Noname manuscript No.
(will be inserted by the editor)

Unveiling Evolutionary Algorithm Representation with DU Maps

Eric Medvet · Marco Virgolin · Mauro Castelli · Peter A. N. Bosman · Ivo Gonçalves · Tea Tušar

Received: date / Accepted: date

Abstract Evolutionary Algorithms (EAs) have proven to be effective in tackling problems in many different domains. However, users are often required to spend a significant amount of effort in fine-tuning the EA parameters in order to make the algorithm work. In principle, visualization tools may be of great help in this laborious task, but current visualization tools are either EA-specific, and hence hardly available to all users, or too general to convey detailed information. In this work, we study the Diversity and Usage map (DU map), a compact visualization for analyzing a key component of every EA, the representation of solutions. In a single heat map, the DU map visualizes for entire runs how diverse the genotype is across the population and to which degree each gene in the genotype contributes to the solution. We demonstrate the generality of the DU map concept by applying it to six EAs that use different representations (bit and integer strings, trees, ensembles of trees, and neural networks). We present the results of an online user study about the usability of the DU map which confirm the suitability of the proposed tool and provide important insights on our design choices. By providing a visualization tool that can be easily tailored by specifying the diversity (D) and usage (U) functions, the DU map aims at being a powerful analysis tool for EAs

Peter A. N. Bosman, Marco Virgolin
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
E-mail: peter.bosman@cwi.nl, E-mail: marco.virgolin@cwi.nl

Mauro Castelli
NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, Lisboa, Portugal
E-mail: mcastelli@novaims.unl.pt

Ivo Gonçalves
INESC Coimbra, DEEC, University of Coimbra, Portugal
E-mail: icpg@dei.uc.pt

Eric Medvet
Department of Engineering and Architecture, University of Trieste, Italy
E-mail: emedvet@units.it

Tea Tušar
Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, Slovenia
E-mail: tea.tusar@ijs.si

practitioners, making EAs more transparent and hence lowering the barrier for their use.

Keywords Representation · Diversity · Usage · GE · WHGE · SGE · GSGP · GOMEA · NEAT · Visualization · Heat maps

1 Introduction

Visualization is a powerful tool for supporting reasoning and is often used to gain insight into the workings of Evolutionary Algorithms (EAs). Over the years, several visualization tools for the analysis of EAs have been proposed [8, 12, 18, 22, 23]. They generally provide exploration of EAs properties at two levels, that of single solutions (heavily dependent on solution representation) and entire populations (mostly showing how the fitness improves over generations). Some of these tools [8, 12, 18, 31] specifically support visualization of the genetic heritage between parents and their offspring and entire ancestry of selected solutions. While providing a lot of useful information and allowing certain in-depth analysis of the progress of an EA, existing tools fall short in providing insights about the appropriateness of solution representation, a component of EAs which has always be of prominent importance [47, 53].

A representation-focused analysis of EAs can be performed by exploring how populations evolve at the genotype level. In particular, we are interested in the *diversity* and *usage* of the genotype in the populations, and their interplay. Understanding how diversity changes during evolution can be used to adapt EA parameters so that the desired balance between exploration and exploitation is achieved [54]. On the other hand, analyzing the usage of the genotype can help tailor the representation to the problem at hand by fine tuning some EA-specific parameters (e.g., the genotype size [36]). In addition, inspecting diversity and usage concurrently can reveal further important information. For example, if the entire diversity of the genotype in the population is concentrated at an unused portion of the genotype, this means that there is no actual diversity in the solutions.

Different diversity measures can be defined and tailored according to the particular EA in analysis. For instance, considering Grammatical Evolution, one possible way to measure the diversity of the genotype in the population relies in counting the different values a gene takes for all the individuals in the population (a high number of different values corresponds to high diversity). Similarly, one viable option to compute the usage of the genotype consists in computing how many times each gene is used while building the solution. After the diversity and usage have been calculated, they can be visualized. The Diversity and Usage map (DU map), first presented by Medvet and Tušar [35], serves exactly the purpose of jointly visualizing genotype diversity and usage for each generation, thus showing how they change during evolution. The DU map was conceived for Grammatical Evolution [48], an EA based on an indirect representation affected by a high representation redundancy [33, 60].

This paper substantially extends and improves the original formulation of the DU map. First, we greatly widen the applicability of the DU map, which was designed specifically for variants of Grammatical Evolution, by making its definition and building process more general. In particular, we show how to build DU maps and how to interpret them in terms of usage and diversity according to the particular representation used by six very different EAs. Second, we improve the design of the DU map by considering guidelines from visualization literature [4,39], and by incorporating feedback from an online user study we designed to validate our contribution. To the best of our knowledge, our work is one of the few in the EC area with a methodical evaluation of the visualization tool.

Concerning the generalization and extension of the DU map to other EAs, we considered six EAs which use widely different representations: genotypes are defined as bit and integer strings, trees, ensemble of trees, and neural networks. The selected algorithms are:

- *Grammatical Evolution* (GE) [48], an EA that uses a context-free grammar to map fixed-length genotypes into phenotypes, in three variants: original GE, Structured Grammatical Evolution (SGE) [28] and Weighted Hierarchical Grammatical Evolution (WHGE) [34];
- *Geometric Semantic Genetic Programming* (GSGP) [37], a Genetic Programming (GP) variant that uses geometric semantic operators instead of the traditional GP operators;
- *Gene-pool Optimal Mixing Evolutionary Algorithm* (GOMEA) [68], an EA that performs variation by modelling and exploiting problem structure at the level of dependencies between genes;
- *Neuro-Evolution of Augmenting Topologies* (NEAT) [56], an EA which simultaneously evolves the topology and weights of a Recurrent Neural Network (RNN).

We first define genotype diversity and usage for all the algorithms, showing how to adapt the DU map to these different representations. Then, we construct and extensively analyze many DU maps for those EAs applied to different problems with different parameter settings. Finally, by carefully analyzing the outcomes of the experiments and of the responses to the questionnaire we attempt to answer the following research questions (RQ):

RQ1: Is the DU map useful for unveiling the properties of an EA representation?
RQ2: Is the DU map useful for gaining insight into the behavior of an EA run on a problem, and for comparing problems?
RQ3: Is the DU map useful for choosing EA parameters or components?

We find that, despite the large differences in the considered EAs, the DU map can be used for both drawing quick, high-level, and general conclusions and for extrapolating low-level, EA-specific considerations, