone in the real world contributes to mimic the important features of the pheromone, it is difficult to control the properties of the chemical substances such as evaporation and diffusion rates. Hence, it is challenging to be used as an experimental tool (Fujisawa et al., 2014; Sugawara et al., 2004). Additionally, sensing technology for chemical substances has to be improved to detect reasonably small amount of chemicals (Purnamadajaja and Russell, 2010).

The methods using RFID chips have advantages in swarm robotics because they use low-cost data carriers and it is free from batteries. However, the fixed size of data carriers does not allow this method to be used with different resolutions (Herianto and Kurabayashi, 2009). Virtualization system is a relatively new method implementing pheromone communication which reads data from robots, sends the data to virtual map for mapping and sharing to the robots in the swarm. In spite of its bidirectional communication and capacity for large-scale swarm robotics application, the virtualisation of sensors and actuators are restricted by the resolution of the grid (Valentini et al., 2018). Light-based methods have a number of advantages which cover certain limitations present in the other methods (Arvin et al., 2015; Garnier et al., 2007). The characteristics of pheromone such as evaporation and diffusion are easily controllable. Furthermore, light-based methods have significantly higher resolution than the methods using RFID chips or virtualization environment. Moreover, different types of pheromone can be implemented since various colors of light can be used (Arvin et al., 2018a; Jackson and Ratnieks, 2006).

One of the light-based artificial pheromone communication systems, COS Φ , (Arvin et al., 2015) has four advantages as follows: (1) Precise pheromone trail can be created by using high-resolution horizontal LCD screen as an arena to project light-based artificial pheromones. (2) Characteristics of pheromone such as evaporation and thickness are easily and precisely modified. (3) Pheromone can be overlapped or suppressed so that positive feedback and negative feedback can be implemented. (4) Numerous types of pheromone can be generated using RGB colors. Although the system has advantageous features listed above, there are points which can be developed for more diverse functions

and applications.

The goal of this work is extending COS Φ system to cover its limitation. Although evaporation and injection of pheromones was clearly replicated in the system, diffusion was not implemented whereas it is a necessary feature of temporal pheromone development (Herianto and Kurabayashi, 2009; Sugawara et al., 2004; Ji et al., 2013). Furthermore, the mathematical model of the pheromone updating is expanded. Therefore, it includes advection of the pheromone by the wind and the advection effect is applied in the system. This expanded system is expected to offer more options to users for bio-inspired swarm robotic studies (Figure 1). Adding the two phenomena has the meaning described below.

- *Diffusion* is the movement of molecules from the area of higher concentration to those of lower concentration. Social insects follow pheromone trail, they detect diffused pheromone rather than contacting to the trail and moving directly along it (Wyatt, 2003). Despite its importance, it was not modelled and implemented in the previous research. Therefore, it is meaningful to apply diffusion effect.
- *Advection* is the transfer of substances or any quantity by the flow of a fluid, like wind. In fact, pheromone, as a substance, is transported by wind which is the flow of the air. Through applying the advection effect with different velocity and direction, we have a more reliable model and figure out how wind influences to the pheromone communication.

This paper is arranged as follows: Section II presents artificial pheromone system, COS Φ , Section III presents the proposed extended properties of the COS Φ , Section IV provides experimental configurations, and Section V presents the results from experiments and discusses the outcomes.

Artificial Pheromone System, COS Φ

In the previous project (Arvin et al., 2015), COS Φ (Communication System via Pheromone) was introduced. It has a software system and a robotic platform. The software system consists of two major parts. The first part is a visual localization system, *SwarmCon* (Krajník et al., 2014). It reads the position of robots using a camera attached above the arena and sends their information to the pheromone releasing system which displays light on the LCD screen. The second part is a pheromone releasing system. After receiving the position data from the localisation system, the pheromone releasing system computes where and how much the pheromone will be injected accordingly. The system also repetitively updates the obtained pheromone data reflecting the development of released pheromone over time in reality. The system subsequently displays a gray-scale image based on the pheromone data on the LCD screen.

COS Φ has three remarkable features which make this system more reliable and user-friendly to be used as an experimental platform for researchers in the field of biology, swarm robotics or other related disciplines: i) It provides highly precise location data of robots and pheromone through *SwarmCon* system. Moreover, its resolution is equal to that of the LCD screen; hence, the smallest controllable size of the pheromone is equivalent to a pixel of the screen. ii) It has high flexibility allowing the users to change experimental conditions, settings and even initial characteristics of the pheromone in order to fit their needs. iii) It is a low-cost platform. COS Φ requires only a low-cost USB camera, an LCD screen as an arena, and the low-cost micro-robots, *Colias* (Arvin et al., 2014).

Pheromone Model

The gray-scale image displayed on the LCD screen computed by the system based on Equation 1.

$$\mathbf{I}(x, y) = \sum_{i=1}^n c_i \Phi_i(x, y) \quad (1)$$

The brightness of a pixel at a position (x, y) , $\mathbf{I}(x, y)$ is determined by the multiplication of $\Phi_i(x, y)$ which represents i^{th} pheromone intensity and c_i which denotes how strong the i^{th} pheromone is displayed on the screen. Since the system can use multiple different types of pheromone, the brightness of a pixel is the sum of the effect of n different pheromones released at a position.

In the previous project (Arvin et al., 2015), the change in the intensity of pheromone at a position (x, y) at a time, $\dot{\Phi}_i$, is defined by

$$\dot{\Phi}_i = \frac{\ln 2}{e_{i\Phi}} \Phi_i(x, y) + \kappa_i \Delta \Phi_i(x, y) + \iota_i(x, y) \quad (2)$$

where $e_{i\Phi}$, κ_i are the evaporation constant and diffusion constant respectively and $\iota_i(x, y)$ denotes the amount of injected pheromone at a time. In the previous study (Arvin et al., 2015), only evaporation and injection terms were applied into the development in the pheromone concentration. The two phenomena influence the pheromone in the ways explained in the below two subsections. Diffusion model is described in the next section.

Evaporation

Evaporation of the deposited pheromone in the real world is a fundamental feature of pheromone as a chemical substance. The intensity of pheromone decays over time due to evaporation. This model exhibits that the pheromone exponentially decays over time without spreading to the adjacent positions. Figure 2 shows how the pheromone evaporates with $e_{i\Phi} = 20$ at $t = \{0, 1, 2, 3\}$ s illustrated by the black, red, green and blue line respectively. The pheromone is initially released from $x \in [200, 300]$ with the intensity of 255.

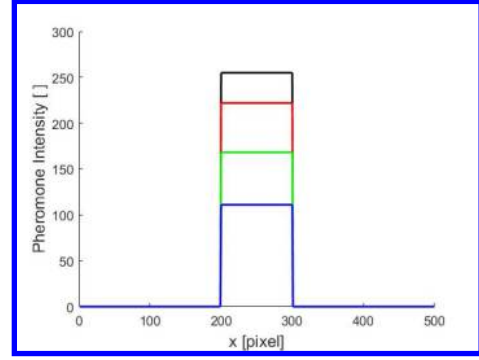


Figure 2: Evaporation of pheromone over time $t = 0$ s (black), $t = 1$ s (red), $t = 2$ s (green) and $t = 3$ s (blue).

Its horizontal axis denotes the x position of the pheromone in the 2-D arena with the size of 500×500 pixels, and the vertical axis is the intensity of the pheromone between 0 and 255.

Injection

The injection $\iota_i(x, y)$ represents the amount of the newly injected i^{th} pheromone at the position (x, y) . In the system, $\iota_i(x, y)$ is defined by

$$\iota_i(x, y) = \begin{cases} s_\Phi, & \text{if } \sqrt{(x - x_r)^2 + (y - y_r)^2} \leq l_\Phi / 2 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where, (x_r, y_r) is the position of the robot, l_Φ is the diameter of the pheromone injected at the time and s_Φ is the pheromone release rate.

Extended Pheromone System

Based on the previous model of the pheromone temporal development and the mathematical model of moving substances through the fluid proposed in (Stam, 2005), the evolution of the pheromone intensity field is developed as follows:

$$\dot{\Phi}_i(x, y) = -\mathbf{u} \cdot \nabla \Phi_i(x, y) - \frac{\ln 2}{e_{i\Phi}} \Phi_i(x, y) + \kappa_i \nabla^2 \Phi_i(x, y) + \iota_i(x, y), \quad (4)$$

where, \mathbf{u} is a two-dimensional vector field which represents the wind velocity field. This equation stems from Navier-Stokes equation which illustrates the motion of fluids. It is assumed that the pheromone is a non-reactive substance. Hence, the pheromone does not vary by chemical reaction but the factors described in Equation 4.

Diffusion

As previously mentioned, diffusion is one of the factors of the pheromone behaviour. In the Equation 4, diffusion is described as $\kappa_i \nabla^2 \Phi_i(x, y)$. For the sake of computation

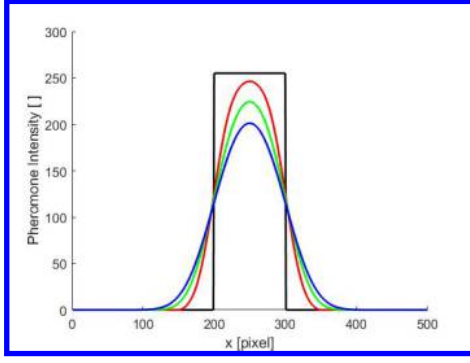


Figure 3: Diffusion of pheromone over time $t = 0$ s (black), $t = 1$ s (red), $t = 2$ s (green) and $t = 3$ s (blue).

simplicity, diffusion is implemented with the approximate model using the Gaussian filter. It contains analogous features to what the actual diffusion model has. i) The total amount of the pheromone is conserved while it diffuses ii) The pheromone at a position is distributed to the neighbour positions and the pheromone from surrounding positions diffuses to the position. Moreover, the intensity of the pheromone at a position increases if it is surrounded by greater intensity of the pheromone. Conversely, the intensity decreases if the intensity of the pheromone of its nearby positions is lower than it.

$$\Phi_i^{k+1}(x, y) = (\omega * \Phi_i^k)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b \omega(s, t) \Phi_i^k(x - s, y - t), \quad (5)$$

where ω is a kernel matrix with the size of $(2a+1) \times (2b+1)$ which is convolved with the matrix of the i^{th} pheromone strength Φ_i^k at k^{th} iteration, and ω is defined by the equation below:

$$\omega(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad w \in \mathbb{R}^{2a+1 \times 2b+1}, \quad (6)$$

where the element at $((a+1), (b+1))$ is assigned as (0,0) of ω and σ is the standard deviation of elements of ω . The elements of the kernel are determined based on the Gaussian distribution. The further an element from the center of the matrix is, the smaller value the element has. Figure 3 shows the diffusion of pheromone at $t = \{0, 1, 2, 3\}$ s illustrated by the black, red, green and blue line respectively. The axes are identical to Figure 2. At $t = 0$ s, the pheromone released on $x = 200-300$ has the intensity of 255. The arena size is also 500×500 pixels and the kernel size is 95×95 pixels and $\sigma = 20$. Every time the pheromone is updated with the diffusion, the area where the pheromone is deposited expands while the maximum intensity of the pheromone decreases.

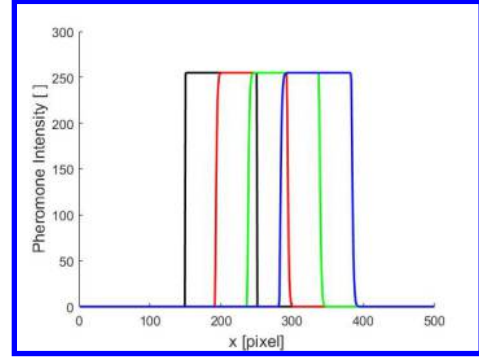


Figure 4: Advection of pheromone over time $t = 0$ s (black), $t = 1$ s (red), $t = 2$ s (green) and $t = 3$ s (blue).

Advection

The change in pheromone intensity at the position (x, y) by advection is simply described as $\mathbf{u} \cdot \nabla \Phi_i(x, y)$ in Equation 4. The dot multiplication of the wind velocity field \mathbf{u} and the gradient of the pheromone can be expressed as:

$$\mathbf{u} \cdot \nabla \Phi_i(x, y) = u_x \cdot \frac{\partial \Phi_i(x, y)}{\partial x} + u_y \cdot \frac{\partial \Phi_i(x, y)}{\partial y}, \quad (7)$$

where u_x and u_y are the x - and y -component of \mathbf{u} , respectively. In this project, same magnitude of u_x and u_y are applied on the entire field. In other words, the wind with a given magnitude and direction blows equally at all the positions. Figure 4 shows the advection of the pheromone along the x axis where $u_x = 50$, namely, the wind speed is 50 pixel/s. The pheromone is initially injected from $x \in [200, 300]$ with the intensity of 255. The black, red, green and blue lines represent the intensity of the pheromone on the x position at $t = \{0, 1, 2, 3\}$ s. It is illustrated that the wind causes the pheromone to be transferred in the almost parallel manner without a considerable change in the shape.

Experimental Setup

There are three different experimental configurations to study: i) effects of diffusion, ii) effects of advection, and iii) combination of both diffusion and advection on the behaviour of the robots. Also, we run a set of experiments excluding pheromone effects as the control.

A circular cue (with diameter of 25 cm) with a maximum intensity of pheromone at the centre, source of pheromone injection, is projected on the right hand side of the area (screen) and the robots are randomly placed on the left hand side of the arena. Each experiment takes 5 min and we analyse the collective behaviour of the proposed system with two metrics which are: i) number of robots at the pheromone source and ii) average distance of the robots from the centre of the pheromone source.

It must be mentioned that the robots do not deposit pheromone during experiments; hence, they only change

their direction towards the highest intensity pheromone trail. The pheromone cue is only injected once at the initial stage $t = 0$ s of each experiment.

Arena Configuration

Arena that is used in this work is analogous to the experimental setup presented in (Arvin et al., 2015). It includes a horizontally placed 42" flat LCD screen, a USB camera, and a computer to track the robots and manage the pheromone system. Figure 1 shows the experimental setup that was used in this work.

Utilizing this system, we are able to determine whether or not a robot is reached the cue. In this regard, at the beginning of each experiment, we store the brightness of each pixel in a matrix I_a . Then, the localisation system detects the tags of four corners and calculates the transformation between the arena and camera coordinate systems. Similarly, it allows us to detect the robots and measure their positions on the field. Then, the visual system takes the brightness of the current image as I_c and compares its with I_a . By finding the largest circular continuous segment in I_c and calculating its position (x_c, y_c) and average brightness b_c , the cue zone is determined.

Robotic Platform

Colias micro-robot (Arvin et al., 2014) was utilised in this study to test the feasibility of the proposed extensions (Figure 5-a). Colias is a low-cost open-source mobile robot uses an AVR micro-controller as its main processor. It has three short-range infrared proximity sensors at the front to detect obstacles and other robots. Colias has two light intensity sensors (Figure 5-b) at the bottom of the robot, s_l and s_r on the left and right hand side, respectively. Motors' rotational velocities, N_l and N_r , are directly controlled using measurements from these two sensors:

$$\begin{aligned} N_l &= \frac{s_l - s_r}{\alpha} + \beta, \\ N_r &= \frac{s_r - s_l}{\alpha} + \beta, \end{aligned} \quad (8)$$

where, α is the velocity sensitivity coefficient and β is a biasing speed. In this work, β relies on the average pheromone intensity, because the higher intensity results in the lower velocity:

$$\beta = 100 - \frac{s_r + s_l}{2}. \quad (9)$$

This relation between β and the sensors measurements is tuned empirically. The main idea is to reduce the speed of motion near source of the pheromone to keep the robots at the high intensity pheromone cue. Therefore, there are two direct impacts of the pheromone on the robots behaviour which are: i) controlling angular velocity of the robot to direct robot to the centre of the pheromone and ii) reducing speed of the robot with increasing pheromone intensity.

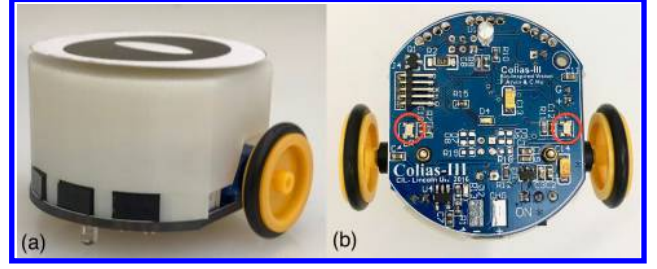


Figure 5: (a) Colias micro-robot and (b) bottom of the robot including light intensity sensors.

Experiments

Three pheromone configurations were implemented. The first configuration was diffusion speed with two settings:

1. *Diffusion-A*: pheromones diffuse by a rate which results in diffusion of 25% of total pheromone till $t=300$ s ($e_{i\Phi} = 1000$, $a, b = 7$ and $\sigma = 0.3$)
2. *Diffusion-B*: pheromones diffuse by a rate which results in diffusion of 50% of total pheromone till $t=300$ s ($e_{i\Phi} = 1000$, $a, b = 7$ and $\sigma = 6$)

The second set of experiments were conducted with various advection speeds:

1. *Advection-A*: pheromone spot moves with the wind speed of 2.27 pixel/s to the left hand side of the arena during $t = 300$ s ($e_{i\Phi} = 1000$, $u_x = 2.27$ and $u_y = 0$)
2. *Advection-B*: pheromone spot moves with the wind speed of 4.53 pixel/s to the left hand side of the arena during $t = 300$ s ($e_{i\Phi} = 1000$, $u_x = 4.53$ and $u_y = 0$)

The third set of experiments were conducted with combining *Diffusion-B* and *Advection-B*. Apart from the 5 distinct configurations which are already mentioned, we applied a simple experiment in which the effect of neither diffusion nor advection is considered. All of these 6 experiments were conducted with two population sizes of $N \in \{4, 6\}$ robots. Moreover, for each configuration, 5 independent runs are applied.

In order to assess the effect of these two phenomena on the behaviour of the system, two different variables are defined. The number of robots on pheromone, a dimensionless variable, is the ratio of number of robots which are located within the cue to all robots in the arena. The second variable is ‘‘coherence distance’’, d_{coh} , indicating the average distance of all robots to the center of cue, which is defined by the following equation:

$$d_{coh} = \frac{1}{N} \sum_{i=1}^N \|(x_i, y_i), (x_c, y_c)\| \quad (10)$$

in which, (x_i, y_i) and (x_c, y_c) are the location of i -th robot and cue center, respectively, and $\|\cdot\|$ is the Euclidean distance operator between two points in a 2-D space.

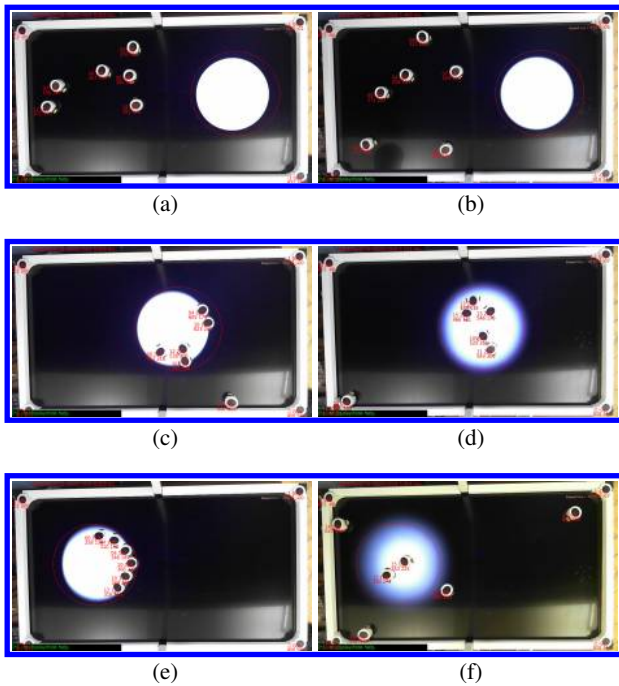


Figure 6: The arena, robots and pheromone from the camera perspective, illustrating the experiments of Advection-B, as well as Diffusion-B + Advection-B . Each row is related to a specific time $t = \{0, 60, 180\}$ s, respectively.

Results & Discussion

The results of the above-mentioned experiments are presented in this section. Figure 6 shows several images from randomly selected experiments at various times showing the effect of diffusion and advection on the system. The images in the left column show an experiment with the configuration of Advection-B with 6 robots, and the images of the right column have the configuration of mixed Diffusion-B and Advection-B. In the left column, the effect of wind on the movement of the cue is shown vividly. The cue starts to move from the right side toward the left side of the arena due to advection. In this case, the number of robots which were able to find and stay at the cue is raised. The effect of diffusion on the pheromone is clearly shown in the right column; The sharp edge of the cue fades by time, resulting in lower number of robots remained in the cue. To such an extent that some of them lost the cue and wander in the arena. On the other hand, it makes the robots to stay closer to the center of cue, and as a result, the coherency of the robots increases. Meanwhile, the advection affects the cue location.

To study the effect of diffusion on the behaviour of robots, Figure 7 demonstrates the pheromone intensity and coherence distance for three different configurations set of experiments: i) No effects, ii) Diffusion-A, and iii) Diffusion-B. As shown in Figure 7 (a), the number of robots reached the

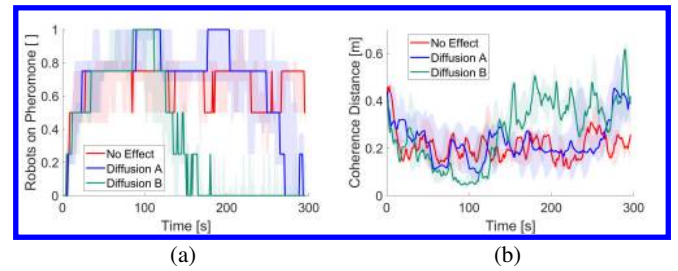


Figure 7: Effect of diffusion on the behaviour of robots. (a) The number of robots on pheromone and (b) Coherence distance

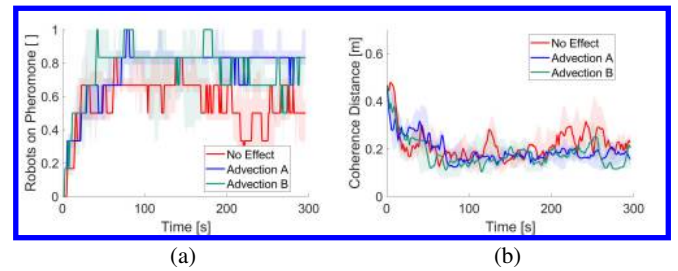


Figure 8: Effect of advection on the behaviour of robots. (a) The number of robots on pheromone and (b) Coherence distance

spot in Diffusion-A and B are close to that of with no effect configuration. However, when time went and the pheromone diffused to the neighbor areas, the cue shrunk and the robots left the cue. As what we expected, this separation happened for Diffusion-B much earlier than Diffusion-A. The diffusion has another impact on the behaviour of robots, which results in more coherence. Prior to separation, the robots in the cue stayed closer to the center of cue. This can be seen not only from Figure 6 (d), but also from the Figure 7 (b).

The effect of advection on the behaviour of robots can be seen in Figure 8, in which 6 robots are utilized. Compared to the 'no effect' configuration, the number of robots which were able to find the cue increased when advection is considered. However, the influence of wind on the coherency is negligible.

Finally, the result of combination of both concepts are illustrated in Figure 9 besides the result of 'no effect' case. We can see that the number of robots on pheromone is generally more than the simple case, but, same as Figure 7, it starts to drop after a time called separation time.

To investigate the effects of different factors on collective behaviour of the swarm, all the results were statistically analysed using 2-way Analysis of Variance (ANOVA). Table 1 and Table 2 show the results of ANOVA test for number of robots on pheromone and coherence distance.

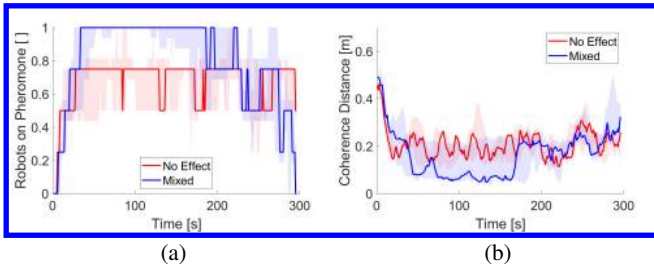


Figure 9: Effect of both diffusion and advection on the behaviour of robots. (a) The number of robots on pheromone and (b) Coherence distance

Table 1: Results of ANOVA test for number of robots on pheromone

Factors	p -value	F -value
Exp. Configuration	0.000	132.708
Time	0.000	2.512

Based on the results, both factors, time and experiment configurations, significantly affected the swarm. However, the configuration was the most significant factor in number of robots on pheromone ($F=132.708$) and coherence distance ($F = 80.156$).

Table 2: Results of ANOVA test for coherence distance

Factors	p -value	F -value
Exp. Configuration	0.000	80.156
Time	0.000	4.639

Conclusion

This paper added two new properties – *diffusion* and *advection*– to the previously developed artificial pheromone system, COS- Φ . Three set of experimental configurations were conducted to investigate the performance of the proposed properties. Coherence distance and number of robots on the pheromone spot were tracked during experiments. The results were statistically analysed and the most effective factor was detected. The future work is to include several robots with capability of injecting pheromones and to study inter-robot interactions using the updated artificial pheromone communication.

Acknowledgement

This work was supported by EPSRC RNE (EP/P01366X/1), EPSRC RAIN (EP/R026084/1), and Czech Ministry of Science and Education grant ‘Research Centre for Informatics’ number CZ.02.1.01/0.0/0.0/16_019/0000765.

References

- Agosta, W. (1992). *Chemical Communication: The Language Of Pheromones*. Henry Holt and Company.
- Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., and Lennox, B. (2018a). Mona: an affordable open-source mobile robot for education and research. *Journal of Intelligent & Robotic Systems*, pages 1–15.
- Arvin, F., Krajník, T., Turgut, A. E., and Yue, S. (2015). COS Φ : artificial pheromone system for robotic swarms research. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 407–412. IEEE.
- Arvin, F., Murray, J., Zhang, C., and Yue, S. (2014). Colias: An autonomous micro robot for swarm robotic applications. *International Journal of Advanced Robotic Systems*, 11(7):113.
- Arvin, F., Turgut, A. E., Krajník, T., Rahimi, S., Okay, I. E., Yue, S., Watson, S., and Lennox, B. (2018b). Φ Clust: Pheromone-Based Aggregation for Robotic Swarms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4288–4294.
- Beckers, R., Holland, O. E., and Deneubourg, J.-L. (2000). *From Local Actions to Global Tasks: Stigmergy and Collective Robotics*, pages 1008–1022. Springer Netherlands, Dordrecht.
- Font Llenas, A., Talamali, M. S., Xu, X., Marshall, J. A., and Reina, A. (2018). Quality-sensitive foraging by a robot swarm through virtual pheromone trails. In *International Conference on Swarm Intelligence*, pages 135–149. Springer.
- Fossum, F., Montanier, J., and Haddow, P. C. (2014). Repellent pheromones for effective swarm robot search in unknown environments. In *IEEE Symposium on Swarm Intelligence*, pages 1–8.
- Fujisawa, R., Dobata, S., Sugawara, K., and Matsuno, F. (2014). Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence*, 8(3):227–246.
- Garnier, S., Tâche, F., Combe, M., Grimal, A., and Theraulaz, G. (2007). Alice in pheromone land: An experimental setup for the study of ant-like robots. In *IEEE Swarm Intelligence Symposium*, pages 37–44.
- Herianto and Kurabayashi, D. (2009). Realization of an artificial pheromone system in random data carriers using rfid tags for autonomous navigation. In *2009 IEEE International Conference on Robotics and Automation*, pages 2288–2293.

- Heylighen, F. (2016). Stigmergy as a universal coordination mechanism i: Definition and components. *Cognitive Systems Research*, 38:4–13. Special Issue of Cognitive Systems Research Human-Human Stigmergy.
- Jackson, B. D. and Morgan, E. D. (1993). Insect chemical communication: Pheromones and exocrine glands of ants. *CHEMOECOLOGY*, 4(3):125–144.
- Jackson, D. E. and Ratnieks, F. L. (2006). Communication in ants. *Current Biology*, 16(15):R570 – R574.
- Ji, J., Song, X., Liu, C., and Zhang, X. (2013). Ant colony clustering with fitness perception and pheromone diffusion for community detection in complex networks. *Physica A: Statistical Mechanics and its Applications*, 392(15):3260–3272.
- Krajník, T., Nitsche, M., Faigl, J., Vaněk, P., Saska, M., Přeučil, L., Duckett, T., and Mejail, M. (2014). A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 76(3):539–562.
- Marsh, L. and Onof, C. (2008). Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, 9(1):136 – 149. Perspectives on Social Cognition.
- Michener, C. and Press, B. (1974). *The Social Behavior of the Bees: A Comparative Study*. Belknap Press Series. Belknap Press of Harvard University Press.
- Purnamadajaja, A. H. and Russell, R. A. (2007). Guiding robots’ behaviors using pheromone communication. *Autonomous Robots*, 23(2):113–130.
- Purnamadajaja, A. H. and Russell, R. A. (2010). Bi-directional pheromone communication between robots. *Robotica*, 28(1):6979.
- Schmickl, T., Stefanec, M., and Crailsheim, K. (2016). How a life-like system emerges from a simple particle motion law. *Scientific reports*, 6:37969.
- Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., and Crailsheim, K. (2009). Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.
- Stam, J. (2005). Stable fluids. pages 121–128.
- Sugawara, K., Kazama, T., and Watanabe, T. (2004). Foraging behavior of interacting robots with virtual pheromone. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 3074–3079.
- Sumpter, D. J. and Beekman, M. (2003). From nonlinearity to optimality: pheromone trail foraging by ants. *Animal Behaviour*, 66(2):273 – 280.
- Theraulaz, G. and Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5(2):97–116.
- Valentini, G., Antoun, A., Trabattoni, M., Wiandt, B., Tamura, Y., Hocquard, E., Trianni, V., and Dorigo, M. (2018). Kilogrid: a novel experimental environment for the kilobot robot. *Swarm Intelligence*, 12(3):245–266.
- Wyatt, T. (2003). *Pheromones and Animal Behaviour: Communication by Smell and Taste*. Cambridge University Press.

Applying Social Network Analysis to Agent-Based Models: A Case Study of Task Allocation in Swarm Robotics Inspired by Ant Foraging Behavior

Georgina Montserrat Reséndiz-Benhumea¹, Tom Froese^{1, 2}, Gabriel Ramos-Fernández^{1, 3} and Sandra E. Smith-Aguilar⁴

¹Institute for Applied Mathematics and Systems Research, National Autonomous University of Mexico, Mexico City, Mexico

²Center for the Sciences of Complexity, National Autonomous University of Mexico, Mexico City, Mexico

³Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional, Mexico City, Mexico

⁴Conservación Biológica y Desarrollo Social A.C., Mexico City, Mexico

gmontserb@comunidad.unam.mx

Abstract

Social network analysis and agent-based modeling are two approaches used to study biological and artificial multi-agent systems. However, so far there is little work integrating these two approaches. Here we present a first step toward integration. We developed a novel approach that allows the creation of a social network on the basis of measures of interactions in an agent-based model for purposes of social network analysis. We illustrate this approach by applying it to a minimalist case study in swarm robotics loosely inspired by ant foraging behavior. For simplicity, we measured a network's inter-agent connection weights as the total number of interactions between mobile agents. This measure allowed us to construct weighted directed networks from the simulation results. We then applied standard methods from social network analysis, specifically focusing on node centralities, to find out which are the most influential nodes in the network. This revealed that task allocation emerges and induces two classes of agents, namely foragers and loafers, and that their relative frequency depends on food availability. This finding is consistent with the behavioral analysis, thereby showing the compatibility of these two approaches.

Introduction

Social network analysis (SNA) has been widely used in the study of biological multi-agent systems (Krause et al., 2015). In recent years, there has been an increasing interest in analyzing animal social networks (Scott and Carrington, 2014). For example, there are studies in social networks of spider monkeys (Ramos-Fernández et al., 2009), crows (Rutz et al., 2012) and social insects (Charbonneau et al., 2013). Similarly, agent-based modeling (ABM) has been applied to the same area. Ramos-Fernández et al. (2006) studied the emergence of animal social structure using agent-based models. Guo and Wilensky (2016), researchers in *Alife*, have demonstrated the utility of agent-based models of social insects as powerful tools to understand complex system principles. Moreover, Wang et al. (2019) studied collective behavior of bacteria, which use signaling systems known as quorum-sensing (QS) to communicate and cooperate. They used an agent-based modeling approach to understand the emergence of complex QS architectures and functions.

On the other hand, there are few studies using these two approaches (SNA and ABM) in combination in artificial multi-agent systems (MAS), particularly, in swarm robotics. Swarm robotics is a recent approach in the field of artificial swarm intelligence to study the coordination of multi-robot systems (MRS) without central control inspired on swarms observed in nature, such as those of social insects. Collective behavior emerges from robot-robot and robot-environment interactions (Tan and Zheng, 2013). There is a strong potential found in mimicking social insect behavior because this is highly convenient for solving complex coordination tasks (Alers et al., 2014). For example, ant foraging behavior induces task allocation as an emergent property, which is suitable for swarm robotics (Labella et al., 2006).

In this study, we are interested in applying social network analysis to agent-based modeling. There are previous studies that successfully combined SNA and ABM (Fontana and Terna, 2015) or SNA and MAS (Ma et al., 2009; Grant, 2009). For a better understanding, we have developed a taxonomy of social interaction models based on the approach of Powers et al. (2018), as shown in Figure 1.

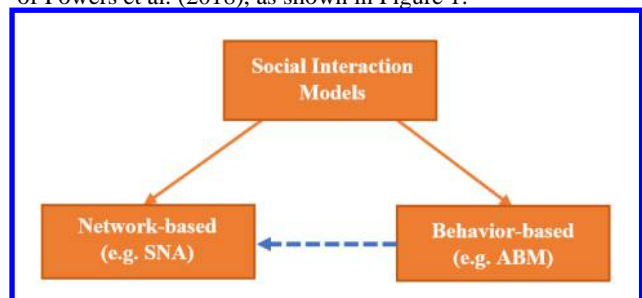


Figure 1: A taxonomy of social interaction models. We have two classes of social interaction models: network-based and behavior-based. Social network analysis (SNA) is an instance of network-based model and agent-based modeling (ABM) is an instance of behavior-based model. We propose there should be a bridge (dashed blue arrow) from behavior-based to network-based models to have a complete perspective of the network dynamics in a complex system in order to get new insights on their emerging properties. That is, moving from agent-based modeling to social network analysis.

Figure 1 shows our proposed taxonomy where we consider there should be a bridge from behavior-based (e.g. ABM) to network-based (e.g. SNA) models to have a complete perspective of the network dynamics in a complex system in order to get new insights on its emerging properties. Thus, our representation of moving from ABM to SNA.

We found that this approach (from ABM to SNA) has not been exploited in foraging and task allocation in swarm robotics. However, there are previous papers using either one of these two approaches (ABM or SNA). Iba (2013), developed agent-based modeling and simulations with swarms; Palestra et al. (2017), modeled and simulated rescue robots using the swarm robotics approach; Koval et al. (2009), introduced a social network to a swarm robotics system in order to improve accuracy in automatic target recognition.

The main goal of this study is to apply our proposed approach, from agent-based modeling to social network analysis, to a case study in swarm robotics inspired by ant foraging behavior to show task allocation as an emergent property of the complex system.

The case of ant foraging behavior in swarm robotics

Task allocation, in social insects, refers to the processes by which a task is carried out by each member of the colony. As examples, we have foraging and brood care. Additionally, these processes adapt to changing conditions (Gordon, 2016). In this paper, we are interested in task allocation as an emergent property of ant foraging behavior.

The main features of ant foraging behavior can be summarized as follows (Labella et al., 2006):

- The ant explores the environment in random displacements until it finds food. There are three cases of how to take it to the nest: (i) the ant pulls it, if it is not too heavy, (ii) the ant cuts it, (iii) the ant uses long or short recruitment (as a result of spreading pheromone trail).
- In individual or collective retrieval, food is directly pulled to the nest.
- When a forager returns to the nest, it unloads food by mouth-to-mouth contact into the *crops* (a pouch located just upstream of their stomachs) of other ants (Greenwald et al., 2018).
- After retrieving food, the ant goes straight back to the location where it found food.

Deneubourg et al. (1987) modeled an ant of the species *Pachycondyla apicalis* as an agent. Each agent has a probability P_l of leaving the nest, that varies depending on prior successes or failures. That is, when an ant retrieves food, its P_l increases by a constant Δ . Conversely, when an ant spends a lot of time without retrieving food, its P_l decreases by a constant Δ . P_l is bounded in the range $[P_{min}, P_{max}]$. They showed, by means of numerical simulations, that this model can explain task allocation and adaptation to the environment in ants (Labella et al., 2006).

The Variable Delta Rule algorithm (VDR) was based on Deneubourg et al.'s model. The main change was to multiply Δ by the number of consecutive successes or failures when increasing or decreasing the probability of leaving the nest, P_l , to carry out experiments in less time (Labella, 2003; Labella et al., 2006). This simple algorithm might be well suited for use in the context of swarm robotics.

Foraging, in test application for multi-robot systems (MRS), refers to searching for objects and taking them to a place called "nest" (Labella, 2003).

A swarm of interacting robots produces emergent behaviors. We can analyze the local interactions that allow the process of self-organization in these robots using social network analysis. Social network analysis studies the structural properties of groups or individuals in a network. It considers the effect of the interconnections on each other (Srivastava et al., 2014).

We developed an agent-based model based on the Variable Delta Rule algorithm to simulate a swarm of robots inspired by ant foraging behavior. Furthermore, for simplicity we focused on one of the main traits of *Pachycondyla apicalis* ants, that is hunting alone, consequently, there is no need of pheromone trails (Monmarché et al., 2000). Therefore, we modeled the case in which each forager takes only a unit of food when having a successful food retrieval without using pheromone trails. Then, we applied social network analysis to show task allocation as an emergent property of this model.

Methods

In this section, we present the methodology and tools that we used to implement, simulate and analyze the agent-based model of swarm robotics.

Variable Delta Rule Algorithm

We implemented the Variable Delta Rule algorithm (Labella, 2003; Labella et al., 2006). It consists in the following rules: each time the mobile agent has a success in food retrieval, the number of successes is increased and multiplied by Δ , then it is added to its P_l . Conversely, if the mobile agent has a failure in food retrieval, the number of failures is increased and multiplied by Δ , then it is subtracted from its P_l . Therefore, each mobile agent's probability of leaving the nest, P_l , is determined by the number of consecutive successful or failed food retrieval events. Note that P_l is bounded in the range $[P_{min}, P_{max}]$. This is shown in Algorithm 1.

Algorithm 1 Variable Delta Rule

Initialization:

```

successes  $\leftarrow$  0
failures  $\leftarrow$  0
 $P_l \leftarrow$  Initial value

```

if food is retrieved then

```

successes  $\leftarrow$  successes + 1
failures  $\leftarrow$  0
 $P_l \leftarrow P_l + (\text{successes} * \Delta)$ 
if  $P_l > P_{max}$  then
     $P_l \leftarrow P_{max}$ 
end if

```

else if timeout then

```

failures  $\leftarrow$  failures + 1
successes  $\leftarrow$  0
 $P_l \leftarrow P_l - (\text{failures} * \Delta)$ 
if  $P_l < P_{min}$  then
     $P_l \leftarrow P_{min}$ 

```

Agent-based model (ABM) of swarm robotics

Environment. The simulated environment is a bounded two-dimensional grid (when a mobile agent reaches an edge it rotates 180 degrees and continues moving) and has a size of 91 x 91 units, with a unique nest located at the center (cluster of brown patches). A unit of the grid is represented by a patch of 5 x 5 pixels. A unit of food is represented by a unit of the grid located in a food source.

A fixed value in the range [0, 200] is assigned to each unit of the grid as follows: the distance between the focal unit of the grid and the center of the nest is calculated, then it is subtracted from 200 to obtain its “nest scent” value. This value is greater as the focal unit of the grid is closer to the nest. This approach is used by mobile agents to find their way back to the nest, it is known as following “nest scent” and it is described as follows: before each step forward when coming back directly to the nest, the mobile agent is going to head toward the greatest value of “nest scent” that is ahead of it and between the angles -45, 0 or 45. This is repeated until reaching the nest (Wilensky, 1997).

On the grid, food sources are clusters of units of food that are established in a fixed position and have a variable size between small (9 units of food), medium (45 units of food) or large (109 units of food). We have three food sources identified with the following colors, from the closest to the furthest from the nest: magenta, lime and turquoise. Figure 2 shows the distribution and different sizes for food sources in the environment. The environment is dynamic. A food source decreases by one unit of food each time a mobile agent has a successful food retrieval.

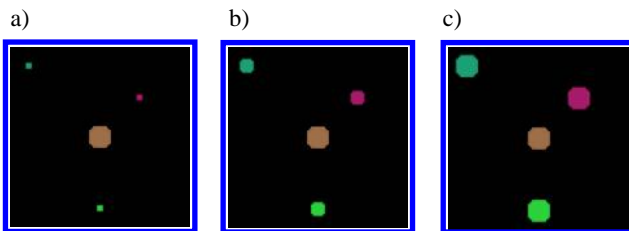


Figure 2: The simulated environments with different sized food clusters: a) Small (9 units of food for each food source), b) Medium (45 units of food for each food source), c) Large (109 units of food for each food source). The nest is in the center of the environment (cluster of brown patches). There are three available food sources, the color of each one indicates the distance to the nest, from the closest to the furthest we have: magenta, lime and turquoise.

Mobile Agents. We consider six mobile agents with initial positions in the center of the nest. Each mobile agent represents a robot. Movements, behaviors and interactions of mobile agents are described as follows:

Movements. Mobile agents have two classes of movements, these are described as follows:

- **Foraging movement:** When a mobile agent is out of the nest, it moves around the environment by random displacements to right and left each time-step, while

considering not to take an occupied unit of the grid where another mobile agent is, as an obstacle avoidance mechanism. A displacement has a maximum turning angle of ± 40 degrees (Wilensky, 1997).

- **Nest seeking movement:** When a mobile agent is returning to the nest, it moves by displacements following the “nest scent” in each time-step. That is, it moves towards the next unit of the grid that has the greatest value of “nest scent” until reaching the nest, while considering not to take an occupied unit of the grid where another mobile agent is, to avoid obstacles.

Behaviors. Each mobile agent assumes one of the following behaviors per time-step depending on its own parameters and environment conditions (Labella et al., 2006):

- **Rest:** Stays in the nest.
- **Search for food:** Explores the environment while checking if there is a unit of food in the path. If there is one, the mobile agent takes it and returns to the nest with food (its number of successes is increased). If there is not one, the mobile agent keeps randomly moving around until a timeout occurs and it returns to the nest without food (its number of failures is increased).
- **Return to nest:** Finds the way back to the nest following the “nest scent” (Wilensky, 1997). It returns to the nest if a unit of food was successfully retrieved or a timeout occurs.
- **Feed:** Transfers food to all the mobile agents in the nest, when arriving to it after a successful food retrieval. Its number of successes is increased by one, therefore its probability of leaving the nest is going to be higher when updating it.

Furthermore, the mobile agents change their color to identify the performed behavior, as shown in Table 1.

Behavior	Color
Rest	Blue
Search for food	Red
Return to nest (with food)	Yellow
Return to nest (without food)	Violet
Feed	Orange

Table 1: Colors representing the behavior of each agent.

Interactions.

- **Agent - Agent (among mobile agents):** When a mobile agent arrives to the nest after retrieving a unit of food, there is an interaction between that mobile agent (emitter) and all the mobile agents in the nest (receivers), which represents food transfer. When a mobile agent is the emitter, its corresponding interaction variables (each one corresponds to an emitter-receiver interaction) increase by one. This is prompted by the forager ant’s interactions with the rest of the colony to feed them. Figure 4 shows an example of interaction among mobile agents.



Figure 4: Interaction between mobile agents. The orange-colored mobile agent (emitter) returned to the nest after retrieving a unit of food, when it arrives to the nest it interacts with all the blue-colored mobile agents (receivers) that are on the cluster of brown patches. This interaction represents food transfer (white arrow) from emitter to receivers and is loosely inspired by a forager ant feeding the rest of the colony in the nest.

- **Agent - Food Source (among mobile agents and food source clusters):** When a mobile agent finds and retrieves a unit of food, there is an interaction between that mobile agent and the retrieved unit of food from a food source, this is inspired by the forager ant’s interactions with a food source. Each time a unit of food is retrieved from a food source, the number of units of food of that food source is decreased and the retrieved unit of food changes to color black to represent it was taken. Figure 5 shows an example of interaction among a mobile agent and a food source.

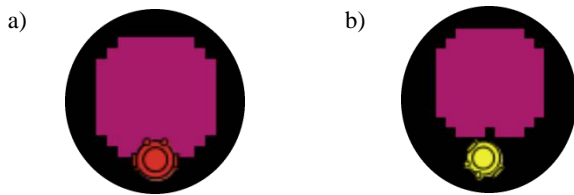


Figure 5: Interaction between a mobile agent and a food source. a) When a red-colored mobile agent finds out a unit of food, it interacts with the food source and b) it changes its color to yellow. The retrieved unit of food changes to color black to represent it was taken.

Experiments

The simulation-based experiments consisted in introducing a swarm of six mobile agents and three food sources (clusters of magenta, lime and turquoise patches), which we varied from small sizes (9 units of food for each food source cluster), medium sizes (45 units of food for each food source cluster) and large sizes (109 units of food for each food source cluster) to show task allocation under changing conditions of the environment. We created 30 instances per food sources size, i.e. 90 simulations in total. Each simulation lasted 2400 time-steps. The model was initialized with the following parameters (Labella, 2003): The search timeout was fixed to 228 units of time, Δ was set to 0.005, P_{min} to 0.0015, P_{max} to 0.05 and P_{mir} to 0.033. Figure 6 shows a representative simulation of the agent-based model of swarm robotics and its components.

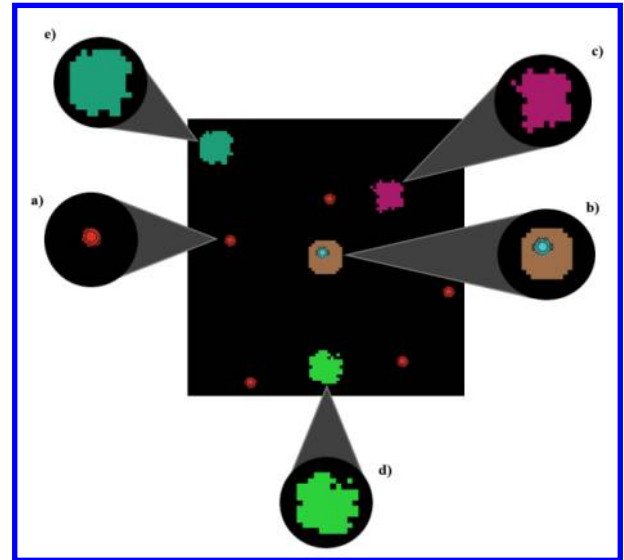


Figure 6: Screenshot of the agent-based model of swarm robotics after 2400 time-steps. a) Red-colored mobile agents are searching for food and b) blue-colored mobile agents are resting in the nest (cluster of brown patches). There are three food sources, from closest to furthest to the nest: c) magenta, d) lime and e) turquoise.

Social network analysis (SNA)

We followed the proposal by Wasserman and Faust (1994), who used network graphs to represent agent structures and network measures such as strength and centrality, to determine the particular role of individuals in the network’s structure. We propose to represent mobile agents and their interactions in each simulation as a weighted directed network and focus on outdegree and weighted outdegree centralities to identify the induced classes, as a result of task allocation: foragers and loafers.

We constructed ninety weighted directed networks, from the 90 simulations, i.e. 30 simulations per food sources size (small, medium and large) as described in the Methods. We added a directed edge between two nodes (source and target) to represent whenever one of the two mobile agents (emitter) interacted with another one in the nest (receiver) to represent food transfer, this is inspired by the forager ants’ interactions with the rest of the colony to feed them. Weights were assigned according to the number of interactions between the two mobile agents. Nodes were labeled with the six mobile agents’ identifiers, from 0 to 5.

Measures were computed for each weighted directed network. We focused on outdegree and weighted outdegree centralities. Degree centrality shows the quality of a network node’s interconnectedness by the number of direct contacts (Landherr et al., 2010). The outdegree is the number of ties that a node directs to others, it is interpreted as a quantity of information that is spread from one node to other (by outgoing edge). A high value is interpreted as sociability (Mansur et al., 2016). The centrality of nodes allows us to identify the most important or central nodes in a network.

Thus, outdegree centrality is a measure of the importance of a node, based on its number of ties. It is interpreted as the involvement of a node in the network. Weighted outdegree centrality is a measure of the importance of a node, based on its strength in terms of the total weight of their connections. It is interpreted as strength of collaborative ties (Opsahl et al., 2010). To calculate node strength, we have the following equation:

$$s_i = \sum_j^N w_{ij} \quad (1)$$

where w is the weighted adjacency matrix and w_{ij} represents the weight of the tie, it is greater than 0 if the node i is connected to node j (Opsahl et al., 2010).

Outdegree centrality can lead us to identify the mobile agents who are the most interconnected to others (i.e. more ties), whereas weighted outdegree centrality can lead us to identify the mobile agents who have the greatest number of interactions (i.e. wider edges) with the rest. Hence, we need both centrality measures to identify the expected induced classes, as a result of task allocation: foragers and loafers. Foragers' task consists in searching and retrieving food to feed the rest of the mobile agents; loafers' task consists in staying in the nest. Thus, foragers must be the most interconnected to the rest (i.e. having more ties) and with the greatest number of interactions (i.e. having wider edges). In the shown networks, node size refers to the value of outdegree or weighted outdegree centralities.

According to Labella's (2003) experimental results with *MindS-bots* (swarm of robots), he found that the distribution of probability of leaving the nest had two peaks and the boundary between the two groups was around 0.025, therefore there were two groups of *MindS-bots*: foragers ($P_l \geq 0.025$) and loafers ($P_l < 0.025$). As described in the Experiments, our model was initialized under Labella's (2003) experimental parameters, thus, we compared the results with the second parameter to identify foragers and loafers: mean probability of leaving the nest (mean P_l). Therefore, those mobile agents with mean $P_l \geq 0.025$ are likely to be foragers (red-colored nodes) and those with mean $P_l < 0.025$ are likely to be loafers (blue-colored nodes).

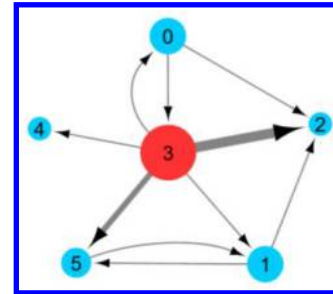
Results

First, we show the results for three representative simulations (each one with a different food sources size). Then, we show in summary the results for the 90 simulations.

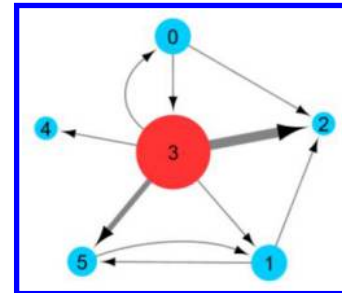
Simulation 1 - Small food sources size

Figure 7 shows the weighted directed network obtained with the results of simulation 1. The mean outdegree centrality of this network was 1.83, that indicates there were few nodes that were the most interconnected to others, in this case, only node 3 had ties to all the other nodes. The mean weighted outdegree centrality was 2.17, that indicates there were few interactions between mobile agents. There were two edges with high weight values, those were (3,2) and (3,5), which represented the greatest number of interactions between the

mobile agents. Node 3 had the greatest number of ties and wider edges, moreover, it has a $P_l > 0.025$, therefore we interpreted it as a forager. The mean probability of leaving the nest of all nodes was 0.021, which was less than 0.025, so we expected more loafers than foragers. Likely agents to be foragers by P_l were represented by red-colored nodes and likely agents to be loafers by P_l were represented by blue-colored nodes. After analyzing the results, we got 1 forager (node 3) and 5 loafers (nodes 0, 1, 2, 4, 5).



(a) Social network 1 with nodes sized by their outdegree centrality



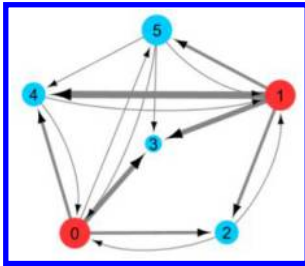
(b) Social network 1 with nodes sized by their weighted outdegree centrality

Figure 7: Graphs of social network 1 (the size of food sources is small) between six mobile agents where node sizes are reflecting: (a) Outdegree centrality, (b) Weighted outdegree centrality. Edge widths are reflecting the number of interactions between mobile agents. Node colors represent the probability of leaving the nest: if $P_l \geq 0.025$ the node is red, therefore, it is likely to be a forager and if $P_l < 0.025$ the node is blue, therefore, it is likely to be a loafer. As it can be seen, node color and size are consistent with each other, that means bigger nodes and probability to be a forager coincide; similarly, smaller nodes and probability to be a loafer also coincide. Therefore, both approaches obtain same results (in this case, 1 forager and 5 loafers).

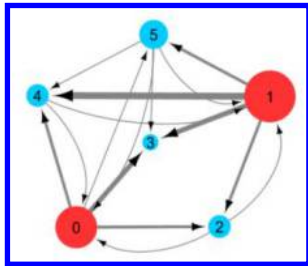
Simulation 2 - Medium food sources size

Figure 8 shows the weighted directed network obtained with the results of simulation 2. The mean outdegree centrality of this network was 2.67, that indicates there was a moderate number of nodes that were the most interconnected to others, more than in Simulation 1. The mean weighted outdegree centrality was 4.5, that indicates there was a greater number of interactions between mobile agents than in Simulation 1.

There were seven edges with high weight values, those were (1,5), (1,4), (1,3), (1,2), (0,4), (0,3) and (0,2) which represented the greatest number of interactions between agents. Nodes 0 and 1 had the greatest number of ties and wider edges, moreover, their $P_l > 0.025$, therefore we interpreted them as foragers. The mean probability of leaving the nest of all nodes was 0.022, which was less than 0.025, so we expected more loafers than foragers. After analyzing the results, we got 2 foragers (nodes 0, 1) and 4 loafers (nodes 2, 3, 4, 5).



(a) Social network 2 with nodes sized by their outdegree centrality



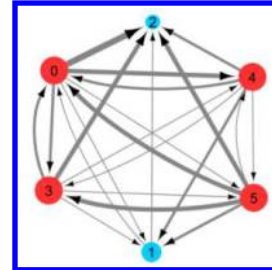
(b) Social network 2 with nodes sized by their weighted outdegree centrality

Figure 8: Graphs of social network 2 (the size of food sources is medium) between six mobile agents where node sizes are reflecting: (a) Outdegree centrality, (b) Weighted outdegree centrality. Edge widths are reflecting the number of interactions between mobile agents. Node colors represent the probability of leaving the nest: if $P_l \geq 0.025$ the node is red; therefore, it is likely to be a forager and if $P_l < 0.025$ the node is blue; therefore, it is likely to be a loafer. As it can be seen, node color and size are consistent with each other, that means bigger nodes and probability to be a forager coincide; similarly, smaller nodes and probability to be a loafer also coincide. Therefore, both approaches obtain same results (in this case, 2 foragers and 4 loafers).

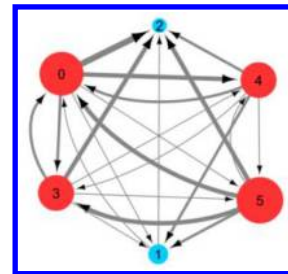
Simulation 3 - Large food sources size

Figure 9 shows the weighted directed network obtained with the results of simulation 3. The mean outdegree centrality of this network was 3.6, that indicates there were many nodes that were the most interconnected to others, more than in Simulations 1 and 2. The mean weighted outdegree centrality was 6.83, that indicates there was a greater number of interactions between mobile agents than in Simulations 1 and 2. There were many edges with high weight values, due to

high food availability. Nodes 0, 3, 4 and 5 had the greatest number of ties and wider edges, moreover, their $P_l > 0.025$, therefore we interpreted them as foragers. The mean probability of leaving the nest of all nodes was 0.029, which was greater than 0.025, so we expected more foragers than loafers. After analyzing the results, we got 4 foragers (nodes 0, 3, 4, 5) and 2 loafers (nodes 1, 2).



(a) Social network 3 with nodes sized by their outdegree centrality



(b) Social network 3 with nodes sized by their weighted outdegree centrality

Figure 9: Graphs of social network 3 (the size of food sources is large) between six mobile agents where node sizes are reflecting: (a) Outdegree centrality, (b) Weighted outdegree centrality. Edge widths are reflecting the number of interactions between mobile agents. Node colors represent the probability of leaving the nest: if $P_l \geq 0.025$ the node is red; therefore, it is likely to be a forager and if $P_l < 0.025$ the node is blue; therefore, it is likely to be a loafer. As it can be seen, node color and size are consistent with each other, that means bigger nodes and probability to be a forager coincide; similarly, smaller nodes and probability to be a loafer also coincide. Therefore, both approaches obtain same results (in this case, 4 foragers and 2 loafers).

Summary of results

The results of the social network analysis applied to the 90 weighted directed networks obtained from the simulation experiments are summarized in Table 2. It reports the mean and standard deviation of number of foragers and loafers.

Figure 10 shows the results of mean and standard deviation of probability of leaving the nest of the six mobile agents in the 30 experiments per food sources size (i.e. 90 experiments in total).

Contrasting the results of Table 2 and Figure 10, we can see that the social network analysis results confirmed the

expectations of number of loafers and foragers obtained by the mean probability of leaving the nest varying the food sources size. Hence, the results proved task allocation among mobile agents as an emergent property of this model, inducing two classes: foragers and loafers. The number of foragers and loafers was adapted to the environment conditions (in this case, food availability).

Food Sources Size	Food availability	Number of Foragers	Number of Loafers
Small	Low	1.1 ± 0.3051	4.9 ± 0.3051
Medium	Medium	3.07 ± 0.7397	2.93 ± 0.7397
Large	High	4.77 ± 0.4302	1.23 ± 0.4302

Table 2: Mean and standard deviation of number of foragers and loafers calculated over 30 simulations per food sources size (i.e. 90 simulations in total) by applying social network analysis to the obtained weighted directed networks. The low values of standard deviation indicate that the behavior of the model was consistent across simulations.

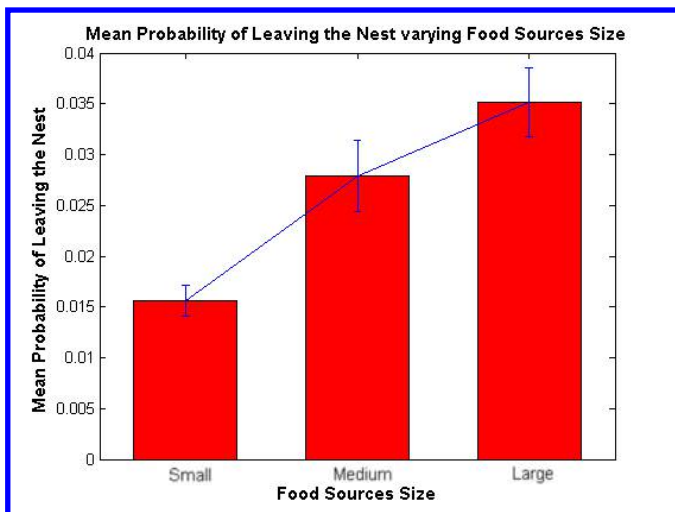


Figure 10: Mean and standard deviation of probability of leaving the nest while varying food sources size. The low standard deviation indicates that the behavior of the model was consistent across simulation experiments. These results show that with a low availability of food (small sized food sources) the mean $P_l < 0.025$, therefore, we expected more loafers than foragers; with a medium availability of food (medium sized food sources) the mean P_l is a little above 0.025, therefore, we expected similar number of loafers and foragers; with a high availability of food (large sized food sources) the mean $P_l > 0.025$, therefore, we expected more foragers than loafers.

Emergent property - Task allocation

In all simulations we observed that task allocation emerged and induced two classes: foragers and loafers. There were

more loafers than foragers with low food availability (i.e. small sized food sources); there was similar number of loafers and foragers with medium food availability (i.e. medium sized food sources); and there were more foragers than loafers with high food availability (i.e. large sized food sources).

Discussion

As we have seen, moving from agent-based modeling (ABM) to social network analysis (SNA) lead us to a better understanding of the complex system by studying its emergent properties. In our agent-based model of swarm robotics we have shown that task allocation emerges and induces the creation of two classes: foragers and loafers. Furthermore, one of our main results was that the number of foragers and loafers changed with the conditions of the environment, as in real ant colonies. It means, task allocation changes as conditions vary (Gordon, 1999). Our model highlights that when more food is available, more foragers appear, and vice versa, as we observed in the weighted directed networks that we created for each simulation results. Thus, we conclude task allocation implies an adaptive and self-organized process (Labella, 2003).

A distinctive property revealed by the social network analysis was that the nodes with the greatest outdegree centralities were the most interconnected with the others (i.e. more ties) and those with the greatest weighted outdegree centralities had wider edges, therefore those nodes which were bigger in both graphs were the most interconnected mobile agents (i.e. having more ties) with the greatest number of interactions (i.e. having wider edges), hence we can call them, the “influentials” in the colony. These are the foragers.

Conclusions and Future Work

To summarize, we presented and analyzed our agent-based model of swarm robotics using social network analysis to show that it exhibits task allocation as an emergent property due to the Variable Delta Rule algorithm (Labella, 2003; Labella et al., 2006), which was inspired by ant foraging behavior. In future work, we are going to explore more complicated scenarios, for example, considering cheaters, those social insects that exploit the benefits of biological cooperation without contributing to them (Dobata and Tsuji, 2009). Moreover, this can be extended by using social network analysis to develop agent-based models, that is, moving from social networks to multi-agent systems in order to establish the measures of those networks and then design agent’s behaviors that will reach those measures. This could potentially be used in order to run game theoretic (network) models in an agent-based modeling framework.

References

- Alers, S., Tuyls, K., Ranjbar-Sahraei, B., Claes, D. and Weiss, G. (2014). Insect-Inspired Robot Coordination: Foraging and Coverage. In Sayama, H., Rieffel, J., Risi, S., Doursat, R. and Lipson, H., editors, *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14)*, pages 761-768. MIT Press, Cambridge, MA.

- Charbonneau, D., Blonder, B. and Dornhaus, A. (2013). Social Insects: A Model System for Network Dynamics. In: Holme P. and Saramäki J., editors, *Temporal Networks*, pages 217-244. Springer, Berlin.
- Deneubourg, J.-L., Goss, S., Pasteels, J. M., Fresneau, D. and Lachaud, J.-P. (1987). Self-organization mechanisms in ant societies (II): Learning in foraging and division of labor. In Pasteels, J. and Deneubourg, J.-L., editors, *From Individual to Collective Behavior in Social Insects*, Experientia Supplementum, 54:177-196. Birkhäuser Verlag, Basel, Switzerland.
- Dobata, S. and Tsuji, K. (2009). A cheater lineage in a social insect: Implications for the evolution of cooperation in the wild. *Communicative & Integrative Biology*, 2(2):67–70.
- Fontana, M. and Terna, P. (2015). From Agent-Based Models to Network Analysis (and Return): The Policy-Making Perspective. *Journal on Policy and Complex Systems*, 2(1):77-92.
- Gordon, D. M. (1999). Interaction patterns and task allocation in ant colonies. In Detrain, C., Deneubourg, J.-L., Pasteels, J. M., editors, *Information Processing in Social Insects*, pages 51-67. Birkhäuser Verlag, Basel, Switzerland.
- Gordon, D. M. (2016). From division of labor to the collective behavior of social insects. *Behavioral Ecology and Sociobiology*, 70(7):1101–1108.
- Grant, T. (2009). Modelling Network-Enabled C2 using Multiple Agents and Social Networks. In Andrighetto, G., Boella, G., Sichman, J. and Verhagen, H., editors, *Proceedings of the Social Networks and Multi-Agent Systems Symposium (SNAMAS-09)*, pages 13-18. The Society for the Study of Artificial Intelligence and the Simulation of Behavior (SSAISB).
- Greenwald, E., Baltiansky, L., and Feinerman, O. (2018). Individual crop loads provide local control for collective food intake in ant colonies. *eLife*, 7: e31730. doi:10.7554/eLife.31730
- Guo, Y. and Wilensky, U. (2016). Small Bugs, Big Ideas: Teaching Complex Systems Principles Through Agent-Based Models of Social Insects. In Gershenson, C., Froese, T., Siqueiros, J. M., Aguilar, W., Izquierdo, E. J. and Sayama, H., editors, *Proceedings of the Artificial Life Conference 2016*, pages 664-665. MIT Press, Cambridge, MA.
- Iba, H. (2013). *Agent-Based Modeling and Simulation with Swarm*. Chapman and Hall/CRC, New York.
- Koval, M. C., Rubinoff, A. E. S., Maghami, M. and Georgiopoulos, M. (2009). Social Network Analysis for Target Recognition in Swarm Robotics. *AMALTHEA REU program*.
- Krause, J., James, R., Franks, D. W. and Croft, D. P. editors (2015). *Animal Social Networks*. Oxford University Press.
- Labella, T. H. (2003). *Prey Retrieval by a Swarm of Robots*. Thesis for the Diplôme d'Études Approfondies (DEA), Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles.
- Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of Labour in a Group of Robots Inspired by Ants' Foraging Behaviour. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):4-25.
- Landherr, A., Friedl, B. and Heidemann, J. (2010). A Critical Review of Centrality Measures in Social Networks. *Business & Information Systems Engineering*, 2(6):371-385.
- Ma, J., Guo, D., Wang, K., Liu, M. and Chen, S. (2009). Colony Evolution in Social Networks Based on Multi-agent System. In Wang, H., Low, K. S., Wei, K. and Sun, J., editors, *Fifth International Conference on Natural Computation (ICNC)*, Vol. 4, pages 594-597. IEEE Computer Society.
- Mansur, A. B. F., Yusof, N. and Basori, A. H. (2016). The Analysis of Student Collaborative Work Inside Social Learning Network Analysis Based on Degree and Eigenvector Centrality. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(5):2488-2498.
- Monmarché, N., Venturini, G. and Slimane, M. (2000). On how *Pachycondyla apicalis* ants suggest a new search algorithm. *Future Generation Computer Systems*, 16(8):937-946.
- Opsahl, T., Agneessens, F. and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245-251.
- Palestra, G., Paziienza, A., Ferilli, S., De Carolis, B. and Esposito, F. (2017). RescueRobot: Simulating Complex Robots Behaviors in Emergency Situations. In Anzalone, S., Farinelli, A., Finzi, A. and Mastrogiovanni, F., editors, *Proceedings of the 4th Italian Workshop on Artificial Intelligence and Robotics (AIRO 2017)*, pages 65-69. CEUR Workshop Proceedings.
- Powers, S. T., Ekárt, A. and Lewis, P. R. (2018). Co-creating Enduring Institutions for Socio-Technical Systems: The Complementarity of Content-based and Value-based Modelling Approaches. In Ikegami, T., Virgo, N., Witkowski, O., Oka, M. Suzuki, R. and Iizuka, H., editors, *Proceedings of the 2018 Conference on Artificial Life (ALIFE 2018)*, pages 105-106. MIT Press, Cambridge, MA.
- Ramos-Fernández, G., Boyer, D. and Gómez, V. P. (2006). A complex social structure with fission–fusion properties can emerge from a simple foraging model. *Behavioral Ecology and Sociobiology*, 60:536-549. doi:10.1007/s00265-006-0197-x
- Ramos-Fernández, G., Boyer, D., Aureli, F. and Vick, L. G., (2009). Association networks in spider monkeys (*Ateles geoffroyi*). *Behavioral Ecology and Sociobiology*, 63(7):999-1013.
- Rutz, C., Burns, Z. T., James, R., Ismar, S. M. H., Burt, J., Otis, B., Bowen, J. and St Clair, J. J. H. (2012). Automated mapping of social networks in wild birds. *Current Biology*, 22(17):R669-R671. doi:10.1016/j.cub.2012.06.037
- Scott, J. and Carrington, P. J. (2014). *The SAGE Handbook of Social Network Analysis*. SAGE Publications Ltd, London.
- Srivastava, A., Anuradha and Gupta, D. J. (2014). Social Network Analysis: Hardly Easy. In *Proceedings of the 2014 International Conference on Reliability, Optimization and Information Technology (ICROIT)*, pages 128-135. IEEE.
- Tan, Y. and Zheng, Z. (2013). Research Advance in Swarm Robotics. *Defence Technology*, 9(1):18-39.
- Wang, Y., Rattray, J. B., Thomas, S. A., Gurney, J. and Brown, S. P. (2019). In *silico* bacteria evolve robust cooperation via complex quorum-sensing strategies. *bioRxiv* 598508. doi:10.1101/598508
- Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications (Vol. 8)*. Cambridge University Press, Cambridge.
- Wilensky, U. (1997). NetLogo Ants model. <http://ccl.northwestern.edu/netlogo/models/Ants>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

On information-optimal scripting of actions

Bente Riegler and Daniel Polani

Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire
b.reichardt@herts.ac.uk d.polani@herts.ac.uk

Abstract

Animals and humans encounter many tasks which permit ritualized behaviours, essentially fixed action sequences or “scripts”, similar to options known from Reinforcement Learning, but proceeding without intermediate decisions. While running a script, they proceed in an open-loop fashion. However even when these are already known, an agent needs to decide whether to perform a basic action or to trigger a script regarding the particular task. Here we study if including such scripts (i.e. behaviour rituals) is advantageous from the point of view of the relevant information required to take the decision to start such a script depending on the tasks. To achieve this, we modify the relevant information framework including sequences of basic actions to the possible actions.

Introduction

Many tasks animals or humans encounter are composed of multiple smaller steps. An agent has typically learned such sequences through repeated solution of the task over time. Such “ritualized” behaviour sequences do not require high-level decision-making for every small step, but may permit solutions where fixed action sequences (ritualized behaviours) are triggered. Whenever this is possible, this leads to informationally significantly cheaper control, because fewer decisions need to be made — only ever when a new script is triggered, while it is running, it operates as an open loop controller. In contrast, if only basic actions are available, a decision may be required in every time step. This is a special, but important, case of the more sophisticated option framework (Van Dijk et al., 2009). The exact script to be triggered depends on the specific task and requires information about the current state of the agent. We ask how much *relevant information* (Shannon information about state required to select an action) is required when scripts - sequences of basic actions - can be used in addition to basic actions. Furthermore, we ask whether scripts make some goal states informationally easier to reach than others.

Perception-Action Loop

The perception-action loop setup for our agent is very similar to Reinforcement Learning. In each state, the agent per-

forms actions and as a result, its state changes. This is modelled as a Markov Decision Process (MDP). A set of states $s \in \mathcal{S}$ models the agent’s position in the world and in each the agent can choose one action $a \in \mathcal{A}$. For the current experiment we assume the individual transitions $p(s_{t_2} | s_{t_1}, a)$, with t_1 and t_2 being the time before and after the action, to be deterministic. In our work, the agent does not have to freshly decide its next action after every single primitive, but can select an action script instead of a primitive, modelled as an MDP with enhanced action set.

The World

The world consists of the set of all states. Here, we consider a small grid world of 5×5 states, one start-state, at least one goal-state and the set of basic actions $\mathcal{A} = \{\textit{north}, \textit{east}, \textit{south}, \textit{west}\}$, with respect to the global directions. The agent carries no internal orientation and is always globally oriented. Every executed basic action incurs a cost of 1. There is no discount over time. We assume there exists a goal which can be any subset of \mathcal{S} . Goal states are modelled as absorbing states in the MDP, i.e. all actions taken in a goal state leave the agent where it is, and do not add further cost. On reaching the goal, the currently executed script is effectively interrupted. The grid is finite and has “walls”, an action that pushes the agent into the wall leaves the agent unchanged and still incurs the usual cost of 1. Since here we only consider optimal policies, no agent will waste effort walking into walls.

The Action Space Extended by Scripts

The main novelty compared to previous studies (Polani et al., 2006) is the action space. To the set \mathcal{A}_b of basic actions, we add a set of scripts. These scripts are a sequential unconditional (open-loop) combination of the basic actions available in the world. Thus, our new action-space consists of all concatenations of at least one basic action \mathcal{A}_b^+ ; in our setting, we assume a maximum length of scripts, and thus a finite selection of possible (basic or composite) actions. In this work, we assume the agent has already learnt all possible actions. The cost of an action is modelled in two slightly different ways: first, a cost of 1 per every basic action in the

script except for actions after reaching the goal; and second, the same, but with an added cost of 1 for taking a decision (note, this MDP cost is *not* informational in the present experiments). The decision cost is a cost only occurring at decision points. The value of 1 is arbitrarily chosen.

Relevant Information

Relevant information for an MDP is defined as the minimal information required about the current state to select an action to achieve a given utility (or, equivalently, in our case, minimal cost, see Polani et al., 2006):

$$\min_{\pi(A|S)} I(S; A). \quad \text{The relevant information is calculated in two steps. Firstly, precalculate the perfect utility of an agent with respect to the given goal states with a value iteration algorithm. Costs of all actions and scripts are accumulated during a single run, until the goal is reached.}$$

Secondly, the relevant information is computed based on this utility. For this, we use the classic Blahut-Arimoto-algorithm from rate-distortion theory. From these results we calculate the policy and identify which actions or scripts are used in which state.

Experiments

In the experiments, the agent may start in any state. And we examine different classes of goal states. We consider the following: **Northern Border:** This goal is composed of all northernmost states of the world. With only basic actions available, the relevant information is zero. In all states the best action is to go *north*. Expanding the action space with scripts of any length changes the optimal policy so that all scripts containing just the basic action *north* are equally probable in all states. Thus, the relevant information stays zero. When a cost for choosing an action (i.e. decision cost) is added, the longest script possible is preferred because it requires fewer decision points. **Central State:** Only the central state of the grid is a goal state. To reach the goal requires different actions from different states. This results in a high relevant information of 0.1 bit per decision without decision cost and 0.4 bits with. This goal leads to a high relevant information and favours all shorter actions over longer scripts. This does not change after adding a decision cost. **Centre Line:** Here a whole line running through the center is set as goal. This setup falls in between the previous ones. It shares the neighbouring goal-states from the first setup with the centre-character of the second setup. Thus, the result should be in between as well. The relevant information turns out to be roughly 0.08 bits for the centre line setup. Shorter actions are preferred over long ones. When a cost for the decision is added, longer scripts are favoured. **Corner State:** Here, the goal is one state in one of the corners of the world. For this setup, we expect a reduced but nonzero amount of relevant information. Indeed, the rele-

vant information becomes about 0.01 bits. Without a decision cost, this setup favours shorter movements. When a decision cost is added, the scripts modelling diagonal movement are favoured in many states, but the relevant information increases to 0.13.

Discussion and Future Work

We find that the main value of scripts is to avoid re-deciding on what to do while they run, since the scripts are favoured when we assume a decision cost. Note, there is no profound justification for the value of the decision cost for now.

The experiments show a use of scripts for the northern border and the central line goal areas, while the setups with single goal states keep using the basic actions. Thus, scripts are useful for wider goal areas but not when specific states need to be reached. The wider goals represent generic tasks, such as extending the body to reach “as high as possible” for which the northern border goal setup is an abstract model, or “somewhere back there” represented abstractly by both the border and the centre line setup.

Note, that, strictly spoken, despite our present assumptions, behavioural scripts in actual organisms may require low-latency feedback and are not necessarily fully open-loop. So, more strictly, one would have to associate some processing cost also to run the scripts. However, our present paper focuses only on the high-level information required to select and activate the scripts.

In future, it will thus be important to also quantify the trade-off between memorizing, processing the low-level script and and saving high-level relevant information. Ultimately, this relates to the question of how hierarchies should be found and organized (Larsson et al., 2017) and how expensive learning itself is.

Acknowledgements

DP wishes to thank Olaf Witkowski for inspiring conversations that led to this study.

References

- Larsson, D. T., Braun, D., and Tsiotras, P. (2017). Hierarchical state abstractions for decision-making problems with computational constraints. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1138–1143. IEEE.
- Polani, D., Nehaniv, C. L., Martinetz, T., and Kim, J. T. (2006). Relevant information in optimized persistence vs. progeny strategies. In *In: Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*. Mit Press.
- Van Dijk, S. G., Polani, D., and Nehaniv, C. L. (2009). Hierarchical behaviours: getting the most bang for your bit. In *European Conference on Artificial Life*, pages 342–349. Springer.

Modeling Fast and Robust Ant Nest Relocation using Particle Swarm Optimization

Hideyasu Sasaki

National Institute of Information and Communications Technology, Tokyo 184-8795, Japan
hsasaki@nict.go.jp

Abstract

Ant nest relocation is smoother and swifter than the same process undertaken by any other animal. Within the population of ants, the ratio that participates in nest relocation is only 58.0% at best and 31.0% at worst. Does such a low active ratio improve or deteriorate ant nest relocation? In this study, we use a particle swarm optimization (PSO) algorithm to simulate real-world ant nest relocation. Our PSO-based algorithm duplicates the velocity and position of an inactive particle (representing an inactive ant) with the velocity and position of an active particle (representing an active ant). The number of particles that the algorithm computes is dramatically reduced, and the global best position can be identified at an early stage. In a series of simulations, our algorithm performs significantly better and faster with active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95% than with the full 100% active ratio. We confirm the robust and stable performance of our algorithm at active ratios of 60%, 80%, and 85%. Clustering of the simulation results shows that low active ratios improve ant nest relocation. Furthermore, three field studies carried out by biology experts empirically demonstrate that we have successfully modeled and simulated real-world ant nest relocation using our PSO-based algorithm.

Introduction

Ants have a reputation for being hardworking animals. For instance, worker ants can relocate an entire nest within a short period of time (Burd et al., 2002). To the best of our knowledge, no other animals can compete with ants in terms of the speed and smoothness of the nest relocation process.

Real-world problems involve multiple objectives that are often in conflict with one another, yet should be optimized simultaneously (Ali et al., 2016a,b; Awad et al., 2013, 2016, 2017). The impressive speed and precise movement of ant nest relocation provides an intuitive solution for multi-objective optimization problems.

The swarm behavior of ant nest relocation has attracted researchers in the computational intelligence community. Various models and simulations have been proposed for real-world ant nest relocation (Chowdhury et al., 2002; John et al., 2008, 2009; Sasaki and Leung, 2013; Sasaki, 2017). However, ant nest relocation has not yet been studied as a

multi-objective optimization problem in terms of the precision and speed of the swarm behavior.

Over the last decade, biologists have reported that many workers of the *Temnothorax* ant species remain inactive and do not work at all (Dornhaus et al., 2008; Charbonneau et al., 2017). Inactive worker ants lick their own bodies, and active worker ants feed and relocate them. During ant nest relocation, active workers shoulder these inactive workers on their backs and move them toward the new nest (Dornhaus et al., 2008). Among *Temnothorax* ant species, behavioral biologists have rigorously studied *Temnothorax albipennis*, also known as the rock ant, for its “scale-free” swarm behavior of nest relocation. Regarding the *Temnothorax albipennis* ant, the active ratio within the populations of worker ants in a colony (hereafter referred to as the *active ratio*) is 58.0% at best and 31.0% at worst. Variations in the active ratio do not depend on the size of the population of worker ants in the colony (Dornhaus et al., 2008). Inactive workers act as a reserve labor force for replacing active workers, however, active workers do not replace inactive workers that have been removed from the nest Charbonneau et al. (2017).

An open problem on the scale-free swarm behavior of the *Temnothorax albipennis* ant is whether such a low active ratio improves or deteriorates ant nest relocation compared with the full 100% active ratio performance (Sasaki, 2018a,b, 2019). Although researchers have proposed models and simulations that accurately describe ant nest relocation, they have not solved this problem. A positive answer to this problem would provide technological inspiration for promising swarm-based algorithms in the context of computational intelligence and in specific aspects of swarm intelligence regarding the active ratio within the populations of computational agents.

In previous work, we have shown that the frequency of mutual contact among worker ants determines the active ratio in ant nest relocation (Sasaki and Leung, 2013; Sasaki, 2017). The frequency of mutual contact rises from 1 to 40 times per minute, and the active ratio gradually rises to the maximum. In this study, we use a particle swarm optimization (PSO) algorithm to model and simulate real-world ant

nest relocation. In particular, we investigate the relation between the active ratio and the performance of worker ants in nest relocation. The simulation results show that our PSO-based algorithm performs significantly better and faster at active ratios that are lower than the full 100% active ratio. We confirm the robust and stable performance of our algorithm at certain low active ratios. The simulation results are supported by three field studies that were carried out by expert ant biologists.

Model

To model ant nest relocation while focusing on the active ratio, we employ a population-based optimization method, namely the original or canonical PSO algorithm developed by Kennedy and Eberhart (1995), and refer to several popular PSO variants (Clerc and Kennedy, 2002; Cleghorn and Engelbrecht, 2018).

The swarm behavior of ant nest relocation can be described using the velocity and position of an ant (John et al., 2008, 2009). Worker ants find the shortest path toward a new nest. The shortest path is found when ants move from an old nest to a new nest and when the distance of nest relocation is minimized. The PSO algorithm describes the velocity and position of particles. There are various optimization methods, and the PSO algorithm and its variants are some of the best techniques for modeling the swarm behavior of ant nest relocation, as they can easily focus on the velocity and position of an ant.

We model the swarm behavior of ant nest relocation using the velocity and position of a particle with two types of “best” positions. Our PSO-based algorithm is initialized with a group of random particles and searches for optima by updating the velocity and position of a particle in each iteration of the algorithm. At every iteration, the velocity and position of a particle are updated in comparison with the two types of best positions, which are evaluated with a fitness function called a “cost function”. One best position is the “personal best position” of a particle, and the other is the “global best position” within the swarm of particles. After finding the two best positions, the velocity and position of a particle are updated at every iteration (Pereira, 2010).

First, our PSO-based algorithm minimizes the output of the cost function with the velocity and position of an “active” particle. The output of the cost function is equivalent to the distance that an “active” worker ant moves from an old nest to a new nest. An active particle that represents an active worker ant is defined and updated in the same way as in the original PSO algorithm developed by Kennedy and Eberhart (1995). The velocity $v \in R$ and position $x \in R$ of active particle $i \in N$ in a population of size S at a discrete time step $t \in N$, which is the t -th number of the iteration process, are defined and updated, respectively, as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t) \{p_i(t) - x_i(t)\}$$

$$+ c_2 r_2(t) \{g(t) - x_i(t)\}, \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

where the random components $r_{1,k}(t)$ and $r_{2,k}(t) \sim U(0, 1)$, k is the vector component of v_i and x_i , and the coefficients c_1 , c_2 , and ω are the personal (cognitive), social, and inertia weights, respectively. The position p represents the personal “best” position that particle i has visited, where “best” means the location where the particle had obtained the lowest cost function evaluation. The position g represents the global “best” position that the particles in the neighborhood of the i -th particle have visited.

The vector component k of the velocity v and position x of particle i represents the direction of movement of the particle. A particle represents a worker ant that moves on the ground. The direction of movement of a particle is defined in a discrete form that consists of eight two-dimensional directions, i.e., $K = 8$, as shown in Fig. 1. An ant not only moves forward and backward, but also shifts in the transverse direction and moves and shifts at the same time. Therefore, we define the direction of movement of a particle that represents a worker ant in a discrete form that consists of eight two-dimensional directions.

The personal best position p of active particle i and the global best position g at time t (or the t -th iteration) are defined and updated with the cost function f as follows:

$$p_i(t+1) = \begin{cases} x_i(t+1) & \text{if } f(x_i(t+1)) < f(p_i(t)) \\ p_i(t) & \text{else,} \end{cases} \quad (3)$$

$$g(t+1) = \arg \min_{p_i(t) \in P(t)} f(p_i(t+1)), \quad (4)$$

where the cost function f takes the form

$$f(x_i^k(t)) = \sum_{i \in S} \sum_{k \in K} \|x_i^k(t)\|^2, \quad (5)$$

where $\|\cdot\|$ denotes the Euclidean norm.

In the modeling process, we choose the sphere function for the cost function f . This simple function provides a straightforward model that allows us to assess the benefits and limitations of our PSO-based algorithm in simulating ant nest relocation. An ant moves around the two-dimensional ground. There are many functions that could

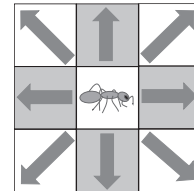


Figure 1: Particle representing a worker ant has eight two-dimensional directions of movement on the ground.

calculate such two-dimensional distances over which the particles move. Almost all field studies carried out by experts in ant biology simply use the Euclidean distance that the ants relocated a nest (Burd et al., 2002; Pratt, 2005b; Franks et al., 2006; Dornhaus et al., 2008, 2009; Charbonneau et al., 2017). Researchers in the computational intelligence community tend to have followed the biological practice of field researchers (Chowdhury et al., 2002; John et al., 2008, 2009; Sasaki and Leung, 2013; Sasaki, 2017, 2018a,b, 2019). We use the sphere function given by (5) through the index i and vector component k for the cost function f , though our PSO-based algorithm could easily use other functions (as will be considered in future work). The cost function f represents the distance between the position x of particle i (representing a worker ant) and the global best position g that leads to the shortest path toward point zero (representing a new nest as the target for movement). The distance that a particle moves can take a positive value or a negative value from point zero, so the square of the difference is computed. The accumulated square of the difference at the position of a particle should be minimized through vector component k .

Next, we define the swarm behavior of “inactive” worker ants in modeling ant nest relocation. An inactive worker ant does not relocate the nest by itself. Instead, active worker ants carry the inactive ants toward the new nest. In our PSO-based algorithm, the velocity v and position x of “inactive” particle i (representing an inactive worker ant) at time t (or the t -th iteration) are defined and updated, respectively, as follows:

$$v_i(t+1) = v_{i-1}(t+1), \quad (6)$$

$$x_i(t+1) = x_{i-1}(t+1) = x_{i-1}(t) + v_i(t+1). \quad (7)$$

The personal best position p of inactive particle i and the global best position g at time t (or the t -th iteration) are defined and updated with the cost function f given by (5), respectively, as follows:

$$p_i(t+1) = p_{i-1}(t+1), \quad (8)$$

$$g(t+1) = p_i(t+1). \quad (9)$$

Fig. 2 shows that our PSO-based algorithm duplicates the velocity and position of inactive particle i with the velocity and position of active particle $i-1$ that is closest to inactive particle i . This duplication of the velocity and position of inactive particles reduces the number of particles that the algorithm computes, allowing the global best position to be identified at an early stage of the iteration.

Simulation

Setting

Our PSO-based algorithm comprises the coefficients c_1 , c_2 , and ω that are the personal, social, and inertia weights. The

parameter configurations follow the earlier work of Clerc and Kennedy (2002) and refer to the Yarpiz Project (2015). Cleghorn and Engelbrecht (2018, 2016) have developed a sophisticated theory for determining parameter configurations and enhanced the stability criteria of particles. However, we use the conservative approach of the original PSO algorithm in our simulations. The memory of the *Temnothorax albipennis* ant lasts for only 1–2 min (Pratt, 2005a). It is reasonable to assume that the parameter configurations for nest relocation are genetically coded in the brains of the ants in a static, rather than dynamic, form. The sophisticated parameter configurations developed by Cleghorn and Engelbrecht (2018, 2016) might lead to overfitting in modeling the swarm behavior of ant nest relocation. Therefore, we follow the parameter configurations of the original PSO algorithm given by:

$$c_1 = \chi\varphi_1, c_2 = \chi\varphi_2,$$

$$\omega = \chi = 2\kappa/|2 - \varphi - \sqrt{(\varphi^2 - 4\varphi)}|, \quad (10)$$

where $\kappa = 1$, $\varphi = \varphi_1 + \varphi_2$, and $\varphi_1 = \varphi_2 = 2.05$.

A lower limit v_{min} and an upper limit v_{max} of the velocity v are assigned through index i by:

$$v_{min} = -v_{max} \text{ and } v_{max} = 0.2(x_{max} - x_{min}). \quad (11)$$

Similarly, a lower bound x_{min} and an upper bound x_{max} of the position x are assigned through index i by:

$$[x_{min}, x_{max}] = [-10, 10]. \quad (12)$$

The lower and upper bounds are applied to the personal best position p through index i and to the global best position g .

The damping ratio ω_{damp} of the inertia weight is set to 1.

Active and inactive particles are assigned according to active ratios of 5%, 10%, ..., 95%, and 100%, as listed in Table 1. During ant nest relocation, the worker ants gather and form a basic unit of a swarm. Behavioral biologists report that the minimal size of this basic unit consists of 20 individuals, and so this is the smallest population size of a swarm of

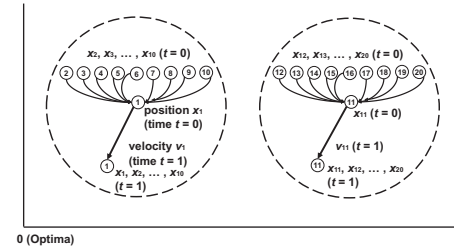


Figure 2: Inactive particles $i = 2, \dots, 10/12, \dots, 20$ duplicate the velocities $v_{i=111}(t = 1)$ and positions $x_{i=111}(t = 0)$ of active particles $i = 1/11$ that are closest to the inactive particles at the active ratio 10%.

Active Ratio	Active Particle “Ⓢ” & Inactive Particle “○”
5%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
10%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
15%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
20%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
25%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
30%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
35%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
40%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
45%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
50%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
55%	① ② ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
60%	① ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
65%	① ② ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
70%	① ② ③ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
75%	① ② ③ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
80%	① ② ③ ④ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
85%	① ② ③ ④ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
90%	① ② ③ ④ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
95%	① ② ③ ④ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
100%	① ② ③ ④ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Table 1: Active and inactive particles at the respective active ratios.

worker ants (Geraghty et al., 2007). Following this finding, we divided the population of particles into groups of 20 particles consisting of active and inactive particles. In every 20 particles, an active particle brings an almost equal number of inactive particles that are closest to the active particle toward the target of zero for the cost function. This simple simulation setting is straightforward and rational for task sharing.

We considered population sizes S ranging from 20 to 400, because swarms of workers from the *Temnothorax albipennis* ant species have been observed to range from a minimum of 20 ants to a maximum of 400 (Geraghty et al., 2007).

The maximum number of iterations was set to 40, 80, 120, 160, 200, 240, 400, or 1000, equivalent to 1, 2, 3, 4, 5, 6, 10, or 25 cycles of mutual contact among ants during nest relocation at a frequency of 40 contacts per minute, respectively.

Our simulations consisted of 50 runs of the PSO-based algorithm for each population size.

Results

The performance of our PSO-based algorithm was measured by the average global best position (hereafter referred to as the *global best*) and the average execution time (hereafter

referred to as the *execution time*) of the algorithm. These metrics evaluate the precision of convergence and the speed of convergence toward the target point of zero for the cost function, respectively. Smaller values of the global best indicate better performance, and a smaller execution time denotes faster performance. The simulations were performed on a Windows 10 Pro (64-bit) laptop PC equipped with a Core i7-7700 3.6 GHz quad-core processor with 16 GB of RAM.

We identified the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio for all population sizes.

We measured the global best and execution time for all population sizes at active ratios of 5%, 10%, ..., 95%, and 100%. Statistical testing was employed to determine whether our PSO-based algorithm performed better and faster than with the full 100% active ratio. The null hypothesis assumed that the global best and execution time were equal to those given by the full 100% active ratio for all population sizes. We used a two-tailed F -test to measure the two-sample variances and a two-tailed t -test for two-sample equality assuming equal/unequal variances.

Behavioral biologists divide the entire population into small populations of 20–100 and large populations of 100–400 (Franks et al., 2006; Dornhaus et al., 2008, 2009). We examined the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the entire population sizes, the small population sizes, large population sizes, and population sizes of 57 and 165, which are the medians of the small population sizes and large population sizes, respectively (Dornhaus et al., 2008). The simulation results of our PSO-based algorithm for these population sizes are omitted from this study due to page limitation.

Discussion

We now discuss the simulation results and show that our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at low active ratios. We also confirm the robust and stable performance of our algorithm at certain low active ratios. Three field studies on ant nest relocation that were conducted by ant biologists are then consulted. Their records provide sufficient support to demonstrate that our PSO-based algorithm performed like real-world ant swarm behavior at low active ratios.

First, we identify the active ratios at which our PSO-based algorithm performed significantly better and faster than the full 100% active ratio. Our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95% for certain maximum numbers of iterations.

Over the entire range of population sizes, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios denoted by a red check

mark (✓), namely 15%, 30%, 75%, and 80% at certain maximum numbers of iterations (see Table 2). With the small population sizes, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 45%, 75%, and 80% for certain maximum numbers of iterations (see Table 3). With the large population sizes, our PSO-based algorithm only performed significantly better and faster than the full 100% active ratio at an active ratio of 75% for certain maximum numbers of iterations (see Table 4). With a population size of 57, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 45%, 55%, and 80%–95% for certain maximum numbers of iterations. Finally, with a population size of 165, our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 45%, 55%, and 80%–95% for certain maximum numbers of iterations. Tables with the population sizes of 57 and 165 are omitted from this study due to page limitation.

We performed the two statistical tests (F -test and t -test) using the simulation results for the various active ratios and maximum numbers of iterations. The results are in conflict in terms of the global best and execution time. Instead of relying on the statistical tests, we used clustering to identify the active ratios at which our PSO-based algorithm performed significantly better and faster than the full 100% active ratio.

We placed a check mark against the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the simulation results. We then applied standard k -means clustering to the simulation results. The results of clustering indicate that our PSO-based algorithm achieved equivalent performance to the full 100% active ratio for a number of active ratios (denoted by the square symbol, □). We excluded the active ratios with a check mark within a square symbol (☑) from the simulation results. Our PSO-based algorithm performed significantly better and faster than the full 100% active ratio at those active ratios indicated by a red check mark (✓).

In the standard k -means clustering, we classified the global best and execution time at active ratios of 5%, ..., 100% for the entire set of population sizes as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Each matrix of the data object for clustering has twenty rows, corresponding to active ratios from 5% to 100%, and two columns for the global best and execution time at each maximum number of iterations. We iteratively calculated the distance between each data object (here, the global best and execution time) and all cluster centers, and assigned the data object to the closest cluster (in terms of Euclidean distance) at every iteration. We iteratively updated the calculation process and assigned the new means as the centroids of the data objects in the new clusters until the assignments no longer changed.

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☐	☐	☐	☐	☐	✓
20%-25%								
30%	☐	☐	☐	☑	☐	☑	☐	✓
35%	☐	☐	☐	☐	☐	☐	☑	☑
40%	☐	☑	☐	☑	☐	☐	☐	☐
45%	☐	☐	☐	☐	☐	☐	☐	☐
50%								
55%	☐	☐	☐	☐	☐	☐	☐	☐
60%	☐	☑	☐	☑	☐	☑	☐	☑
65%	☐	☑	☐	☐	☐	☑	☐	☐
70%	☐	☐	☐	☐	☐	☑	☐	☐
75%	☐	☐	☐	☐	☐	☐	✓	☐
80%	☐	☑	☐	☐	☐	☐	☑	✓
85%	☐	☐	☐	☑	☐	☐	☐	☐
90%-95%								
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 2: Active ratios that performed better and faster than the full 100% active ratio across all population sizes (✓).

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☐	☐		☐	☐	✓
20%-25%								
30%	☐	☐	☐	☑		☑	☐	✓
35%	☐	☐	☐	☐		☐	☐	☐
40%	☐	☑	☐	☑	☐	☐	☐	☐
45%	☐	✓	☐	☐		☐	☐	☐
50%								
55%	☐	☐	☐	☐		☑	☐	☐
60%	☐	☑	☐	☑	☐	☑	☐	☑
65%	☐	☑	☐	☐	☐	☑	☐	☐
70%	☐	☐	☐	☐		☑	☐	☐
75%	☐	☐	☐	☐		☐	✓	☐
80%	☐	☑	☐	☐		☐	☑	✓
85%	☐	☐	☐	☑	☐	☐	☐	☐
90%	☐							
95%	☐							
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 3: Active ratios that performed better and faster than the full 100% active ratio with small population sizes (✓).

Active Ratio	Maximum Number of Iterations							
	40	80	120	160	200	240	400	1000
5%-10%								
15%	☐	☐	☑	☑	☐	☑	☐	
20%-25%								
30%	☐	☐	☐	☑	☐	☑	☐	
35%	☐	☐	☐	☑	☐	☐	☑	☑
40%	☐	☐	☐	☑	☐	☑	☐	☐
45%	☐	☐	☑	☐	☑	☑	☐	☐
50%								
55%	☐	☐	☐	☑	☐	☐	☐	☐
60%	☐	☐	☐	☑	☐	☐	☐	☐
65%	☐	☐	☑	☑	☐	☑	☐	☐
70%	☐	☐	☑	☑	☐	☑	☐	☐
75%	☐	☐	☐	☐	☐	☐	✓	☐
80%	☐	☐	☑	☑	☐	☑	☑	☐
85%	☐	☐	☑	☑	☐	☐	☐	☐
90%								
95%								
100%	☐	☐	☐	☐	☐	☐	☐	☐

Table 4: Active ratios that performed better and faster than the full 100% active ratio with large population sizes (✓).

At the active ratios denoted by square symbols, the simulation results of our PSO-based algorithm were ultimately assigned to the cluster that included the simulation results for the full 100% active ratio. We repeated the clustering process using 2–11 clusters, and found that the assignments remained unchanged when six clusters were used. Thus, the number of clusters was set to six.

Next, we identify the active ratios that remained robust and stable over various ranges of maximum numbers of iterations (e.g., 40–80, ..., and 40–1000). Our PSO-based algorithm was robust and stable at the three active ratios of 60%, 80%, and 85%, over certain ranges of maximum numbers of iterations.

For the entire set of population sizes and the small population sizes, the performance of our PSO-based algorithm was neither robust nor stable for any active ratios (see Tables 5 and 6). With the large population sizes, our PSO-based algorithm was robust and stable at an active ratio denoted by a red check mark (✓), namely 80% for maximum numbers of iterations from 40 to 1000 (see Table 7). With a population size of 57, our PSO-based algorithm was robust and stable at two active ratios, 60% and 85%, over maximum numbers of iterations from 40 to 240 and from 40 to 1000, respectively. With a population size of 165, our PSO-based algorithm was robust and stable at an active ratio of 60% over maximum numbers of iterations from 40 to 80. Tables with the population sizes of 57 and 165 are omitted from this study due to page limitation.

We averaged the rank of the simulation results in terms of the global best and execution time for active ratios of 5%, ..., 100% over ranges of maximum numbers of iterations 40–80, 40–120, 40–160, 40–200, 40–240, 40–400, and 40–1000 for the entire set of population sizes, as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Check marks indicate the active ratios at which our PSO-based algorithm performed better and faster than the full 100% active ratio in the averaged ranking of the simulation results over the various ranges of maximum numbers of iterations. We applied standard k -means clustering to the averaged ranking results. The clustering results show that our PSO-based algorithm was equivalent to the full 100% active ratio at a number of active ratios (denoted by the square symbol, □). We excluded active ratios with a check mark within a square symbol (☑) from the averaged ranking results. The performance of our PSO-based algorithm was robust and stable at the active ratios indicated by a red check mark (✓).

In the standard k -means clustering, we classified the global best and execution time at active ratios of 5%, ..., 100% for the entire set of population sizes, as well as the small population sizes, large population sizes, and population sizes of 57 and 165. Each matrix of the data object for clustering has twenty rows, corresponding to active ratios 5%–100%, and 4–16 columns consisting of the global best

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	□	□	□	□	□	□	
20%-25%							
30%	□	□	□	□	□	□	
35%-45%	□	□	□	□	□	□	□
50%							
55%	□	□	□	□	□	□	□
60%	□	□	☑	□	□	□	☑
65%-75%	□	□	□	□	□	□	□
80%-85%	□	□	□	□	□	□	□
90%-95%							
100%	□	□	□	□	□	□	□

Table 5: Robustness and stability of active ratios for all population sizes (✓).

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	□	□	□	□	□	□	
20%-25%							
30%	□	□	□	□	□	□	
35%	□	□	□	□	□	□	□
40%	☑	□	□	□	□	□	□
45%	☑	□	□	□	□	□	□
50%							
55%	□	□	□	□	□	□	□
60%	☑	☑	☑	□	□	☑	☑
65%-75%	□	□	□	□	□	□	□
80%-85%	□	□	□	□	□	□	□
90%-95%	□	□	□	□	□	□	□
100%	□	□	□	□	□	□	□

Table 6: Robustness and stability of active ratios with small population sizes (✓).

Active Ratio	Range of Maximum Numbers of Iterations						
	40-80	-120	-160	-200	-240	-400	-1000
5%-10%							
15%	□	□	□	□	□	□	
20%-25%							
30%	□	□	□	□	□	□	
35%-40%	□	□	□	□	□	□	□
45%	□	□	□	□	□	☑	□
50%							
55%	□	□	□	□	□	□	□
60%	□	□	☑	□	□	□	□
65%-75%	□	□	□	□	□	□	□
80%	□	□	□	□	□	□	□
85%	□	□	☑	☑	☑	☑	✓
90%-95%							
100%	□	□	□	□	□	□	□

Table 7: Robustness and stability of active ratios with large population sizes (✓).

and execution time at maximum numbers of iterations 40–80, ..., 40–1000. We iteratively calculated the distance between each data object (here, the global best and execution time) and all cluster centers, and assigned the data object to the closest cluster (in terms of the Euclidean distance) at every iteration. We iteratively updated the calculation process and assigned the new means as the centroids of the data objects in the new clusters until the assignments no longer changed.

Active ratios with square symbol indicate that the aver-

aged ranking results were ultimately assigned to the cluster that included the averaged ranking results at the full 100% active ratio (again, the number of clusters is six).

The results presented above are supported by three field studies on ant nest relocation that were carried out by expert ant biologists (Pratt, 2005b; Dornhaus et al., 2009, 2008). These field studies show that the maximum active ratios range 31.0%–58.0% across all population sizes, 31.0%–56.0% for small population sizes, and 52.0%–58.0% for large population sizes.

Active ratios 30%–60% in our simulation results completely cover the range of maximum active ratios observed in the field studies. The maximum of 58.0% is between the active ratios of 55% and 60% in our simulation results, and is slightly closer to the latter value. The performance of our PSO-based algorithm was not only significantly better and faster than the full 100% active ratio, but was also robust and stable at an active ratio of 60%. Thus, based on the empirical support of these three field studies, we have successfully simulated ant nest relocation at low active ratios using our PSO-based algorithm.

We chose the sphere function as the objective in our PSO-based algorithm. With this simple function, we generated various sets of simulation results and identified the benefits and limitations of the algorithm very clearly. Although we have only presented simulation results obtained with the sphere function, we could apply various cost functions to our PSO-based algorithm. We will conduct simulations using other functions in future studies.

Our modeling has been limited to the relation between the active ratio and ant nest relocation, because real-world ant nest relocation is a scale-free swarm behavior that depends not on the population size, but on the frequency of mutual contact among ants, which determines the active ratio.

Conclusion

Ant nest relocation is smoother and swifter than the same process performed by other animals. However, many worker ants are never involved in nest relocation. An open problem concerns whether this low active ratio improves ant nest relocation. In this study, we used an algorithm based on the original PSO and simulated real-world ant nest relocation. Our PSO-based algorithm mimics the real-world ant swarm behavior of active and inactive workers in ant nest relocation. Our algorithm duplicates the velocity and position of an inactive particle, which represents an inactive worker ant, with the velocity and position of an active particle, representing an active worker ant. Clustering of the simulation results of our PSO-based algorithm provides sufficient support to demonstrate that the simulation results are reliable. Our PSO-based algorithm performs significantly better and faster than the full 100% active ratio at active ratios of 15%, 30%, 35%, 45%, 55%, 60%, and 75%–95%. In particular, we have confirmed the robust and stable performance of our

PSO-based algorithm at active ratios of 60%, 80%, and 85%. The active ratio of 60% is very close to the maximum active ratio of 58.0% reported in three field studies conducted by expert ant biologists. We have shown that low active ratios significantly improve ant nest relocation in terms of speed and the precision of movement. Our model is limited to the relation between the active ratio and ant nest relocation, because the population size has no impact on the relocation process. The simulations used a simple cost function, though we could apply other functions to the algorithm. In future work, we will conduct simulations using various cost functions.

References

- Ali, M. Z., Awad, N. H., and Duwairi, R. M. (2016a). Multi-objective differential evolution algorithm with a new improved mutation strategy. *International Journal of Artificial Intelligence*, 14(2):23–41.
- Ali, M. Z., Awad, N. H., Suganthan, P. N., and Reynolds, R. G. (2016b). A modified cultural algorithm with a balanced performance for the differential evolution frameworks. *Knowledge-Based Systems*, 111(1):73–86.
- Awad, N. H., Ali, M. Z., and Duwairi, R. M. (2013). Cultural algorithm with improved local search for optimization problems. In *Proceedings of the 15th IEEE Congress on Evolutionary Computation*, pages 284–291. IEEE Press, Piscataway, NJ.
- Awad, N. H., Ali, M. Z., and Duwairi, R. M. (2017). Multi-objective differential evolution based on normalization and improved mutation strategy. *Natural Computing*, 16(4):661–675.
- Awad, N. H., Ali, M. Z., Suganthan, P. N., and Jaser, E. (2016). A decremental stochastic fractal differential evolution for global numerical optimization. *Information Sciences*, 372(1):470–491.
- Burd, M., Archer, D., Aranwela, N., and Stradling, D. J. (2002). Traffic dynamics of the leaf-cutting ant, *Atta cephalotes*. *The American Naturalist*, 159(3):283–293.
- Charbonneau, D., Sasaki, T., and Dornhaus, A. (2017). Who needs ‘lazy’ workers? inactive workers act as a ‘reserve’ labor force replacing active workers, but inactive workers are not replaced when they are removed. *PLoS ONE*, 12(9):e0184074.
- Chowdhury, D., Guttal, V., Nishinari, K., and Schadschneider, A. (2002). A cellular-automata model of flow in ant trails: non-monotonic variation of speed with density. *Journal of Physics A: Mathematical and General*, 35(41):L573–L577.
- Cleghorn, C. W. and Engelbrecht, A. P. (2016). Particle swarm optimizer: the impact of unstable particles on performance. In *Proceedings of the 7th IEEE Symposium Series on Computational Intelligence*, pages 1–7. IEEE Press, Piscataway, NJ.
- Cleghorn, C. W. and Engelbrecht, A. P. (2018). Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, 12(1):1–22.

- Clerc, M. and Kennedy, J. (2002). The particle swarm - exploration, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.
- Dornhaus, A., Holley, J.-A., Pook, V. G., Worswick, G., and Franks, N. R. (2008). Why do not all workers work? colony size and workload during emigrations in the ant *Temnothorax albipennis*. *Behavioral Ecology and Sociobiology*, 63(1):43–51.
- Dornhaus, A., Holley, J.-A., and Franks, N. R. (2009). Larger colonies do not have more specialized workers in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 20(5):922–929.
- Franks, N. R., Dornhaus, A., Best, C. S., and Jones, E. L. (2006). Decision making by small and large house-hunting ant colonies: one size fits all. *Animal Behaviour*, 72(3):611–616.
- Geraghty, M. J., Dunn, R. R., and Sanders, N. J. (2007). Body size, colony size, and range size in ants (*Hymenoptera: Formicidae*): are patterns along elevational and latitudinal gradients consistent with Bergmann’s rule? *Myrmecological News*, 10:51–58.
- John, A., Schadschneider, A., Chowdhury, D., and Nishinari, K. (2008). Characteristics of ant-inspired traffic flow: applying the social insect metaphor to traffic models. *Swarm Intelligence*, 2(1):25–41.
- John, A., Schadschneider, A., Chowdhury, D., and Nishinari, K. (2009). Traffic like collective movement of ants on trails: absence of jammed phase. *Physical Review Letters*, 102(10):e108001.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. IN *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE Press, Piscataway, NJ.
- Pereira, G. (2010). *Particle Swarm Optimization in Artificial Life by Example*. The codes are publicly available at <http://web.ist.utl.pt/gdgp/VA/pso.htm>.
- Pratt, S. C. (2005a). Quorum sensing by encounter rates in the ant *Temnothorax albipennis*. *Behavioral Ecology*, 16(2):488–496.
- Pratt, S. C. (2005b). Behavioral mechanisms of collective nest-site choice by the ant *Temnothorax curvispinosus*. *Insectes Sociaux*, 52(4):383–392.
- Sasaki, H. (2017). Modeling time-sensitive swarm dynamics. In *Proceedings of the 8th IEEE Symposium Series on Computational Intelligence*, pages 2489–2496. IEEE Press, Piscataway, NJ.
- Sasaki, H. (2018a). A PSO-based algorithm describes ant nest move. In *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics*, pages 243–248. IEEE Press, Piscataway, NJ.
- Sasaki, H. (2018b). Particle swarm optimization describes ant nest move at low participation rate. In *Proceedings of the 9th IEEE Symposium Series on Computational Intelligence*, pages 989–996. IEEE Press, Piscataway, NJ.
- Sasaki, H. (2019). Modeling ant nest relocation at low active ratio by particle swarm optimization. In *Proceedings of the 21st IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ. (to appear).
- Sasaki, H. and Leung, H. (2013). Trail traffic flow prediction by contact frequency among individual ants. *Swarm Intelligence*, 7(4):307–326.
- Yarpiz Team (2015). *Particle Swarm Optimization in MATLAB*. In Yarpiz - Academic Source Codes and Tutorials. The codes are publicly available at <http://yarpiz.com/50/ypea102-particle-swarm-optimization>.

Collective Event Detection Using Bio-inspired Minimalistic Communication in a Swarm of Underwater Robots

Joshua Cherian Varughese^{1,2}, Hannes Hornischer^{1,3}
Ronald Thenius¹, Franz Wotawa² and Thomas Schmickl¹

¹ Institute for Biology, University of Graz, Austria

² Institute for Software Technology, Graz University of Technology, Austria

³Institute of Systems Sciences, Innovation and Sustainability Research, University of Graz, Austria
ronald.thenius@uni-graz.at

Abstract

Mobile sensor networks and robotic swarms are being used for monitoring and exploring environments or environmental events due to the advantages offered by their distributed nature. However, coordination and self-organization of a large number of individuals is often costly in terms of energy and computation power, thus limiting the longevity of the distributed system. In this paper we present a bio-inspired algorithm enabling a robotic swarm to collectively detect anomalies in environmental parameters in a self-organized, reliable and energy efficient manner. Individuals in the swarm communicate via 1-bit signals to collectively confirm the detection of an anomaly while minimizing energy spent for communication and taking measurements. This algorithm is specifically designed for a swarm of underwater robots called “aMussels” to examine a phenomenon referred to as “anoxia” which results in oxygen depletion in the lagoon of Venice. We present the algorithm, conduct simulations and robotic experiments to examine the performance of the algorithm with respect to early detection of anoxia while minimizing energy consumption.

Introduction

With the decreasing size of computation and memory devices, the number of computers has been increasing dramatically. Koh and Magee (2006) observed that computing power available per dollar has increased by a factor of ten roughly every four years over the last quarter of a century. The increase in the available computation power has brought massively parallel multi-agent systems such as ubiquitous computers (Kim and Follmer, 2017), IoT systems (Gubbi et al., 2013), swarm robotics (Zahadat and Schmickl, 2016; Witkowski and Ikegami, 2016) to the forefront. Such systems with increasingly large number of individual interacting parts pose challenges to the traditional top down control schemes. Therefore, decentralized computing strategies with little or no top down control are being widely explored and implemented (Schmickl et al., 2008; Cazangi et al., 2005). Inspiration for such strategies are drawn from self organizing natural systems such as starling murmurations (Cavagna et al., 2010), honeybee colonies (Seeley, 1992), slime mould aggregates (Durstun, 1973; Bonner, 1949).

We present a decentralized algorithm for a swarm of underwater robots to detect, to collectively validate and to report significant variations in environmental parameters. In a nutshell, if a member of the swarm detects an anomaly in its measurements it will register an event and alert its neighbors. A periodic oscillator and traveling wave based communication paradigm inspired by slime mould and fireflies is then used to periodically communicate with neighbors who registered the event. As soon as the event is validated by a sufficient number of neighbors, an alert is sent to a base station.

The algorithm presented here is developed for a swarm of underwater robots which is intended to detect the *anoxia* phenomenon (Runca et al., 1996) in the lagoon of Venice. During anoxia the oxygen content of a small part of the lagoon decreases suddenly, resulting in the death of flora and fauna and thus damaging the local ecosystem. Thenius et al. (2018a) suggest a strategy for examining and documenting this recurring phenomenon by utilizing a swarm of underwater robots. According to this strategy, a team of underwater robots known as “aMussels” will be used for monitoring a set of environmental parameters, including oxygen concentration levels to detect the phenomenon within the framework of project subCULTron (subCULTron, 2015)¹. On the one hand, the detection of anoxia by individual robots needs to be validated with a number of neighbors before a global alarm can be raised for greater reliability. On the other hand, underwater communication is expensive, noisy and therefore, exchange of information between the robots needs to be minimized. For this reason we focus on detecting and validating the event of anoxia with a swarm of robots while considering the modes of communication available on the aMussel robots.

Many environmental monitoring systems which use sensor networks for collecting data in a large area have focused

¹In subCULTron, a heterogeneous swarm of autonomous aquatic robots (aMussels, aPads and aFish) and associated algorithms are being developed for collective exploration and monitoring of environmentally diverse underwater niches in the lagoon of Venice. Please refer to <http://www.subcultron.eu/> for further details.

on reducing the energy consumption in order to increase network longevity (Zhou et al., 2015; Kaur and Sood, 2017). However, most of the existing protocols and algorithms for underwater sensor networks focus on making the algorithms usable for a wide range of communication devices, especially for deep sea environments. By contrast, we focused on designing an event detection algorithm suitable for the low cost, narrow bandwidth and low payload communication devices used in subCULTron. Specifically, aMussels are equipped with three kinds of underwater communication devices. Modulated blue light communication and electric sense for extremely short ranges (~ 1 meter) as well as a low-frequency acoustic nanomodem for comparatively longer range underwater communication (~ 100 meter). In addition to that, aMussel robots are not mere sensor nodes but have the capability of diving up and down in a water column. This enables the aMussels to dive up to the water surface and report the occurrence of an event using ultra long range communication devices rather than forwarding packets to the sink nodes like traditional sensor networks do. Keeping these constraints and special capabilities of the aMussel robots in mind, in the following we suggest an algorithm for event detection in autonomous swarms of robots.

Related Work

Many algorithms and protocols have been proposed and implemented for improving deep sea monitoring using underwater wireless sensor networks (UWSNs)(Zhou et al., 2015; Debont et al., 2012). Although many techniques used in the classical wireless sensor networks can be used for their underwater counterparts, communication in an underwater environment is especially challenging. Therefore, we will give a brief overview of some event localization schemes suggested for UWSNs.

Since underwater sensor networks might be mobile due to underwater currents, the communication protocol presented by Zhou et al. (2015) includes a “heartbeat” which periodically communicates with neighboring nodes and constructs a routing tree. Additionally, the system aggregates data and processes it locally to detect an event before forwarding the event to the sink node.

Debont et al. (2012) suggest a solution for event detection using a cyclic graph model. The solution optimally places “beacon” nodes which act as location-aware references for surrounding nodes. In Debont et al. (2012) the authors showed that the intelligent placement of beacons reduces the number of sends required by 80 % as compared to a naïve placement. In case of an event, a message is forwarded to the beacon node which in turn acts as a buffer to collect more event messages from other nodes. Then a batch of messages containing alarms is forwarded to the sink node. While such an implementation is helpful for underwater sensor networks in general, it requires elaborate routing protocols and non-minimal message lengths.

Repeated and periodic exchange of information is employed in the above implementations of event detection. While a “heartbeat” signal (Zhou et al., 2015) is important for dynamic deep sea sensor networks where nodes move with currents, it is a costly solution for the shallow lagoon of Venice with minimal water movement. As previously mentioned, each aMussel robot is able to dive up to the surface of the lagoon and use their ultra long range communication capabilities to alert the base station and therefore obviates beacon nodes as in (Debont et al., 2012) or “data aggregator” nodes. In addition, the construction of an elaborate routing tree as suggested by Zhou et al. (2015) and (Debont et al., 2012) necessitates an increased communication payload which is not necessary for event detection in systems such as the robotic swarms developed within the framework of subCULTron.

The Algorithm

To enable a swarm to detect and report an event in an energy efficient manner, we introduce three modes of operation of the robots (or agents): “observation mode”, “alert mode” and “event mode”. A schematic representation of the different modes can be found in Figure 1.

- Observation mode: Initially, all agents are in this mode where they periodically take measurements but refrain from any means of active communication.
- Alert mode: If an agent in the observation mode receives any active communication signal, it will enter the alert mode. In the alert mode an agent also increases its frequency of measurement in order to detect a prospective event as early as possible but refrains from any active communication.
- Event mode: As soon as an agent deduces the potential occurrence of an event from its measurements, it enters the event mode. In this mode an agent takes measurements with higher frequency in order to observe changes in the environment and collect data with higher temporal resolution. Furthermore, in the event mode agents periodically send 1-bit signals which are received by neighboring agents. If an agent in the event mode receives a signal, it simply relays the signal. In the event mode the agents also estimate the number of other swarm members which are in the event mode. As soon as an agent reaches a sufficiently high estimate of other agents detecting an event, it reports the event to the base station. The mechanism used by the agents to estimate the number of agents in the event mode is explained below.

We designed the algorithm to validate the event with a sufficient number of neighboring agents in order to cope with sensor failure and hence potentially erroneously reported events. The swarm validates the occurrence by estimating

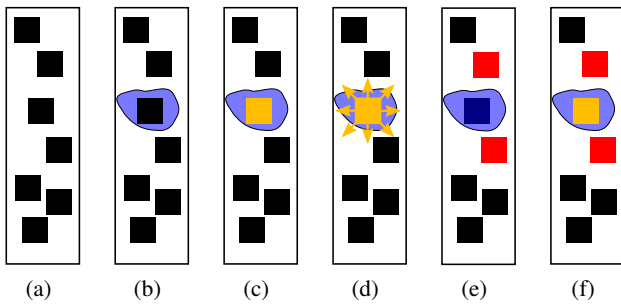


Figure 1: A schematic representation of the modes of operation. (a): A swarm of agents in observation mode (black). (b): An irregularity in environmental parameters occurs (blue area). (c): The agent in the blue area takes a measurement and detects the irregularity. The agent enters the event mode (yellow). (d): The agent in the blue area sends a signal to neighboring agents. (e) The neighboring agent receive the signal and subsequently enter alert mode (red). The agent which sent a signal enters stays insensitive to incoming signals (dark blue) for a defined duration after which, in (f) it is able to receive signals again.

the total number of agents in the swarm which are in event mode. To explain how this was realized, we first introduce two concepts.

First, every time an agent in event mode sends a signal, it gets relayed by all the other agents in event mode. This leads to a wave-like propagation of signals through a sub-swarm of agents in event mode. In order to ensure forward propagation of the signal, we introduce a “refractory” time (t_{ref}), where every agent stays insensitive to signals for a brief period after sending a signal. The propagation of a wave through the sub-swarm of robots which has detected an event is schematically shown in Figure 2.

Secondly, all agents have an internal timer with a defined and fixed periodicity P . An agent in the event mode broadcasts a signal once during each period of this internal timer. Due to the relaying of the signals by other agents in the event mode, every signal broadcast by the agents reach the other agents in the sub-swarm. This wave of signals propagating through the sub-swarm combined with the periodicity of communication enables the agents to estimate the number of agents in event mode. This estimation is done by counting the number of signals the agents receive during one period of their internal timers. The time at which an agent sends a signal during this period is chosen randomly in order to avoid overlapping signals. As soon as an agent detects a predefined number of agents (n_*) in the event mode, it dives up to the water surface and reports the event to a base station via the ultra long range communication mode.

The wave like propagation of signals through the swarm is inspired by the cyclic Adenosine Monophosphate (cAMP)

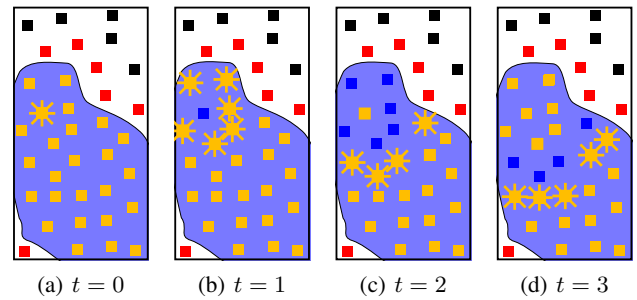


Figure 2: A schematic representation of the relaying of signals in the event mode. (a): An agent in event mode initiates the sending of a signal (yellow star). After signaling, an agent stays insensitive to incoming signals for a specified duration (shown in dark blue). The consecutive relaying of this signal by neighboring agents in event mode is shown in (b) - (d), respectively.

signaling used by *dictyostelium discoideum* (Chisholm and Firtel, 2004). The internal timer with a defined periodicity is analogous to timer like behavior found in fireflies (Buck and Buck (1966), Camazine et al. (2001)). Further details of this bio-inspired communication method are explained in Thenius et al. (2018b).

Simulation

We utilized Netlogo 6.0.2 for conducting simulations and the fundamental units for space and time are referred to as “patches” and “ticks” respectively. Agents are randomly distributed within a system of size 110×110 patches with periodic boundary conditions. Agents have a communication range of 9, i.e., a signal can be received by all agents within euclidean patch-to-patch distance of $s_r = 9$. We take the perception of communication for the individual agents to be circular and therefore the communication area to be $s_A \approx \pi \cdot 81$. At a random position in the system the anoxic phenomenon is initialized with an area of 1 patch and spreads to all adjacent patches within the Moore neighborhood at a constant rate (*ticks*). The agents can detect anoxia solely at the exact position where they are located. All agents choose random times during each of their internal periods at which they measure and potentially send signals. The refractory time during which agents stay insensitive to incoming signals after sending a signal is $t_{ref} = 5$ ticks. Figure 3 shows two screen shots of an exemplary simulation in an early state as well as its final state where an agent reports the occurrence of an event. If not stated otherwise, the number of agents necessary to agree on the occurrence of an event in order to report it is set to $n_* = 5$. The parameters are deliberately selected to demonstrate the working of the algorithm and those parameters which affect the performance will be introduced in the upcoming section.

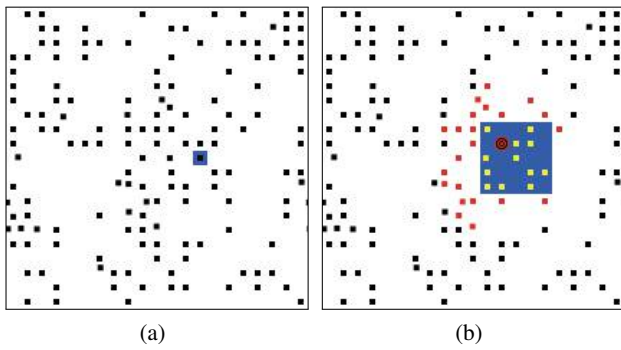


Figure 3: Screen shots of a simulation run of the presented algorithm. The agents are placed randomly in the arena with uniform probability; in (a) and (b) the blue domain represents the presence of anoxia. The black, yellow and red squares represent the individual robots which are in observation mode, event mode and alert mode, respectively. The red target symbol in (b) represents the agent in event mode which sends an alert to the base station.

Swarm level parameters

In order to quantify the performance of the algorithm, we introduce some parameters which reflect the characteristics and performance of the swarm of robots monitoring the environment for the event.

- Measurement and communication periodicity (P): This parameter signifies the periodicity with which all agents in the system communicate. P is measured in unit ticks.
- Density of robots (D): This parameter measures how densely or sparsely the agents are spread in the environment. D is measured in robots/ s_r^2 where $s_r^2 \cdot \pi$ is the area of perception of one robot.
- Time until reporting (T): Assuming the start of anoxia at tick = 0, the number of ticks taken until an agent sends a message to the base station. The unit of measurement of this parameter is ticks.
- Area of spread of anoxia(A): The total area anoxia covers until an agent sends a message to the base station. The unit of measurement of this parameter is in patches.

Results

In Figure 4 we show the dependence of time T until agents report an event on the periodicity P of agents. Every data point is averaged over 100 independent simulations.

As P approaches 1, the time until reporting diverges towards infinity. For $P < 100$, the agents communicate so frequently that due to the refractory time associated with each broadcast, they rarely receive signals. Thus, agents rarely get to confirm that other agents share their opinion on the

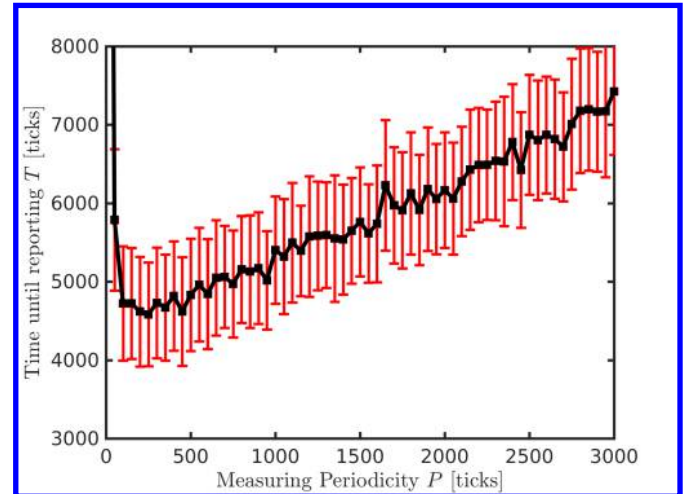


Figure 4: Time T until reporting an event versus measuring periodicity P , i.e., the periodicity of agents taking measurements of environmental parameters and optionally (if they are in event state) send signals. Parameters: $D = 1$, $t_{ref} = 5$ ticks.

occurrence of an event. For $P \approx 300$, the time until reporting has a minimum value of $T \approx 4500$ ticks. Thereafter T grows in an approximately linear manner for increasing P . In the extreme case of $P \rightarrow \infty$, the area of the event spreads throughout the system and is detected only when agents first measure and then signal. From Figure 4, we derive an optimal measuring periodicity of $P \in [200, 500]$ for which time T is at a minimum. Since within the interval the time to report the event T remains rather stable, we choose $P = 500$ in order to minimize signal collisions among the pinging agents.

Figure 5 shows the time until reporting of an event (averaged over 100 independent simulations per data point) versus the spatial density of agents D , i.e., the average number of agents within an area of s_r^2 . For $D = 1$, the average time until reporting of an event is $T \approx 6200$ ticks. For increasing D , T decreases until for $D = 3.5$ it reaches a plateau at $T \approx 2000$ ticks. As D approaches 0, we expect T to diverge, since the density is too low for agents to communicate with neighbors and therefore too low for confirming the occurrence of an event. However, for $D > 3.5$ we reach a regime where agents are sufficiently well connected such that a further increase in density does not change the collective behavior or performance of the swarm anymore.

Since for $D > 3.5$ the time T does not dramatically decrease further and therefore the performance in detecting events as fast as possible does not further increase, for the following simulations we choose $D = 3.5$ as optimal parameter value.

For a set of agents within the vicinity of an occurring event we identified a set of parameters for optimal perfor-

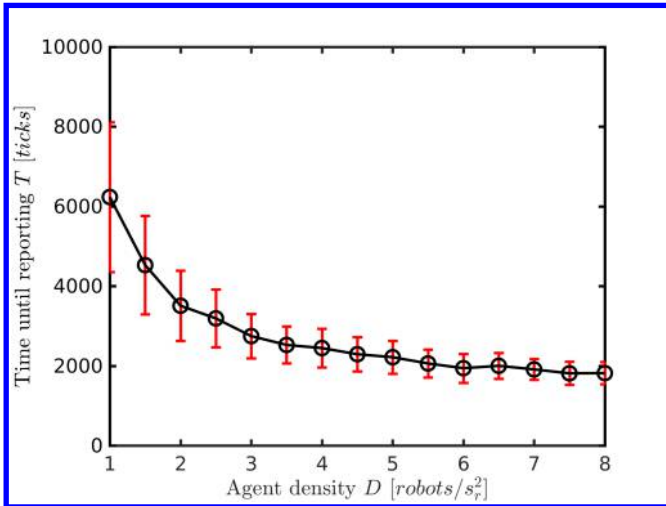


Figure 5: Time T until reporting of an event versus agent density D , i.e., the number of agents within an area of s_r^2 .

mance of the swarm, i.e., for minimizing the time until reporting an event. However, in case a swarm succeeds in detecting and reporting an event relatively quickly, the area of the event is comparably small and therefore the majority of all agents in the swarm is not within the neighborhood of the event. Furthermore, in a practical application a swarm is likely to be deployed over a long time period until an event might occur. In order to minimize energy consumption not only the communication between agents can be minimized but also the frequency of taking measurements in the observation mode. In the following we examined, how well a swarm performs if we decrease the periodicity P in taking measurements for all agents in observation mode. As soon as an agent enters alert mode or event mode, they adjust their periodicity P to the original value again. Therefore, agents in observation mode measures less frequently, but as soon as they detect an anomaly, they take measurements (and potentially send signals) more frequently.

Figure 6 shows the time T until reporting an event versus the factor k by which the periodicity in taking measurements for agents in observation mode is reduced (black circles). All data points shown are averaged over 100 independent simulations. For $k = 1$, agents have the same periodicity P in all states and take on average $T \approx 2700$ ticks until reporting an event. Up to $k = 4$ the time T fluctuates around $T = 2900$ ticks or slightly increases. For $k > 4$ time T increases linearly. For k approaching ∞ , agents in observation mode take measurements (linearly) increasingly rarely such that over time the event area spreads out until agents first measure and subsequently report the event. Therefore for large k a linear increase is expected. The number of measurements taken until reporting of the event are also shown in Figure 6. The blue squares denote the total number of

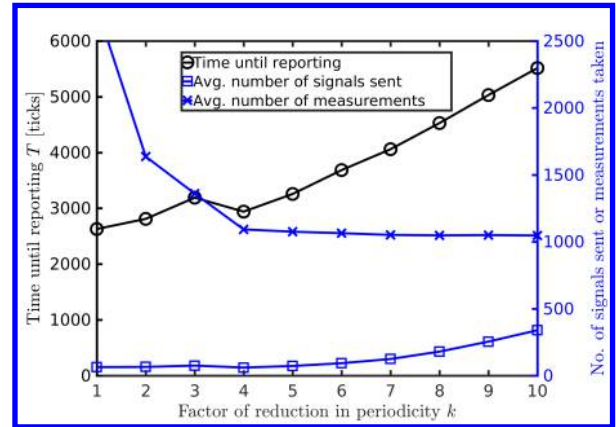


Figure 6: Time until reporting T (black circles) versus the factor by which the agents in the observation mode reduce their P as compared to the agents in alert mode or event mode. The number of signals sent and measurements taken (blue squares and crosses, respectively) are also visualized against the factor of reduction k . Parameters: $P = 500$, $t_{ref} = 5$ ticks.

measurements taken by the swarm, averaged over the simulation runs. With increasing k , the agents which are in observation mode detect the phenomenon later, thus letting anoxia spread to a larger area. A large number of agents then transition into event mode, therefore increasing the number of messages sent. The blue crosses denote the total number of measurements taken averaged over each simulation run. As expected, as periodicity is scaled down by an increasing k , the number of measurements taken decreases.

We can conclude from the graph that for the given system a value of $k = 4$ will produce relatively fast reporting of events while reducing the number of measurements taken and therefore reducing the energy consumed.

Robotic Experiments

In order to validate the algorithm, we implemented it on the aMussel robots and tested it under laboratory conditions. Five aMussels were arranged in a linear manner in an arena as shown in the photographs in Figure 7 (a). A projector was used to manipulate the local ambient light around the aMussels to represent oxygen content in the lagoon of Venice. The robots were programmed to register an event when the measured ambient light fell below a particular threshold. As the experiment progressed, the dark patch on the right side of the arena expanded towards the center of the arena as shown in in figures 7 (b), (c) and (d). The robots were programmed to light up the LEDs on their top-caps to represent their mode of operation. As shown in Figure 7 (b), the first robot from the right transitioned into the event mode as represented by the green LED on its top-cap. At the same time, a signal was broadcast by this robot using the short range

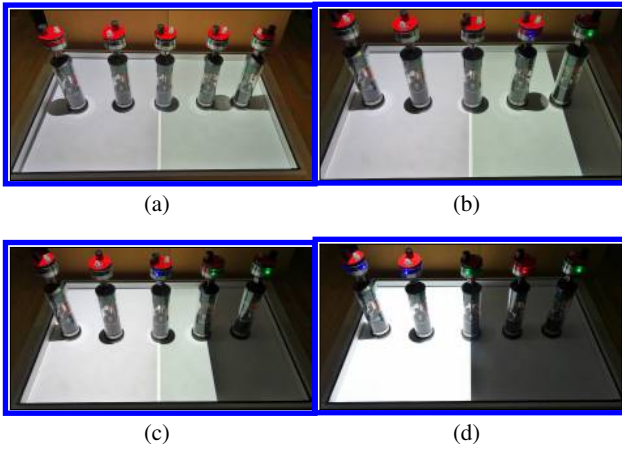


Figure 7: An exemplary run from the robotic experiments conducted with aMussels under laboratory conditions. The agents indicate their modes of operation using the LEDs on their respective top-caps. The blue, green and red LEDs signify the alert mode, the event mode and contacting base station respectively. When there is no LED lit, it means that the agents are in the observation mode.

modulated blue light communication module. This signal triggered the second robot from the right into the alert mode as shown by the blue LEDs on its top-cap. As the dark patch expanded towards the center of the arena (figures 7 (b) - (d)), the robots transitioned into the alert mode and subsequently into the event mode. In the event mode, the robots counted the incoming signals to estimate the number of other robots in event mode. In Figure 7 (d), the event threshold of $n_* = 3$ was reached for the second robot from the right. It sent a signal via bluetooth to the monitoring station and lit up the red LEDs on its top-cap to signal a validated event in its locality.

Discussion

In the above sections, we presented a simple but practical bio-inspired event detection algorithm for detecting the anoxia phenomenon in the lagoon of Venice using the aMussel robots. Our solution is designed to minimize the amount of communication needed while specifically taking into account the communication and locomotion capabilities of the aMussel robots. Although energy minimization is an ongoing subject of research in sensor networks, the main body of existing literature focuses on reducing the frequency of communication. In contrast to the energy minimization discussed in Zhou et al. (2015) and Debont et al. (2012), this work not only focuses on reducing the number of messages required for effective coordination but also on minimizing the length of each message. From figures 4 and 5, we can conclude that the periodicity of measurement and communication as well as the density of distribution of robots are crucial factors in limiting the spreading time of a phenomenon

like anoxia. Figure 6 implicitly shows the energy consumed for taking measurements by displaying the total number of measurements taken by the swarm as well as the average number of measurements taken per robot.

For a spreading phenomenon it is intuitive that an increase in the number of robots which are monitoring the area ensures a faster detection of a spreading phenomenon. Figure 5 shows how the density of robots in the area of interest affects the performance of the algorithm. In order to ensure an interconnected network, a requirement for the presented algorithm to reliably work, each robot needs to be connected to the rest of the swarm through at least one other robot being in its communication range. It follows that there must be at least one robot per sensor area. Assuming a perfectly circular sensor radius of robots, the limits to density of robots D and its relation to sensor area (s_A), and the event threshold (n_*) can be modeled by equations 1 and 2. In simulations, we randomly placed robots in the environment. In reality, rather than using a naïve placement or random distribution of robots, we will generate a set of GPS positions to deploy the aMussels. Since there is a set threshold of n_* agents necessary to agree on detecting the anoxia across a preset maximum area of A_{max} , the density can be so chosen to be D according to Equation 2.

$$D_{min} = 1 \text{ robots}/s_A \approx \frac{0.32}{s_r^2} \text{ robots} \quad (1)$$

$$D = \frac{n_*}{A_{max}} \text{ robots}/s_A \quad (2)$$

Figure 4 shows the relation between periodicity P of measurement and communication and the time until reporting T . In general, as P decreases, the time taken for reporting the event decreases since faster measurement and communication enables faster detection. However, below a particular value P_{min} , the event cannot be detected due to signal collisions as shown by the initial spike in Figure 4. Therefore P has to be selected so as to minimize the probability of signal collisions. Generally, P has to be large enough to allow each agent to communicate in a different “time slot”. Each of these slots consists of a temporal span for each robot to send the one bit signal and complete its associated refractory time. As a rule of thumb, we used the relation $P_{min} = 3 n_*(t_{ref} + 1) \text{ ticks}$ in order to allow n_* agents to communicate in different temporal slots. The maximum value of P can be so selected to minimize T and therefore is a design choice.

Some assumptions were made in this paper which need revisiting when considering the application of the algorithm to detect anoxia or any other environmental phenomenon. Anoxia is a local phenomenon which spreads to its surroundings. In this paper, we simplified the dynamics of this phenomenon by assuming a constant spreading rate starting from a random patch and spreading radially outwards. In reality the periodicity of measurement and communication

as well as sensor placement needs to be modified according to the actual dynamics of the phenomenon being examined. In contrast to anoxia, not all events that are of interest have a spreading nature. While tragedies like oil spills move through the water body (Oudhuis and Tengblad, 2018), an aircraft crash (Kaiser, 2014) is an example of an event of interest that does not spread. In this paper we assume that an event can be detected easily using simple thresholding of some parameters. While this is true for the detection of anoxia, other methods such as machine learning (Bahre-pour et al., 2009) or other event specific techniques can be employed for the detection of events. Even in such cases, the algorithm presented in this paper can be employed for confirming the detected event with neighbors using minimal communication.

As part of future work, the algorithm presented here will be tested in the field for anoxia detection. There are many more energy saving techniques which can be employed but which are likely to require more than 1-bit communication. A study of energy consumed per bit can be conducted in order to establish the relation between energy consumed and the communication payload. This will enable the implementation of more sophisticated algorithms for event detection. Such a method might allow for a certain amount of node to node data exchange.

In conclusion, we presented a simple, bio-inspired, energy conserving event detection algorithm for the detection of the anoxia phenomenon in the lagoon of Venice using the aMussel robots. The suggested algorithm can go beyond detecting anoxia using the subCULTron system. Robotic or sensory systems with limited local communication can utilize the algorithm presented here to generate an alarm based on the number of swarm members that detected a local event.

Acknowledgements

This work was supported by EU-H2020 FET Project subCULTron, funded by the European Unions Horizon 2020 research and innovation programmer under grant agreement No 640967. Furthermore, this work was supported by the COLIBRI initiative at the University of Graz.

References

Bahre-pour, M., Meratnia, N., and Havinga, P. J. (2009). Use of AI techniques for residential fire detection in wireless sensor networks. In *AIAI Workshops*, pages 311–321.

Bonner, J. T. (1949). The social amoebae. *Scientific American*, 180(6):44–47.

Buck, J. and Buck, E. (1966). Biology of synchronous flashing of fireflies. *Nature*, 211:562–564.

Camazine, S., Denenbourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). Synchronized flashing among fireflies. pages 143–166. Princeton University Press, Princeton.

Cavagna, A., Cimorelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F., and Viale, M. (2010). Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, 107(26):11865–11870.

Cazangi, R. R., von Zuben, F., and Figueiredo, M. F. (2005). Beeadhoc: An energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*, pages 121–128, Washington, D.C., USA.

Chisholm, R. L. and Firtel, R. A. (2004). Insights into morphogenesis from a simple developmental system. *Nature reviews Molecular cell biology*, 5(7):531–541.

Debont, M., Jamshaid, K., Shihada, B., and Ho, P. (2012). Event localization in underwater wireless sensor networks using monitoring courses. In *2012 1st IEEE International Conference on Communications in China (ICCC)*, pages 769–774.

Durston, A. (1973). *Dictyostelium discoideum* aggregation fields as excitable media. *Journal of Theoretical Biology*, 42(3):483–504.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Kaiser, S. A. (2014). Legal considerations about the missing Malaysia airlines flight MH 370. *Air and Space Law*, 39(4):235–244.

Kaur, N. and Sood, S. K. (2017). An energy-efficient architecture for the Internet of Things (IoT). *IEEE Systems Journal*, 11(2):796–805.

Kim, L. H. and Follmer, S. (2017). UbiSwarm: Ubiquitous robotic interfaces and investigation of abstract motion as a display. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):66.

Koh, H. and Magee, C. L. (2006). A functional approach for studying technological progress: Application to information technology. *Technological Forecasting and Social Change*, 73(9):1061–1083.

Oudhuis, M. and Tengblad, S. (2018). BP and deepwater horizon: A catastrophe from a resilience perspective. In *The Resilience Framework*, pages 71–87. Springer.

Runca, E., Bernstein, A., Postma, L., and Silvio, G. D. (1996). Control of macroalgae blooms in the lagoon of Venice. *Ocean Coastal Management*, 30(2):235 – 257.

Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.

Seeley, T. D. (1992). The tremble dance of the honey bee: Message and meanings. *Behavioral Ecology and Sociobiology*, 31:375–383.

subCULTron (2015). Submarine cultures perform long-term robotic exploration of unconventional environmental niches. <http://www.subcultron.eu/>.

- Thenius, R., Moser, D., Varughese, J. C., Kernbach, S., Kuksin, I., Kernbach, O., Kuksina, E., Mišković, N., Bogdan, S., Petrović, T., Babić, A., Boyer, F., Lebastard, V., Bazeille, S., Ferrari, G. W., Donati, E., Pelliccia, R., Romano, D., Van Vuren, G. J., Stefanini, C., Morgantini, M., Campo, A., and Schmickl, T. (2018a). subCULTron - Cultural Development as a Tool in Underwater Robotics. In Lewis, P. R., Headland, C. J., Battle, S., and Ritsos, P. D., editors, *Artificial Life and Intelligent Agents*, pages 27–41. Springer.
- Thenius, R., Varughese, J. C., Moser, D., and Schmickl, T. (2018b). WOSPP-A wave oriented swarm programming paradigm. *IFAC-PapersOnLine*, 51(2):379–384.
- Witkowski, O. and Ikegami, T. (2016). Emergence of swarming behavior: Foraging agents evolve collective motion Based on signaling. *PloS one*, 11(4):e0152756.
- Zahadat, P. and Schmickl, T. (2016). Division of labor in a swarm of autonomous underwater robots by improved partitioning social inhibition. *Adaptive Behavior*, 24(2):87–101.
- Zhou, Z., Xing, R., Duan, Y., Zhu, Y., and Xiang, J. (2015). Event coverage detection and event source determination in underwater wireless sensor networks. *Sensors*, 15(12):31620–31643.

Collective Change Detection: Adaptivity to Dynamic Swarm Densities and Light Conditions in Robot Swarms

Mostafa Wahby¹, Julian Petzold¹, Catriona Eschke¹, Thomas Schmickl², and Heiko Hamann¹

¹Institute of Computer Engineering, University of Lübeck, Lübeck, Germany

²Artificial Life Lab, University of Graz, Graz, Austria

hamann@iti.uni-luebeck.de

Abstract

Robot swarms are known to be robust to individual robot failures. However, a reduced swarm size causes a reduced swarm density. A too low swarm density may then decrease swarm performance, that should be compensated by adapting the individual behavior. Similarly, swarm behaviors can also be adapted to changes in the environment, such as dynamic light conditions. We study aggregation of swarm robots controlled by an extended variant of the BEECLUST algorithm. The robots are asked to aggregate at the brightest spot in their environment. Our approach efficiently adapts this swarm aggregation behavior to variability in swarm density and light conditions. First, each robot individually monitors its environment continuously by sampling its local swarm density and perceived light condition. Second, we exploit the collaboration of robots by letting them share features of these measurements with their neighbors by communication. In extensive robot swarm experiments with ten robots we validate our approach with dynamically changing swarm densities and under dynamic light conditions. We find an improved performance compared to robot swarms without communication and without awareness of the swarm density.

Introduction

Swarm density has usually a significant impact on swarm performance in general and consequently on scalability (Hamann, 2018), for example, as observed in collective decision-making processes (Khaluf et al., 2017). Known advantages of swarm robotics, such as robustness and scalability, depend on a swarm's density and most swarm behaviors are sensitive to changes in swarm density. With large swarm sizes, more robots break, get lost, or may intentionally be removed. As an effect, the swarm density can decrease at runtime. Similarly, an operator may add robots to the system or the operating area may decrease over time, both effectively increasing the swarm density. Online changes in density have an impact on the swarm performance and adaptivity to these changes may prove to increase efficiency (Kernbach et al., 2013). Similarly, adapting the swarm behavior to dynamic environmental changes may be essential even for survival (Mallon et al., 2001) or may at least increase performance (Bonabeau et al., 1999).

In this paper, we make two main contributions that address two main challenges of designing swarm algorithms that adapt to dynamic densities and dynamic environments: (1) swarm density and environment are global features that can only be efficiently detected collectively; (2) for both, we face a tradeoff between filtering noise while reacting quickly to changes. In statistics, this problem is known as change detection (Picard, 1985) requiring to minimize the detection delay but also the cost of false alarms. Only here we require a decentralized implementation. Once a change in the swarm density is detected collectively, robots may be able to react individually. In the case of a too high density, robots may take themselves out of the game by leaving the operating area (Mayya et al., 2018). In the case of too low density, we get the more interesting challenge because we have to adapt the individual behavior appropriately, as studied here.

Previously, we have investigated the automatic adaptation of a robot swarm's behavioral features to the light settings in static environments (Wahby et al., 2016). This was based on an initial calibration phase by individual robots without exploiting collaboration between robots. However, certain environmental features may only be detected collectively (Schmickl et al., 2007) and it may be required to share information (Valentini et al., 2016). Here, we present an extension that adapts the robot behaviors to dynamic light conditions and exploits collaboration using communication.

Our approach is based on a site-specific aggregation behavior (Trianni et al., 2003; Soysal and Şahin, 2007). The robot swarm is supposed to aggregate at a spot of specific environmental conditions. Here, the aggregation spot is supposed to be the brightest spot. A robot can explore the whole arena, but it cannot know whether a yet unexplored brighter area exists or a previously explored area has increased its brightness meanwhile. Hence, a swarm robot should always stay explorative and as a sum of individual behaviors also the robot swarm should stay adaptive to changes.

As control algorithm we use BEECLUST (Schmickl and Hamann, 2011; Schmickl et al., 2008; Kernbach et al., 2009), a standard approach to aggregation in swarm robotics. BEECLUST is inspired by the behavior of young

honeybees (Szopek et al., 2013) that collectively find and select the warmest spot in a temperature gradient. According to the behavioral model they do that in a decentralized way and without any explicit communication. This behaviour was modeled and studied often in swarm robotics (Schmickl et al., 2009; Valentini and Hamann, 2015; Correll and Hamann, 2015), and a fuzzy-logic-based extension was proposed (Arvin et al., 2012, 2014), but with the original environmental feature of temperature being replaced by luminance. Here, the desired aggregation area is indicated by a light spot. We want a majority of the swarm to aggregate at the brightest light spot while a minority still stays adaptive as it keeps exploring the environment. We test a dynamic environment with changing light conditions (Schmickl et al., 2009). Initially, there are two peaks in the light distribution that we can see as local and global optima. It is essential that the swarm has a global awareness of the light distribution. A greedy search is not an option as the swarm needs to avoid both local optima and splitting between bright areas. The light conditions change three times during an experiment requiring three switches of the swarm’s aggregation spot. The swarm robots need to explore the arena continuously and re-allocate quickly.

We implemented our proposed extension to the BEECLUST algorithm on Thymio II robots (Riedo et al., 2013) and tested our approach in 25 robot experiments of 24+ minutes. We analyze the obtained results and find that the adaptation to dynamic environments is effective, that sharing gathered information between robots by communication is advantageous, and that our adaptivity approach to dynamic swarm densities is also effective.

A Short History of BEECLUST

BEECLUST is a bio-inspired aggregation algorithm, derived from the navigation behavior of young honeybees in temperature gradients within beehives (Schmickl and Hamann, 2011; Schmickl et al., 2008; Kernbach et al., 2009; Hamann, 2018). The original behavioral model of bees is translated to a swarm robot controller where a light gradient is considered instead of a temperature gradient. Experimenting with light simplifies sensing compared to temperature measurements (temperature sensors are less sensitive, are fragile, and have longer latencies). The original BEECLUST algorithm is defined as follows:

1. A robot moves forward.
2. If a robot approaches the arena borders, it turns away to a random direction and continues with step 1.
3. If a robot meets another robot, it stops and measures the local luminance. The higher the luminance, the longer its waiting time is (i.e., the amount of time the robot stays stationary).
4. After the waiting time is over, the robot turns away to a random direction and continues with step 1.

The BEECLUST algorithm produces a characteristic

swarm-level behavior: Initially several small robot clusters form across the environment. When the waiting time has elapsed, robots leave their cluster again and, hence, clusters may disappear. Clusters in brighter areas persist longer and have a higher probability to increase in size by being approached by additional robots. There is a positive feedback loop on cluster growth because bigger clusters have a bigger probability to grow. In addition, robots within a big cluster may not be able to leave after their waiting time has elapsed because they are physically blocked. This contributes as additional positive feedback. Finally, one or a few big clusters form. A few robots still leave the cluster, explore the possibly dynamic environment, and may come back, joining the cluster again.

The key point for the BEECLUST algorithm to succeed is the correct mapping of environmental conditions to the resulting waiting times. Proper waiting times depend on environmental features, for example, the light distribution, and the swarm density (i.e., the number of robots in a given area). A lower swarm density means that robots are less likely to approach a robot cluster, hence, longer waiting times would compensate for this effect and ensure that clusters still have a good chance of growing. High swarm densities may result in a limited competition between clusters, such that clusters at suboptimal positions do not dissolve as quickly as desired. Similarly, bright robot arenas may in average give too long waiting times and dark arenas too short waiting times.

Schmickl and Hamann (Schmickl and Hamann, 2011) defined the waiting time as a sigmoid function

$$w(I) = \frac{w_{\max} I^2}{I^2 + c}, \quad (1)$$

where w_{\max} is a predetermined maximum waiting time in seconds, I is the locally measured luminance, and a constant $c = 4.86 \times 10^5$. The swarm’s clustering performance depends on the constants w_{\max} and c . If they are set incorrectly, then robots do not cluster at all or cluster at undesired places. Similarly, these values need to be adapted dynamically in the case of a dynamic light gradient or dynamic swarm densities.

In a previous study, we have introduced an approach to automatically adapt the waiting function to any light configuration by setting the maximal waiting time w_{\max} to a big value which is then scaled according to the minimum light intensity I_{\min} and maximum light intensity I_{\max} measured by the robot in the arena (Wahby et al., 2016). This information is then used to rescale the light intensity \bar{I} to the interval $[0, 1]$ by

$$\bar{I} = \frac{I - I_{\min}}{I_{\max} - I_{\min}}. \quad (2)$$

An additional initial calibration phase was required, where the robots explore the arena and collect illumination samples in regular intervals. For the used arena dimensions ($3 \times 2.5 \text{ m}^2$), a calibration period of three minutes was found to

be sufficient for estimating I_{\min} and I_{\max} . Then, the robots adapt their waiting time individually using \bar{I} instead of I in eq. 1 and start the aggregation phase. This approach succeeds in the automatic adaption to surrounding light conditions. However, if light conditions change during the productive phase of the experiment, then the robot swarm may fail to cluster at the brightest spot. Also, without communication each robot has its own estimation of the light conditions that may lead to inconsistent waiting times and aggregation behavior within the swarm.

Boosting BEECLUST for Dynamic Swarm Densities and Dynamic Environments

We extend the BEECLUST algorithm by making the robots aware of the swarm density and by allowing for more collaboration between robots in order to share measurements (see Fig. 1). When a robot moves, it measures the average time taken between robot-to-robot-encounters t_r , to get an estimate of the swarm density. Instead of a calibration phase, the robots collect measurements of the light during the actual execution of the experiment. Each robot measures and stores a light intensity sample every second. Robots share features of these measurements ('communicate parameters', Fig. 1) to collectively perceive the dynamic environment and to adapt their waiting times accordingly. Not every robot has to travel individually to all relevant places in the arena to collect appropriate measurements.

According to kinetic gas theory, the mean free path $\ell = (n\sigma)^{-1}$ is the average distance traveled by a moving particle between successive collisions with other moving particles. n is the number density of molecules (ideal gas law) and $\sigma = \pi(2r)^2$ is the effective cross-sectional area for spherical particles with radius r . This indicates that, in swarm robotics, the density of moving robots on a fixed area may scale similarly, however, inversely proportional to the average robot-to-robot-encounter time interval t_r (Kernbach et al., 2013). For example, if we halve the swarm size, then the robots measure on average a 100% increase of time t_r , that is, they should also increase their waiting time by 100% for effective clustering.

Therefore, we include the average time between two robot encounters t_r into the waiting function as a product

$$w(\bar{I}) = \frac{t_r}{12} \frac{w_{\max} \bar{I}^2}{\bar{I}^2 + c}, \quad (3)$$

which substitutes the original waiting function given in eq. 1. We have to introduce another constant here as we normalize t_r by 12. This corresponds to the average value measured for t_r (in seconds) in our arena. By normalizing t_r we can keep the value of the maximal waiting time w_{\max} as it is and preserve its interpretability.

Robots share their measured values by communication with other swarm members to ensure the consistency of

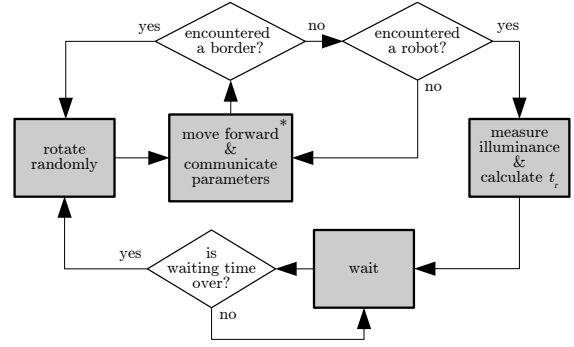


Figure 1: Finite state machine describing the adaptive BEECLUST algorithm.

their behaviors and to enhance the swarm's collective performance. Robots locally exchange their adaptivity parameters (I_{\min} , I_{\max} , and t_r) at every time they are in proximity of other robots. After receiving parameters from a neighbor, a robot checks the new I_{\min} and I_{\max} and computes a new average t_r . According to the robots' *memory lifetime*, the measurements are considered outdated after a certain amount of time (90 seconds). The *memory lifetime* introduces a trade-off between the adaptivity and speed.

Robot and Experiment Setup

Thymio II is a small mobile robot designed for children (Riedo et al., 2013), it measures approximately $11 \times 11 \times 5 \text{ cm}^3$ in width, depth, and height. Its programming is based on ASEBA (Magnenat et al., 2011), which is an open source modular architecture for event-based robot control. The robot is based on a differential wheeled drive, and is equipped with many sensors, such as seven horizontal infrared sensors, two ground infrared sensors, a 3-axis accelerometer, a remote control infrared receiver, and a temperature sensor. Thymio II robots also provide attachment knobs for Lego™ bricks on top, which allows an easy installation of hardware extensions. The infrared sensors are used for kin recognition, ensuring a robot makes a transition to the stopped state only when another robot is detected at a distance of 5 cm or less. A second purpose of the infrared sensors is robot-to-robot communication. Two Thymios can communicate over distances of up to about 25 cm, as long as at least one of the seven IR emitters is within line of sight of a horizontal sensor of the other robot. For bi-directional communication this alignment has to be reciprocal.

In our experiments, the robots need to communicate the extremes of their memorized light intensity measurements I_{\min} and I_{\max} as well as their swarm density estimation parameter t_r . However, the angular displacement of the horizontal sensors does not provide a full circumferential view. The sides of a Thymio robot have no infrared sensors, (see Fig. 2) leaving two large blind areas without

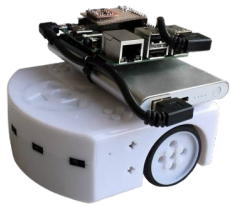


Figure 2: Thymio II robot with a Raspberry Pi, an ambient light sensor, and a power bank attached to the Lego™ attachment knobs on top of the robot.

possible communication. To increase the chances for communication, we rotate the robots on the spot while in the ‘stopped state’ which vastly increases frequency of successful communications of waiting robots. Thymios can send 11 bit messages at 100 ms intervals, with one bit being used by the firmware itself leaving 10 bits for user data. The first two bits determine the type of the communicated parameter (00 for I_{\min} , 01 for I_{\max} , and 10 for t_r), and the parameter value is encoded in the remaining eight bits as integer on the interval $[0, 255]$. In the case a robot receives two subsequent messages of the same type and value, the earlier message is discarded to avoid duplicates. The robots don’t relay received messages and delete them after the *memory lifetime* has been reached. This is an important feature that ensures robustness because it prevents spreading of faulty or outdated information (Schmickl et al., 2006). It also acts as first line of defense against Byzantine robots.

According to our experiment setup, the robots go through different phases of light intensity configurations (see Fig. 4) of at least six minutes each to simulate a dynamic environment. Therefore, we set the *memory lifetime* to 90 seconds, which is a good compromise between quick adaptation to such quick changes and still relying on averaged measurements. Although our motivation is to maximize the degree of adaptivity in our system, a few parameters, such as as the *memory lifetime* and the normalization of t_r by 12 in Eq. 1, are required. Certainly, shorter *memory lifetimes* may be advantageous in even more dynamic environments and longer intervals may help in environments that change slowly. However, we assume that these environmental features are unknown a priori and cannot be anticipated.

Following our earlier approach (Wahby et al., 2016), we extend the capabilities of the Thymio II robots by a hardware extension consisting of a Raspberry Pi, an ambient light sensor (TSL45315), a power bank, and a Lego™ attachment plate (see Fig. 2). The ambient light sensor is interfaced with the Raspberry Pi through the I²C bus and it provides an output range that corresponds to environmental conditions ranging from 3 lux to 2.2×10^5 lux. The Raspberry Pi connects to the Thymio II through the USB port and the D-Bus



Figure 3: The 3.0×2.5 m² arena is bound by a white line on the ground (duct tape) emulating virtual walls that can be detected by the robots’ ground sensors. $N = 10$ robots are initially placed in an evenly spaced grid.

interface, and considers the robot as a sensor/actuator unit¹.

In order to have comparability to previous studies based on other versions of the BEECLUST algorithm as in (Wahby et al., 2016), (Schmickl et al., 2008), and (Schmickl et al., 2009), we constructed an arena of size 3.0×2.5 m² (see Fig. 3). This arena is bounded by a white line on the ground emulating virtual walls that can be detected by the robots’ ground sensors. In the case a ground sensor detects a virtual wall, the robot rotates into the opposite direction for a random amount of time between 0.5 and 2.2 seconds, before it continues moving forward. The virtual walls allow for using the horizontal proximity sensing exclusively for kin recognition, which increases the reliability.

When moving forward, the robots operate in full speed of 20 cm/second. Two projectors are fixed to the ceiling above the arena, that allows the projection of two light spots (each at the middle of one half of the arena), in different intensity configurations (see Fig. 4). With this setup we can demonstrate different aggregation behaviours while keeping the configuration simple and reproducible. At the beginning of each experiment, ten robots are initially placed in an evenly spaced grid (see Fig. 3). All the experiments are video captured by a digital camera, which is also fixed to the ceiling. We divided each experiment into non-overlapping intervals of five seconds. For every interval, we counted the maximum number of clustered robots (i.e., in the stopped state) at each of the two light spots. Then the median, minimum and maximum values at every four intervals (from $n = 5$ repetitions) are used to construct the graphs in Figs. 5 and 6.

Results

We tested four variants of the BEECLUST algorithm: *offline* (with initial calibration but no online illumination sampling and no communication, similar to our previous approach in (Wahby et al., 2016)), *online non-communicating*

¹source code:

https://git.iti.uni-luebeck.de/CEschke/A_BEE.git

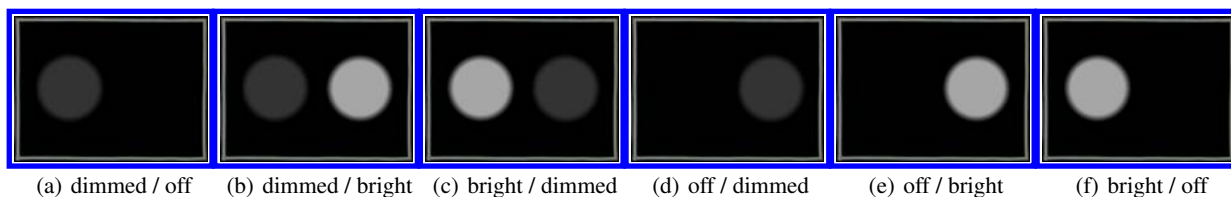
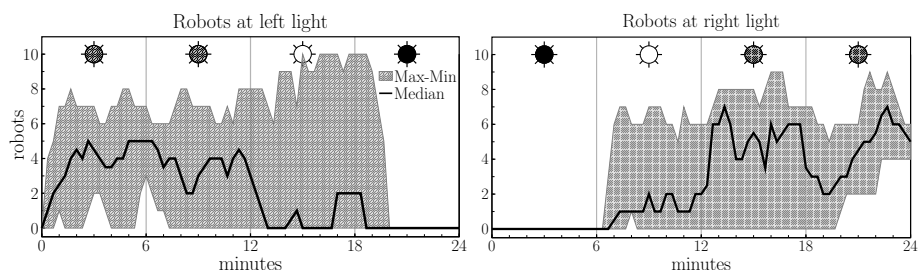
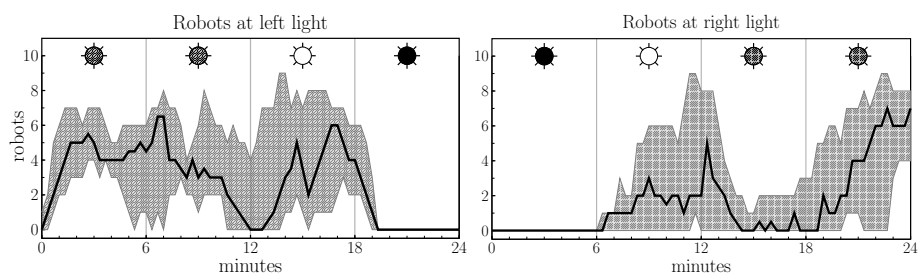


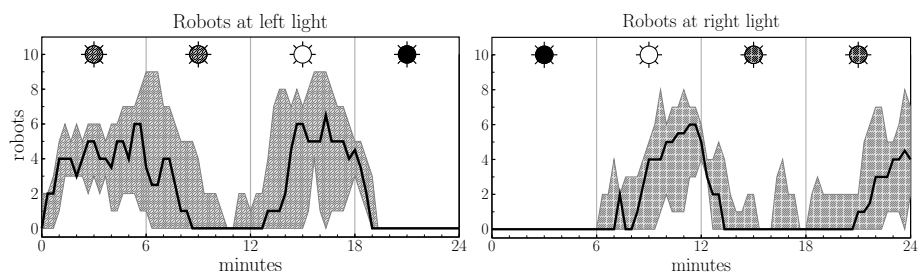
Figure 4: Light configurations of the arena during the different phases of the two kinds of experiments. In the light adaptivity experiments (our first three sets of experiments) the light configuration sequence (a) \rightarrow (b) \rightarrow (c) \rightarrow (d) is used. For the swarm density adaptivity experiments (the last two set of experiments) (e) followed by (f) is used. The projector operates at 100% brightness to project bright spots, which is measured as about 30 lux by the light sensor of our robots. 20% brightness is projected for dimmed spots, which is measures as about six lux. Both light spots are about 70 cm in diameter.



(a) offline approach (initial calibration phase, no online illumination sampling and no communication; control experiment).



(b) online non-communicating approach (online illumination sampling but no communication).



(c) online communicating without density estimation approach (with online illumination sampling, swarm density sampling, and communication).

Figure 5: Static density runs, analysis of practical robot experiments using $N = 10$ Thymio robots and the different variants of the BEECLUST algorithm in consecutive phases of light settings. Each of the five rows represents the results of $n = 5$ repetitions and gives the number of aggregated robots at the left and right area. The shown data is obtained by processing the time series in 20 second segments, extracting the maximum, minimum, and median number of robots observed over all $n = 5$ repetitions within that segment. The sun-like symbols indicate the dynamically changing light intensity at the corresponding projected light spot. A white sun indicates a bright light spot, a gray sun indicates a dimmed light spot, and a black sun indicates no light.

(with online illumination sampling but no communication), *online communicating without density estimation* (with online illumination sampling and communication), and *online communicating with density estimation* (with online illumination sampling, swarm density sampling, and communication). The first three BEECLUST variants were tested in a first experiment setup with dynamic light conditions and static density. There are four different consecutive phases of light settings, six minutes each (see Fig. 4). In the second experiment setup we tested the adaptivity to dynamic swarm densities using the *online communicating without density estimation* BEECLUST variant as control experiment and the *online communicating with density estimation* BEECLUST variant. This second experiment setup consists of only two phases with a slightly longer duration of 14 minutes each, where the swarm is manually halved from $N = 10$ robots to $N = 5$ robots after 14 minutes. Each experiment is repeated $n = 5$ times (totaling to 25 robot experiments)². We test for statistical significance by comparing numbers of clustered robots pooled over the whole phase from all $n = 5$ repetitions using the Wilcoxon signed-rank test ($p \leq 0.001$).

In Fig. 5(a) we give the results for the *offline* BEECLUST variant as control experiment. Before the experiment starts, the robots adapt their waiting time individually to the light configuration in the first phase (*dimmed / off*). Here, the robots learn that the dimmed spot is the brightest possible spot, and in case they encounter a brighter spot in the later phases, the maximum waiting time computed during the initial calibration is still considered. Focusing on the medians given in Fig. 5(a), we notice that during the first two minutes in the first phase, the number of robots clustering at the dimmed spot increases about linearly. Then a median of about five is reached and only minor fluctuations occur due to scouting behaviors until the end of the phase. In the next phase (*dimmed / bright*), a bright spot emerges at the right side of the arena. However, clusters of only two robots form at the bright spot, because the robots are assured by the initial calibration phase that they are at the brightest spot already. In this and the third phase, following the minimum, maximum, and median values of clustering robots, it is obvious that the robots are unable to locate the brighter spot until the bright spot disappears in the last phase.

In Fig. 5(b) we give the results for the *online non-communicating without density estimation* variant, that is, robots do not require an initial offline calibration phase but sample light measurements online. Similar to the above experiment, the robots form a cluster at the dimmed spot during the first phase. When the bright spot emerges at the right side of the arena, the cluster of about five robots at the dimmed spot required almost the whole duration of the phase to dissolve and to form again at the bright spot. Possibly robots are assured by locally collected informa-

tion that they are at a bright spot already, while the scouting robots fail to communicate the information about the new bright spot. From the results of the remaining two phases it is obvious that this variant outperforms the *offline* approach in terms of adapting to dynamic light conditions (significantly more clustered robots during both phases, $p \leq 0.001$). It deals relatively well with a required aggregation at the dimmed spots but fails to aggregate many robots quickly at the bright spots (second phase, right; third phase, left).

In Fig. 5(c) we show the results for testing the *online communicating without density estimation* variant, that is, we test whether we can improve performance once we allow the robots to communicate their light adaptivity parameters (I_{\min} and I_{\max}). Similar to the previous two setups, the robots form a cluster at the dimmed spot during the first phase. When the bright spot emerges at the right side of the arena, the cluster at the left side quickly dissolves and the robots form a new cluster at the bright spot. The scouting robots probably detect the bright spot in the arena and collect light intensity samples of bigger values (i.e., acquired high I_{\max}). They successfully share their knowledge with the robots clustered at the dimmed spot. This way the aggregated robots notice that they are currently clustering at a local optimum without having to explore the arena themselves. This allows to quickly dissolve the cluster. We find that this variant is performing significantly better than the *online non-communicating without density estimation* variant when bright light spots emerge in new locations. In the third phase, the cluster at the dimmed spot quickly dissolves and forms again at the bright spot. In the last phase, the dimmed spot is at the right side of the arena. Since the robots can retain an I_{\max} value for up to three minutes after it was acquired by another robot and then communicated, the first half of this phase shows no reliable cluster formation. Probably the waiting times of the robots are still scaled for an optimum with a bright spot and not a dimmed spot. During the last three minutes of the experiment, however, they form a cluster at the right side. Especially the performance of the *online communicating without density estimation* variant during phase two and three is a clear improvement over the *online non-communicating without density estimation* variant. The performance increase is statistically significant (Wilcoxon signed-rank test, $p \leq 0.001$).

Finally, we test our BEECLUST approach in a setting with dynamic swarm densities. We compare the performance of our *online communicating with density estimation* BEECLUST approach, see Fig. 6(b), to a control experiment using the *online communicating without density estimation* variant, see Fig. 6(a). In the first phase (*off / bright*) we start with $N = 10$ robots. Big robot clusters of up to nine or ten robots are formed at the bright spot. At the beginning of the second phase, the bright spot is switched to the left side and half of the robot swarm (five robots) is randomly manually removed from the arena, only $N = 5$ robots remain. The

²video available online: <https://vimeo.com/271398596>

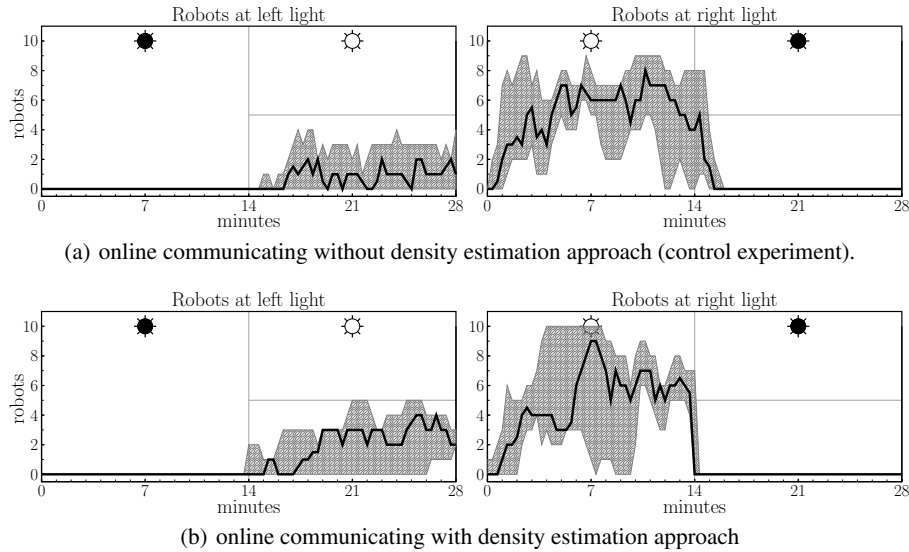


Figure 6: Dynamic density runs, analysis of practical robot experiments using initially $N = 10$ Thymio robots, then the swarm size is manually reduced to $N = 5$ robots at $t = 14$ min. To identify the advantage of the density estimation approach, only the 2nd phase ($14 < t \leq 28$ min) is relevant. Significantly more robots aggregate at the left light for $t > 14$ min using density estimation. $n = 5$ repetitions per row, number of aggregated robots at the left and right area, dynamically changing light intensity.

clusters quickly dissolve as usual due to the shared knowledge from scouts. The *online communicating with density estimation* variant succeeds in forming robot clusters of up to five robots (i.e., the complete swarm) at the bright spot after about seven minutes. The *online communicating with density estimation* variant outperforms the control experiment because significantly more robots cluster during the second phase of the experiment ($p < 0.001$). In these two experiments light intensity was not changed, hence, clustering at the opposite side of the arena was not delayed due to the robots' local history. The observed behavior in this setup indicates the ability of our approach to adapt to the dynamic change in the swarm size and successfully aggregate the swarm at the desired area. We have analyzed the measured robot-to-robot encounter time t_r of an arbitrary robot. In the first phase, t_r is about 13 seconds in average. In the second phase, it increases to about 18 seconds in average. Hence, the robots successfully detect the change in swarm density and adapt their behavior. All experiment data, photos, and videos are available online (Wahby et al., 2019).

Discussion and Conclusion

We have extended the BEECLUST algorithm with a method to adapt waiting times to dynamic swarm densities. Times between robot-to-robot encounters and environmental features (information about the light distribution) are measured and gathered by each robot, shared by communication, and aggregated. This way we contribute a solution to the challenge of collectively detecting global features and

their changes, that is, collective change detection. We also found parameters to balance the tradeoff between minimal detection delay and false alarms. We have successfully verified our approach in 25 swarm robot experiments that show a clear improvement in performance over the non-adaptive variant. Adaptivity to swarm density is a relevant feature in swarm robotics with possible impact on the swarm's scalability properties. Similarly, adaptivity to environmental features is crucial for the operation in dynamic environments. Both properties enable long term autonomy and are essential in real world applications (e.g., environmental monitoring).

An interesting extension in future work is to combine collective perception (Schmickl et al., 2007) with other scenarios of swarm robotics to allow for adaptivity to even more features. Other options for future work include a more intensive study of adaptation to swarm density and scalability. A better control of when data shared with other robots is outdated could be achieved by adding time stamps. However, the Thymio's infrared-based communication may be too primitive and WiFi or scalable protocols could be used instead. Finally, we could try to protect our adaptivity process against malicious robots that intentionally or by accident spread wrong measurements. Methods of error detection and correction in distributed robot systems may be useful (Tarapore et al., 2015).

References

Arvin, F., Turgut, A. E., Bazyari, F., Arıkan, K. B., Bellotto, N., and Yue, S. (2014). Cue-based aggregation with a mobile

- robot swarm: a novel fuzzy-based method. *Adaptive Behavior*, 22(3):189–206.
- Arvin, F., Turgut, A. E., and Yue, S. (2012). Fuzzy-Based Aggregation with a Mobile Robot Swarm. In *Swarm Intelligence (ANTS'12)*, volume 7461 of *Lecture Notes in Computer Science*, pages 346–347, Berlin. Springer.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford Univ. Press, New York.
- Correll, N. and Hamann, H. (2015). Probabilistic Modeling of Swarming Systems. In Kacprzyk, J. and Pedrycz, W., editors, *Springer Handbook of Computational Intelligence*, pages 1423–1431. Springer.
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Springer.
- Kernbach, S., Häbe, D., Kernbach, O., Thenius, R., Radspieler, G., Kimura, T., and Schmickl, T. (2013). Adaptive collective decision-making in limited robot swarms without communication. *The International Journal of Robotics Research*, 32(1):35–55.
- Kernbach, S., Thenius, R., Kornienko, O., and Schmickl, T. (2009). Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic Swarm. *Adaptive Behavior*, 17:237–259.
- Khaluf, Y., Pinciroli, C., Valentini, G., and Hamann, H. (2017). The Impact of Agent Density on Scalability in Collective Systems: Noise-Induced versus Majority-Based Bistability. *Swarm Intelligence*, 11(2):155–179.
- Magenat, S., Régnard, P., Bonani, M., Longchamp, V., and Mondada, F. (2011). ASEBA: A Modular Architecture for Event-Based Control of Complex Robots. *IEEE/ASME transactions on mechatronics*, 16(2):321–329.
- Mallon, E. B., Pratt, S. C., and Franks, N. R. (2001). Individual and Collective Decision-Making During Nest Site Selection by the Ant *Leptothorax albipennis*. *Behavioral Ecology and Sociobiology*, 50:352–359.
- Mayya, S., Pierpaoli, P., and Egerstedt, M. (2018). Voluntary retreat for decentralized interference reduction in robot swarms. *arXiv preprint arXiv:1812.02193*.
- Picard, D. (1985). Testing and estimating change-points in time series. *Advances in Applied Probability*, 17(4):841–867.
- Riedo, F., Chevalier, M., Magneat, S., and Mondada, F. (2013). Thymio II, a Robot that Grows Wiser with Children. In *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on*, pages 187–193. IEEE.
- Schmickl, T. and Hamann, H. (2011). BEECLUST: A Swarm Algorithm Derived from Honeybees. In Xiao, Y., editor, *Bio-inspired Computing and Communication Networks*. CRC Press.
- Schmickl, T., Hamann, H., Wörn, H., and Crailsheim, K. (2009). Two Different Approaches to a Macroscopic Model of a Bio-Inspired Robotic Swarm. *Robotics and Autonomous Systems*, 57(9):913–921.
- Schmickl, T., Möslinger, C., and Crailsheim, K. (2006). Collective perception in a robot swarm. In *International Workshop on Swarm Robotics*, pages 144–157. Springer.
- Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). Collective Perception in a Robot Swarm. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *Lecture Notes in Computer Science*, Heidelberg/Berlin, Germany. Springer-Verlag.
- Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: cooperative decision making Based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.
- Soysal, O. and Şahin, E. (2007). A Macroscopic Model for Self-organized Aggregation in Swarm Robotic Systems. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *Lecture Notes in Computer Science*, pages 27–42, Berlin, Germany. Springer-Verlag.
- Szopek, M., Schmickl, T., Thenius, R., Radspieler, G., and Crailsheim, K. (2013). Dynamics of Collective Decision Making of Honeybees in Complex Temperature Fields. *PLoS ONE*, 8(10):e76250.
- Tarapore, D., Lima, P. U., Carneiro, J., and Christensen, A. L. (2015). To err is Robotic, to Tolerate Immunological: Fault Detection in Multirobot Systems. *Bioinspiration & biomimetics*, 10(1):016014.
- Trianni, V., Groß, R., Labella, T. H., Şahin, E., and Dorigo, M. (2003). Evolving Aggregation Behaviors in a Swarm of Robots. *Lecture Notes in Artificial Intelligence*, 2801:865–874.
- Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016). Collective Perception of Environmental Features in a Robot Swarm. In Dorigo, M., Birattari, M., Li, X., López-Ibáñez, M., Ohkura, K., Pinciroli, C., and Stützle, T., editors, *Swarm Intelligence*, pages 65–76, Cham. Springer International Publishing.
- Valentini, G. and Hamann, H. (2015). Time-Variant Feedback Processes in Collective Decision-Making Systems: Influence and Effect of Dynamic Neighborhood Sizes. *Swarm Intelligence*, 9(2-3):153–176.
- Wahby, M., Petzold, J., Eschke, C., Schmickl, T., and Hamann, H. (2019). Supplementary material online. <https://doi.org/10.5281/zenodo.1419277>.
- Wahby, M., Weinhold, A., and Hamann, H. (2016). Revisiting beelust: Aggregation of swarm robots with adaptiveness to different light settings. *EAI Endorsed Transactions on Collaborative Computing*, 2(9).

Droplet-based synthetic biology: chemotaxis and interface with biology

Silvia Holler¹ and Martin Michael Hanczyc^{1,2}

¹Laboratory for Artificial Biology, Department of Cellular, Computational and Integrative Biology (CIBIO)
University of Trento, 38123, Trento, Italy

²Chemical and Biological Engineering, University of New Mexico, MSC01 1120, Albuquerque, NM 87131-0001, USA
silvia.holler@unitn.it

Abstract

Liquid droplets possess some life-like behaviors and have been the subject of artificial life studies. Life-like behaviors such as fission, fusion and movement can be artificially recreated exploiting highly simplified chemical systems. Recently we showed that droplet-based chemotactic systems can be interfaced with biological systems (1). We developed a chemotactic droplet able to move light cargos such as hydrogel alginate capsules embedded with living cells as a transporter. We transported efficiently and in a sterile way a few types of bacteria and yeast, and we are now modifying our protocols to transport efficiently human cell lines. We recently discovered that some eukaryotic cell lines release surfactants when placed in our artificial transport system, thereby reinforcing the interface between the artificial and living systems. This is an example of not only how the interface between artificial life and biological life could be designed but how the one system can augment the other. In this case the living system produces the surfactants that the droplet needs for cargo transport and the artificial system provides the transport for the otherwise sessile mammalian cells.

Introduction

Protocell systems are examples of bottom-up synthetic biology. A leading property of cells and living organisms is the ability to move. Motile protocells can be created using simple chemical systems: for example, droplet of oil in water or droplet of water in oil. Lively droplets of water in oil were first described by Otto Bütschli in 1892 (2). He used alkaline water droplets in olive oil to initiate a saponification reaction. This simple protocell system recreated an entity that moved and seemed to behave like an amoeba. Since then many researchers have been developing oil droplet systems as models of living systems (3), (4). For example, the research of the group of Hagan Bayley in Oxford created 3D customized patterns of water droplets in oil with stable lipid bilayers forming the droplet-droplet interfaces as mimics of living tissues (5). In addition these networks of droplets with integrated porins can show current transmission. Each droplet in this system can be complemented with cell-free expression systems controlled by light activating protein expression (6). In this way they demonstrated that

life-like behaviours such as current transmission and protein expression can be activated even in water-in-oil droplets.

We mainly focused our work on chemotactic 1-decanol motile droplets. Chemotaxis is defined as a stimulated migration towards an increasing (or decreasing) chemical gradient, and 1-decanol droplets, formed in an aqueous medium containing decanoate at high pH, show chemotaxis when a chemical gradient is placed in the external aqueous environment. Droplets using such chemical gradients are able to solve 3d mazes, displaying a rudimentary artificial intelligence. This kind of movement can be compared to already well-described system of eukaryotic chemotaxis. For example, *Dictyostelium amoebae* migrates along an increasing concentration of cyclic adenosine-3',5'-monophosphate (cAMP) (7). Cejkova et al. showed in 2014 1-decanol chemotaxis towards a salt source (8). This system works even in mazes (9) and can be exploited to transport non living (9) objects. There is a challenge and benefit to begin to interface living and artificial systems to exploit potential synergies, increase robustness or increase the functionalities of both systems. We then attempted to interface the purely artificial decanol droplet system with living cells.

We therefore show how to interface the purely artificial decanol droplet system with living cells, preserving the function of both systems.

Protocell-cell transport system

We developed our artificial chemotactic system to make it compatible with natural living systems by creating a partially hydrophobic alginate capsule as a protective unit that can be precisely embedded in a droplet, transported along chemical gradients and deposited. This system was able to transport *Escherichia coli*, *Bacillus subtilis* and *Saccharomyces cerevisiae*. Both bacteria survived the transport. However, yeast survived but not in a consistent and repeatable way. The droplet containing a capsule with live cargo could be manipulated with salt gradients several times with the capsule remaining stably attached to the droplet. In addition, several capsules can be stably fixed to a single decanol droplet. For a video of this system, see:

<https://www.youtube.com/watch?v=zCB2bPhFoCI>. We afterwards conceived the idea to develop this system to transport mammalian eukaryotic cells. To do this we needed to evolve the droplet chemotaxis system under conditions more conducive to physiological environments. We decreased the pH from 12 to 7 and tried to transport A549 cells inside our alginate capsules. We found that A549 cells can be encapsulated in alginate hydrogels and survive. When in capsules incubated in growth medium DMEM, the cells survive and secrete into their environment some compounds that lower the surface tension and act as surfactants. The water phase in which capsules are incubated shows, if analyzed using pendant drop method with a tensiometer, a reduction in surface tension (60-55 mN/m) if compared to water (72 mN/m) and can be used in our artificial system as chemotactic water phase. Some of the molecules secreted by the cells modulate the surface tension of the alginate capsule. This surface modification allows the normally hydrophilic hydrogel capsule to associate efficiently and for an extended time with the hydrophobic 1-decanol droplets (up to 1 hour in the case of cell culture water phase mixed 1:1 with water). The secretion of surfactants is probably due to the mucus secreting phenotype transition of A549 cells. This transition leads to surfactant release in the water phase in which the capsules are incubated. This surfactant secretion is shown only when A549 are in capsules and this demonstrates that the integration of the biological system (A549) with the artificial one (capsules) can be exploited to increase the functionalities of the system. In addition, this association is selective for live cells as dead or non-proliferating cells do not produce the required amount of surfactant. The capsule containing live cells can then be transported using chemical gradient to a specific location and dropped though the addition of a water phase with concentrated surfactant.

We show that chemotactic droplet systems can interface with biological systems and transport live cells in petri dishes, but other scenarios are possible. Active droplets containing cells could be applied in smaller environments such as microfluidic chips, leading to the implementation of next generation technologies for cell screening (e.g. live vs dead). Chemotaxis systems and alginate capsules are inexpensive and easy to manipulate and could be applied more widely. For example, alginate capsules could be exploited to delivery bacteria or enzymes for improved bioremediation (10). Droplets determine the transport to locations not accessible by human hands and capsules could protect bacteria from harsh environmental conditions (11). This same approach could be used for environmental planning, to test possible bacterial/enzymatic/chemical treatments, transported by capsules and droplets, on systems with reduced scale. Therefore we expect a certain degree of societal impact through the ongoing development of this artificial life technology.

Acknowledgements

This work was financially supported in part by the European Commission FP7 Future and Emerging Technologies Proactive 611640 (EVOBLISS) and by the European Union's Horizon 2020 research and innovation program under grant agreement 824060 (ACDC).

References

- Holler, S. Porcelli, C. Ieropoulos, I. A. and Hanczyc, M. M. (2018) Transport of Live Cells Under Sterile Conditions Using a Chemotactic Droplet. *Sci Rep* 8:8408
- Bütschli, O. (1892) Untersuchungen über microscopische Schaume und das Protoplasma. Leipzig.
- Armstrong, R. and Hanczyc, M.M. (2013) Bütschli dynamic droplet system. *Artif Life*. Summer-Fall 19(3-4) 331-46
- Caschera, F. Rasmussen, S. and Hanczyc, M.M. (2013) An Oil Droplet Division-Fusion Cycle *ChemPlusChem* 78 52?54
- Bayley, H. Cronin, B. Heron, A. Holden, M.A. Hwang, W. Syeda, R. Thompson, J. and Wallace, M. (2008) Droplet interface bilayers *Mol Biosyst.* 4(12) 1191?1208
- Booth, J. M. Restrepo Schild, V. Box, S. J. and Bayley, H. (2017) Light-patterning of synthetic tissues with single droplet resolution. *Sci. Rep.* 7, 9315
- Ševčíková, H. Čejková, J. Krausová, L. Příbyl, M. Štěpánek, F. and Marek, M. (2010) A new traveling wave phenomenon of *Dictyostelium* in the presence of cAMP. *Physica D: Nonlinear Phenomena* 239 879-888
- Čejková, J. Novak, M. Štěpánek, F. and Hanczyc, M.M. (2014) Dynamics of chemotactic droplets in salt concentration gradients. *Langmuir* 30 11937-11944
- Čejkova, J. *et al.* (2016) Chemotaxis and chemokinesis of living and non-living objects. Springer, *Advances in Unconventional Computing* 256-260
- Pieper, D. H. and Reineke, W. (2010) Engineering bacteria for bioremediation. *Curr. Opin. Biotechnol.* 11(3) 263-270
- Islam, M. A. Cheol-Heui, Y. Yun-Jaie, C. and Chong-Su C. (2010) Microencapsulation of Live Probiotic Bacteria. *J. Microbiol. Biotechnol.* 20(10) 1367-1377

Autocatalysis in a Hierarchically Organized Inorganic Chemical Network

Cole Mathis¹, Haralampos Miras¹ and Leroy Cronin¹

¹School of Chemistry, University of Glasgow, Glasgow G12 8QQ, UK.
Lee.Cronin@Glasgow.ac.uk

Abstract

One of the salient features of living systems is presence of autocatalytic chemical reaction networks. Here we present a stochastic model of an inorganic autocatalyst, which is derived directly from empirical results. Using the model, we can explore the emergence of autocatalysis and its consequences on the larger, hierarchical, chemical network. This model provides a useful tool to study the emergence and organization of autocatalytic chemical networks and the effect autocatalysis has on the global system dynamics.

A characteristic feature of living systems is the existence of autocatalytic cycles, which enables the exponential growth of individuals and thus evolutionary dynamics (Cronin & Walker 2016, Kauffman 1992). Accordingly, a primary goal of origins of life and artificial life research is the design and synthesis of autocatalytic chemical reaction networks (Ashkenasy et al, 2017, Ashkenasy et al 2004, Kauffman 1992). Here we demonstrate a stochastic kinetic model of network autocatalysis based on an inorganic chemical system (Miras et al 2010). This model provides mechanistic insights into how autocatalytic cycles can be stabilized in real chemical systems and how their emergences effects the large-scale dynamics of hierarchically organized chemical networks. Our goal is to use this model to generate design constraints for observing autocatalytic reactions in the experimental system.

We model the formation of large molybdenum nano-structures, known as Mo-Blue wheels (Miras et al 2010). In experimental platforms, these large structures are synthesized using flow through systems fed with Molybdate (MoO_4^-) monomers in a reduced solution (Miras et al 2010). Current analytical techniques cannot elucidate the exact assembly mechanism for large structures such as the Mo_{154} wheel (Miras et al 2010). It is hypothesized that Molybdenum intermediates (Mo_6) are stabilized around a Mo_{36} structure. This mechanism would imply that the Mo_{36} structure would be capable of templating both Mo_6 building blocks, and the larger wheel structure. This introduces the possibility that the Mo_{36} structure is both autocatalytic (since it is formed from the Mo_6 structures it templates) and cross-catalytic in that it templates the larger Mo_{154} structure. Structural studies suggest that the Mo_{36} can be ejected from the larger scale structure enabling it to participate in further templating reactions. These mechanisms may lead to the exponential amplification of the wheel after the emergence of the Mo_{36} template.

The Mo-Blue wheel system presents an interesting chemical system because it contains three clear hierarchical levels of organization: small intermediates (Mo_1 - Mo_6), small templates (Mo_{36}) and larger structures (Mo_{154}). The existence of these clearly delineated levels of organization enables us to explore the consequences of autocatalysis on the levels above and below the proposed autocatalytic entity.

To explore the consequences of this proposed mechanism we developed a model of the Mo-Blue dynamics. Our model is implemented using a kinetic Monte Carlo algorithm (Gillespie, 1977). This technique represents all reactions as either uni-molecular degradation reactions ($A \rightarrow B + C$) or bimolecular ($A + B \rightarrow C$). Structures and intermediates form as the products of bimolecular reactions between building blocks and other intermediates. We assume that all molecules (besides Molybdate monomers, Mo_1) can degrade into component parts. We initialized the system with 10^6 Mo_1 molecules. The model uses Mo_1 , Mo_2 , and Mo_6 , as building blocks, which are given a different degradation rate. Both the Mo_{36} (template) and Mo_{154} (Wheel) form in these simulations. The Mo_{36} templates host intermediate compounds and act to enhance the net rate of bimolecular reactions with those intermediates. This is included in the model by increasing the reaction rate constant of bimolecular reactions when one of the reactants is bound to a template.

To characterize the formation of Mo Blue structures, simulations were run using different sets of bimolecular rate constants. We first modeled the system by inhibiting templates and setting the rate constant to 1.0 for all bimolecular reactions, such that the only differences in propensities were due to the differences in the reduced mass and the relative abundances of molecules. In this setting the only structure which formed was Mo_{36} , albeit in relatively low abundance. We next included the effect of templating for the formation of Mo_6 (templated by the Mo_{36}) and the formation of Mo_{154} (also templated by Mo_{36}). We found that while including the effect of templating did result in dramatic increase in the formation of Mo_{36} (both in rate and in steady state abundance), it did not ensure the formation of Mo_{154} or Mo_{132} . While intermediate compounds between the Mo_{36} and the Mo_{154} or Mo_{132} formed readily, those intermediates always degraded before forming complete structures, resulting in many “frustrated attempts.” We propose several mechanisms which overcome this limitation without *fine-tuning* the model. Including any one of these mechanisms in the model results in

the robust formation of larger structures over a large range of parameters.

The model recovers the dynamical characteristics of the experimental system. In typical simulations the abundance of Mo₁₅₄ remains 0 for a time followed by a period of explosive growth. This feature is also seen in experimental data when the solution is not seeded with Mo₃₆. An example time series of this effect is shown in figure 1.

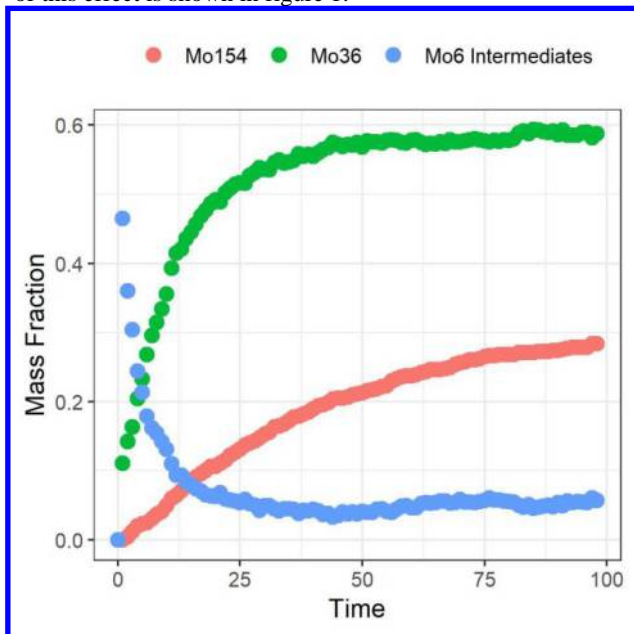


Figure 1: Typical time series from the stochastic simulation. The mass fractions of different molecular species are shown over time. At the beginning of the simulation small intermediates (blue) form rapidly but are transformed into the Mo₃₆ template (green), which then enables the emergence of the larger Mo₁₅₄ wheel (red).

We are currently using this model to investigate alternative mechanisms which could reproduce these dynamical features. We expect to generate new testable hypotheses from the proposed mechanistic pathway which can ultimately be tested in the lab. We are exploring the interactions between the three levels of organization and identifying the most dramatic observable differences between alternative mechanistic pathways.

By interweaving experimental constraints and computational studies we will be able to characterize and explore autocatalysis in a purely inorganic chemical reaction network. Thanks to the hierarchical organization of this particular chemical system we hope to develop new theoretical insights related to how autocatalytic processes drive dynamics within and between different levels.

References

- Ashkenasy, G., Hermans, T. M., Otto, S., & Taylor, A. F. (2017). Systems chemistry. *Chemical Society Reviews*, 46(9), 2543-2554.
- Ashkenasy, G., Jagasia, R., Yadav, M., & Ghadiri, M. R. (2004). Design of a directed molecular network. *Proceedings of the National Academy of Sciences*, 101(30), 10872-10877.
- Kauffman, S. A. (1992). *Origins of order in evolution: self-organization and selection*. In *Understanding Origins* (pp. 153-181). Springer, Dordrecht.
- Miras, H. N., Cooper, G. J., Long, D. L., Bögge, H., Müller, A., Streb, C., and Cronin, L. (2010). Unveiling the transient template in the self-assembly of a molecular oxide nanowheel. *Science*, 327(5961), 72-74.
- Cronin, L., & Walker, S. I. (2016). Beyond prebiotic chemistry. *Science*, 352(6290), 1174-1175.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25), 2340-2361.

Synthetic Biology in the Brain: A Vision of Organic Robots

Ithai Rabinowitch¹

¹The Hebrew University of Jerusalem, Ein Kerem, Jerusalem, 9112102 Israel
ithai.rabinowitch@mail.huji.ac.il

Abstract

Synthetic biology lies on the interface between natural and artificial life. It consists of the assembly of natural biological components into artificially configured biological systems. A main focus of synthetic biology has been the engineering of new gene circuits that can produce artificial cellular functions. I propose to scale up this approach to include, beyond single cells and gene circuits, also entire multi-cellular organisms and the brain circuits that regulate their behavior. Such synthetic biology in the brain will offer new ways for understanding how brain connectivity relates to brain function, and could ultimately lead to futuristic technologies such as neuronally-programmed organic robots or biologically-based brain repair. As a first step towards this ambitious goal I have developed a technique for genetically inserting new synaptic connections into the nervous system of the nematode worm *C. elegans*, enabling the manipulation of information flow in the nervous system and the reprogramming of whole animal behavior in this organism. This approach may be expanded and adapted to other genetic models, and opens the way to possible new forms of artificial life. Such technology, if practiced responsibly, could offer considerable benefits to science, industry and medicine.

Synthetic biology in the brain

Synthetic biology elegantly fuses the fields of engineering and biology, to accomplish the goal of designing and constructing new biological systems out of basic biological parts (Xie and Fussenegger, 2018). The rationale is that the creative process and the practical challenges faced when building a system can substantially contribute to understanding how that system works and to establish causal links between the system's organization and its operation. Synthetic biological systems can be considered as a special form of artificial life, which strongly hinges on natural life. On the one hand, they are composed entirely of organic matter and follow biological principles of operation. On the other hand, they are designed and constructed by human beings intended for human benefit. The organic nature of synthetic biological systems makes them self-reproducible, ecologically compatible with other organisms and the environment, and fully degradable. A unique combination of features that can rarely be found in other forms of artificial life.

Many synthetic biological applications focus on single cells and on the gene networks that control their function (Bashor and Collins, 2018). The potential outcomes are spectacular. For example, synthetic bacteria that could monitor, synthesize and regulate drug administration in a patient's body (Flores Bueso, et al. 2018); or engineered microalgae that could produce

biofuels (Jagadevan, et al. 2018). I propose to expand synthetic biology beyond single cells or populations of single cells, to the realm of multi-cellular organisms. These modified animals exhibiting novel artificial behaviors could substantially enrich the repertoire of synthetic biology, producing more complex and farther-reaching forms of artificial life. Multi-cellular animals, just like single cells, are fundamentally governed by networks of gene interactions. However, the direct coordination and control of their overall behavior is produced by higher order networks of neurons and the synaptic connections that link between them. If it were possible to design and implement specific synaptic connections in the nervous system of an animal, then, in principle, new behaviors could be derived. At some point, an animal harboring an accumulation of such engineered connections, or perhaps new synthesized neural circuits, or, ultimately, an overall redesign of neural connectivity could arguably qualify as a form of artificial life.

Engineering synaptic connections in worms

Caenorhabditis elegans is a 1 mm long nematode worm (Fig. 1a) that dwells in soil and compost, where it feeds on bacteria. Its nervous system consists of only 302 neurons, interlinked by a set of several thousand synaptic connections, which constitute its *connectome*. In fact, the *C. elegans* connectome is the first and, to date, only connectome of any animal to have been mapped. It is intriguing to consider the potential impact of editing the *C. elegans* connectome and forming within it new synthetic patterns of connectivity. Could new behaviors be programmed into the worm in this way? How would such novel behaviors coexist with native ones? To what extent could the entire lifestyle of the worm be reshaped through synthetic design of its neural circuits? Such synthetic biology at the level of the nervous system could help elucidate fundamental principles of brain structure-function relations.

One can think of various hypothetical ways to manipulate, modify and establish new patterns of synaptic connectivity in a live organism. One possible approach is to genetically insert new synthetic synapses into existing neural circuits. Like other metazoans, *C. elegans* uses both chemical and electrical synapses for neural communication. Chemical synapses are complex in structure and are thus challenging to construct artificially. Electrical synapses or gap junctions, in contrast, are considerably simpler. In vertebrates, gap junctions are composed of *connexin* proteins that assemble together to form hemi-channels embedded in the cell membrane. When two compatible hemi-channels contact each other, they fuse to form

a gap junction, a physical channel that enables the passage of charged ions between the connected neurons (Fig. 1b). Invertebrates, like *C. elegans*, do not possess connexin proteins. Instead, they use *innexins* for constructing gap junctions, an independently evolved protein family. Ectopic expression of connexin in *C. elegans* neuron pairs could thus lead to the artificial formation of a gap junction, functionally and specifically linking the two neurons. I have found that connexin driven by cell-specific promoters readily expresses in *C. elegans* neurons in a punctate form, typical to synaptic proteins (Fig. 1c). Moreover, some of these puncta appeared in close apposition (Fig. 1c, inset), suggestive of putative gap junctions. To test the functionality of such presumed engineered electrical synapses, I expressed connexin in two sensory neurons (Fig. 1d). Normally, only one of these neurons responds to a certain stimulus (Fig. 1d, top). However, when, and only when, both neurons ectopically expressed connexin, the responses became equalized, consistent with a coupling of these neurons by a gap junction (Rabinowitch, et al. 2014).

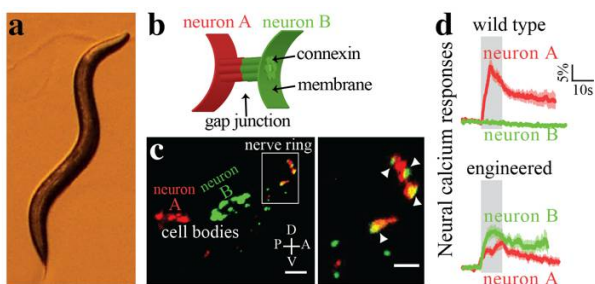


Figure 2: Genetic insertion of new synaptic connections between *C. elegans* neurons. (a) *C. elegans* is a 1mm long nematode worm. (b) Electrical synapses or gap junctions are composed (in vertebrates) of connexin proteins expressed in two adjacent neurons. These form hemi-channels in each neuron and fuse together into a complete gap junction that electrically couples the neurons. (c) Ectopic expression of connexin fused to mCherry or GFP in two *C. elegans* neurons, A (AWC) and B (AIA). The rectangle, enlarged in the inset, marks the nerve ring, the region in which most synaptic contacts in *C. elegans* occur. Putative gap junctions are visualized as proximally localized puncta (▲). Scale bars: 5 μm and 2 μm (d) Calcium responses of two (ASEL and ASER) neurons to a stimulus (salt removal).

Using the techniques described in Rabinowitch, et al. 2014 I also examined the capacity of engineered electrical synapses to reshape simple worm behaviors. For example, in *C. elegans*, the polymodal sensory neuron, ASH, is specialized for detecting noxious stimuli, which normally elicits a withdrawal response (Fig. 2a), resulting from the activation of premotor interneurons, such as AVA (Fig. 2b), and the inhibition of AVB, which otherwise drives forward acceleration. Strikingly, synthetically coupling of ASH to AVB caused worms to move forward, towards noxious stimuli, rather than escaping them by reversing (Fig. 2c), demonstrating a significant functional impact of a specific engineered synaptic connection. A similar principle enabled artificial switching of worm navigation towards a food-related odor, into avoidance of that odor, mimicking an effect that is otherwise attainable through training (Fig. 2d-f). I am currently applying these methods to generate completely novel behaviors in *C. elegans*, and am developing additional techniques for altering its connectome.

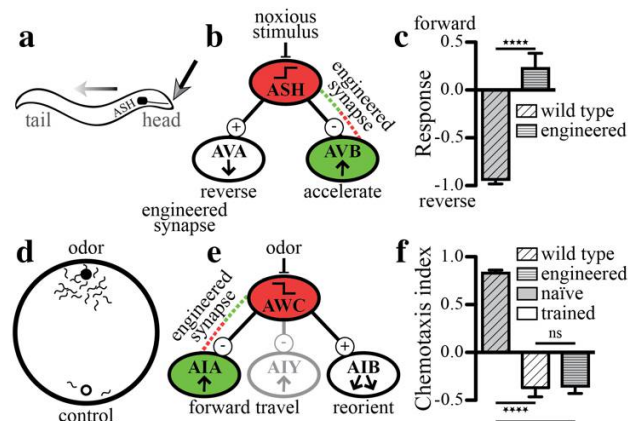


Figure 1: Engineered connections produce behavioral changes. (a) Worms reverse to escape noxious stimuli detected by the ASH sensory neuron. (b) Simplified circuit illustrating ASH connections to premotor interneurons AVA and AVB, which elicit reversing and forward acceleration, respectively. (c) An engineered ASH-AVB connection causes the worm to approach rather than avoid a noxious stimulus (unpublished). (d) Worms use chemotaxis to migrate towards attractive odors. (e) Simplified circuit shows the connectivity between olfactory sensory neuron, AWC, and downstream interneurons important for navigation. (f) An engineered AWC-AIA connection switches behavior from attraction to aversion, mimicking the effects of training (unpublished).

Future prospects

This work illustrates an encouraging step towards a long-term vision of extensive rewiring of the nervous system. Such advances could enhance our understanding of how neural structure determines brain function and could ultimately pave the way to the creation of small organic robots – a new form of artificial life, such as nematode worms programmed to distribute fertilizer among crops and hunt down pests, or to crawl into a patient, perform a medical procedure, and then leave. A long path awaits until such visions may become reality, and considerable ethical, safety and societal considerations will have to be weighed, but the potential gains for science and society are immense. Now is the time to start planning this journey.

References

- Bashor, C. J. and Collins, J. J. (2018). Understanding biological regulation through synthetic biology. *Annual Review of Biophysics*, 47: 399–423.
- Flores Bueso, Y., Lehouritis, P., and Tangney, M. (2018). In situ biomolecule production by bacteria; a synthetic biology approach to medicine. *Journal of Controlled Release*, 275:217–228.
- Jagadevan, S., Banerjee, A., Banerjee, C., Guria, C., Tiwari, R., Baweja, M., and Shukla, P. (2018). Recent developments in synthetic biology and metabolic engineering in microalgae towards biofuel production. *Biotechnology for Biofuels*, 11:185.
- Rabinowitch, I., Chatzigeorgiou, M., Zhao, B., Treinin, M., and Schafer, W. R. (2014). Rewiring neural circuits by the insertion of ectopic electrical synapses in transgenic *C. elegans*. *Nature Communications*, 5:4442.
- Xie, M. and Fussenegger, M. (2018). Designing cell function: assembly of synthetic gene circuits for cell biology applications. *Nature Reviews Molecular Cell Biology*, 19:507–525.

Periodic collective behaviors of organic solvent droplets on the surface of aqueous surfactant solutions

Shinpei Tanaka

¹ Graduate School of Integrated Arts and Sciences, Hiroshima University, 1-7-1 Kagamiyama, Higashi-Hiroshima 739-8521, Japan
shinpei@hiroshima-u.ac.jp

Abstract

Active matter sometimes exhibits life-like complex spatio-temporal patterns. Here we report on complex oscillatory behaviors of droplets floating on an aqueous surfactant solution. Even if the droplets consist of only simple chemicals, the behaviors they exhibit are unexpectedly complicated. They are likely induced by the interaction among droplets, which is mediated through the surface tension field as well as Marangoni flow field created by the droplets themselves.

Introduction

Active matter can be a model of primitive life. Whereas they obey relatively simple laws of physics, which makes the analysis of their behaviors possible, the behaviors they can exhibit are rich and sometimes comparable with biological systems (Toyota et al., 2006; Hanczyc, 2014). For example, droplets of aqueous propylene glycol solution placed on a glass plate exhibit self-propelled motion where inter-droplet interaction mediated by diffusion field creates complex dynamic structures (Cira et al., 2015). Camphor boats floating on a circular channel (Suematsu et al., 2010; Ikura et al., 2013) show peculiar clock-like spatio-temporal patterns. It has even been shown that a droplet can evolve with the help of robotics (Gutierrez et al., 2014).

Recently we found even more complex behaviors in a system of organic-solvent droplets floating on aqueous solution (Tanaka et al., 2015, 2017; Čejková et al., 2019). There, dissolution of organic solvent from droplets decreases the surface tension of aqueous surface, which propels the droplets. When many droplets coexist, the concentration field of dissolved solvent from a droplet overlaps according to the relative position of droplets. Thus the dissolution of surface active substances induces interaction among elements that are providing the substances. Then the elements can show complex spatio-temporal patterns as clusters. In this paper, we summarize our recent findings on complex spatio-temporal structures observed in a chemically simple system.

Methods

Our system consists of droplets floating on a surface of aqueous surfactant solution. The droplets are made of ethyl salicylate (ES, Tokyo Chemical Industry) mixed with paraffin liquid (Sigma-Aldrich). The volume of a droplet was fixed at 10 μ L. They are put on the surface of 35 mM aqueous sodium dodecyl sulfate (Tokyo Chemical Industry) solution in a glass dish of 90 mm in diameter. The droplets contains also dye, Oil Red O (Nacalai Tesque, Tokyo) for visualization. The Oil Red O sometimes affects the motion of droplets, so its concentration was fixed at a low value, 0.005wt%, to minimize the effect.

Results

Figure 1 shows how chains of droplets form ring structures and then collapse (Tanaka et al., 2017). Their motion is periodic, which is explained from a state where a ring of droplets is fully formed [Fig. 1(a)]. First, the ring breaks at a certain position and then starts shrinking [Fig. 1(b) and (c)] to a collapsed state. Then new rings appear in the collapsed state [Fig. 1(c) and (d)] which immediately start growing [Fig. 1(d) and (e)]. Finally, a fully-expanded ring is formed, and the process repeats itself. This wiggling mode of motion continues more than several hours, sometimes days. Whereas two chains coexisted in a case shown in Fig. 1, the number of chains changes with time. Two or more chains sometimes merge into a chain and a chain sometimes breaks into several chains.

We recently showed that nearly same oscillatory motion can be observed also in a decanol system, despite the difference of chemical and physical conditions (Čejková et al., 2019).

Discussion

We found a complex collective mode of motion of droplets, which consist of only a few chemically simple molecules. These droplets are driven by inhomogeneous surface tension field around them, which is created by the dissolution of ES from the droplets (Tanaka et al., 2015). They interact with each other through the surface tension field, as well as

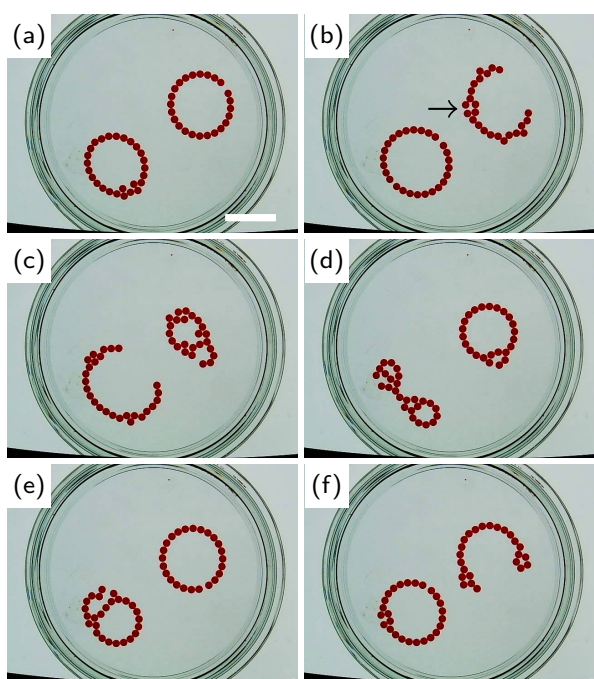


Figure 1: Periodic formation and destruction of rings of droplets. The elapsed time from the first picture was: (a) 0 s, (b) 43 s, (c) 85 s, (d) 127 s, (e) 170 s, and (f) 212 s. The scale bar is 20 mm. The arrow in (b) shows two two-droplet chains align in parallel. A new ring starts growing from this structure.

Marangoni flow induced by the gradient of surface tension. While the details of the interaction among the droplets are not yet fully understood, characteristic features of the interactions are summarized as follows.

First, the droplets do not easily merge together, even if they seem to touch with each other. Precise mechanism of this stabilization is unclear. The stability depends on the condition, such as ES and Oil Red O concentration. For example, if the concentration of Oil Red O is high, they become vulnerable to merge.

Second, there coexists short-range attraction and long-range repulsion, judging from chain structures formed by droplets. Recently it was found that the mixture of short-range attraction and long-range repulsion can induce a chain-like lamellar structure in a confined geometry (Pękalski and Ciach, 2018). A peculiar feature of our dynamic system is, however, that the chains can be formed even in a low number density state as seen in Fig. 1. Moreover, the interaction seems dependent on the arrangement of droplets, so that it changes with time.

Third, if we focus on the properties on the chains, they have a tendency to shrink in the direction parallel to them. So a chain starts shrinking and collapsing as soon as it breaks

at a point. On the other hand, two chains repel each other in the direction perpendicular to them. This repulsion likely creates rings of chains. In fact, a new ring starts its expansion in a collapsed state as soon as two two-droplet chains align in parallel [Fig. 1(d), arrow] serving as its nucleus. In combination, chains behave like electric lines of force, apart from the fact that the chains here can break easily.

Conclusions

The droplets of this system gradually dissolving into solution as well as on the surface of solution. The amount of dissolution is small, but it can create large enough force to propel the droplets with the help of surface tension gradient. It was shown here that even a tiny amount of changes in a simple chemical system can induce its complex behaviors, thanks to the instability of surfaces or interfaces. Given the fact that there are potentially many chemicals that can be used to observe similar spatio-temporal patterns, it is possible to induce more complex behaviors, even those comparable to life's complexity.

Acknowledgments

This work was supported by JSPS KAKENHI (17K05613).

References

- Cira, N. J., Benusiglio, A., and Prakash, M. (2015). Vapour-mediated sensing and motility in two-component droplets. *Nature*, 519:446–450.
- Gutierrez, J. M. P., Hinkley, T., Taylor, J. W., Yanev, K., and Cronin, L. (2014). Evolution of oil droplets in a chemorobotic platform. *Nature*, 5:5571.
- Hanczyc, M. M. (2014). Droplets: Unconventional protocell model with life-like dynamics and room to grow. *Life*, 4:1038–1049.
- Ikura, Y. S., Heisler, E., Awazu, A., Nishimori, H., and Nakata, S. (2013). Collective motion of symmetric camphor papers in an annular water channel. *Phys. Rev. E*, 80:012911.
- Pękalski, J. and Ciach, A. (2018). Orientational ordering of lamellar structures on closed surfaces. *J. Chem. Phys.*, 148:174902.
- Suematsu, N. J., Nakata, S., Awazu, A., and H., N. (2010). Collective behavior of inanimate boats. *Phys. Rev. E*, 81:056210.
- Tanaka, S., Nakata, S., and Kano, T. (2017). Dynamic ordering in a swarm of floating droplets driven by solutal marangoni effect. *J. Phys. Soc. Jpn.*, 86:101004.
- Tanaka, S., Sogabe, Y., and Nakata, S. (2015). Spontaneous change in trajectory patterns of a self-propelled oil droplet at the air-surfactant solution interface. *Phys. Rev. E*, 91:032406.
- Toyota, T., Tsuchi, H., Yamada, K., Takakura, K., Ikegami, T., and Sugawara, T. (2006). Listeria-like motion of oil droplets. *Chem. Lett.*, 35:708–709.
- Čejková, J., Schwarzenberger, K., Eckert, K., and Tanaka, S. (2019). Dancing performance of organic droplets in aqueous surfactant solutions. *Colloids Surf. A*, 556:141–147.

A Candidate Self-Propagating System Enriched by Chemical Ecosystem Selection

Lena Vincent¹, H. James Cleaves^{2,3} and David A. Baum^{1,4}

¹Wisconsin Institute for Discovery, University of Wisconsin-Madison

²Earth-Life Science Institute

³Institute for Advanced Study

⁴Department of Botany, University of Wisconsin-Madison
dbaum@wisc.edu

Abstract

The surface metabolism theory posits that adaptive evolution initiated when autocatalytic chemical systems became spatially localized on mineral surfaces. We searched for such surface-limited metabolisms (SLiMes) using a chemical ecosystem selection paradigm. This involves creating a prebiotic microcosm containing mineral grains and a “soup,” rich in food and potential sources of chemical energy, and then serially transferring a subset of the grains to a new microcosm containing fresh soup and new grains. This repeated dilution should enrich for chemical systems that can self-propagate more rapidly than the rate of serial dilution, and such enrichment should be detectable based on changes in microcosm chemistry over the course of multiple transfers. We deployed chemical ecosystem selection on several different soups and minerals and identified a combination that appears to be conducive to the enrichment of a SLiMe. In these conditions, chemical changes were observed over the first 12-18 transfers, most notably a loss of both orthophosphate and organics (as detected by optical density) from the solution. This loss from the solution correlated with the appearance of fractal structures on the surface of the grains. The putative SLiMes show clear evidence of self-propagation ability and manifest basic ecological dynamics. Ongoing work is evaluating the systems’ evolutionary capacity.

Introduction

The surface metabolism model, first presented 30 years ago by Wächtershäuser, suggests that the first self-propagating systems were autocatalytic sets of simple organic compounds adsorbed onto mineral surfaces (Wächtershäuser, 1988). Once seeded, these surface-limited metabolisms (SLiMes) could use fluxes of food and energy to generate all of their components, resulting in lateral growth as they collectively propagated over the surface (Baum, 2015). Furthermore, because rare chemical reactions can alter or expand an autocatalytic network, SLiMes could be evolvable (Vasas et al., 2012; Baum, 2018). In the context of a plausible prebiotic environment such as the seafloor, the turnover of mineral surface could select for variants that are more stable, more competitive,

and/or better at colonizing newly exposed mineral (Baum, 2019). We used chemical ecosystem selection (Baum & Vetsigian, 2017), a procedure which enriches SLiMes based on their ability to repeatedly colonize new mineral surfaces, to identify a putative SLiMe that emerges repeatedly when incubating synthetic prebiotic soups with pyrite.

Approach

Chemical ecosystem selection involves incubating simulated prebiotic soups with mineral grains and mimicking the active turnover of the mineral surface expected to occur in natural environments. We used a rich chemical soup, reasoning that the more diverse the inputs the higher the likelihood of an autocatalytic systems being present (Kauffman, 1986; Mossel and Steel, 2005; Virgo, et al., 2013). We included minerals to provide a surface, which is needed to spatially segregate cooperating species and might also provide useful catalytic functions. Most of the experiments reported here transferred 10% of either the grains, or the grains and soup every 2-3 days (future experiments will examine the effect of liquid transfer only). The serial dilution protocol means that self-propagating systems will only become enriched over transfers if they are initially rare but can move from grain to grain at a rate greater than 10X each 2-3 days. Furthermore, if multiple systems are present or arise over time (e.g., through addition of new side-reactions), our procedure should enrich for variant SLiMes that propagate faster. To seek evidence of systematic changes in the chemistry of the systems over multiple transfers, we monitored several chemical proxy traits of the solutions following each incubation. In addition to looking for changes over transfers, we also look for heritable differences among lineages (a lineage being a chain of “parent” and “offspring” vials), and compare experimental lineages with control lineages that are generated in parallel with a certain set of experimental vials, but have only

undergone one prior round of transfer. We have deployed the protocol on some dozen different combinations of mineral substrate, organic soup, energy source, and atmospheric gases and have identified at least one set of conditions that consistently provides evidence of SLiMe formation.

Results

We have found that incubating a synthetic, enriched prebiotic soup with natural pyrite grains maintained under low-oxygen conditions produced significant and repeatable evidence of a self-propagating system. Over the first 10+ transfers, chemical proxy traits (free orthophosphate and optical density at UV-Vis wavelengths) show systematic changes suggesting autocatalysis (=self-propagation). During this period of change, lineages differ heritably and we frequently find dramatic and significant differences between experimental and control lineages. This evidence of self-propagation is seen whether we transfer just mineral grains or a grain/soup slurry.

Coincident with the reductions in phosphate and organics from the solution, we observe an increase in the abundance of branched fractal structures on the surface of the pyrite (Fig. 1). These structures are observed never to touch or over-grow one another, suggesting a diffusion-limited growth process. We hypothesize that these structures correspond to SLiMes, which nucleate rarely, but once nucleated are able to transfer readily to other grains. Fractal structures are only formed when organics are present but can form when phosphate is replaced by sulfate. We are conducting additional experiments to further characterize the putative SLiMes and their chemical composition.

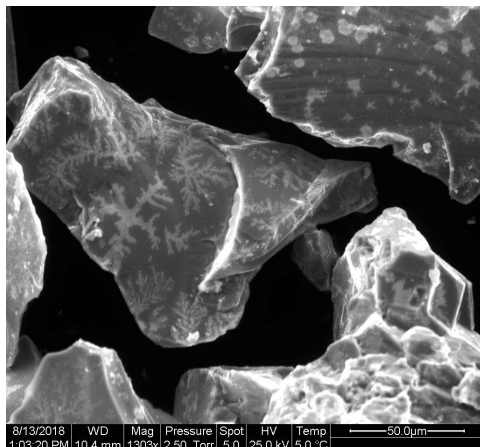


Figure 1: Environmental scanning electron microscopy image of pyrite grains from a lineage exposed to 18 rounds of serial transfer.

We observed that once a proxy trait (e.g., phosphate) reaches a certain critical value, the lineages suddenly return to their initial value. After a few further transfers, lineages begin declining again, resulting in a multigenerational oscillation. We interpret this phenomenon as an ecological boom-and-bust cycle, in which SLiMes reach carrying capacity at which point they deplete their food within the

incubation period, resulting in dissolution of all structures. We are testing this hypothesis using experiments in which we dilute populations to prevent them reaching carrying capacity. This experiment will also allow us to see if the rate of self-propagation increases over generations, which would suggest a capacity for adaptive evolution.

Significance

Our discovery that SLiMes appear to be able to arise spontaneously in plausible prebiotic environments provides circumstantial support for the surface-metabolism model. If we find evidence that the systems are evolvable, our data will support a model in which selection is initiated in SLiMes prior to the formation of compartments. It will be interesting to see how chemically similar these putative SLiMes are to extant life and how alike the systems are that are enriched in different iterations. Moreover, the chemical ecosystem selection framework can be adapted to explore a nearly infinite number of combinations of chemical soups, mineral grains, and environmental factors, which would permit a broad search for conditions conducive to the spontaneous emergence of systems capable of self-propagation and open-ended evolution.

Acknowledgements

Funding for this work comes from NASA (80NSSC17K0296), NSF (CHE-1624562), and the University of Wisconsin Graduate School. We thank all our CESPOoL collaborators and the Wisconsin Institute for Discovery Origins of Life research team, past and present.

References

- Baum, D.A. (2015). Selection and the origin of cells. *Bioscience*, 65(7):678-684.
- Baum, D. A., & Vetsigian, K. (2017). An experimental framework for generating evolvable chemical systems in the laboratory. *Origins of Life and Evolution of Biospheres*, 47(4): 481-497.
- Baum, D.A. (2018). The origin and early evolution of life in chemical composition space. *Journal of Theoretical Biology*, 456, 295-304.
- Kauffman, S.A. (1986). Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119:1-24.
- Mossel, E., & Steel, M. (2005). Random biochemical networks: The probability of self-sustaining autocatalysis. *Journal of Theoretical Biology*, 233(3):327-336.
- Vasas, V., et al. (2012). Evolution before genes. *Biology Direct*, 7(1)
- Virgo, N., and Ikegami, T. (2013). Autocatalysis before enzymes: the emergence of prebiotic chain reactions. In Lió, P., Miglino, O., Nicosia, G., Nolfi, S., and Pavone, N., editors, *Advances in Artificial Life ECAL 2013*, pages 240-247. MIT Press, Cambridge, MA.
- Wächtershäuser, G. (1988). Before enzymes and templates: theory of surface metabolism. *Microbiological Reviews*, 52:452-484.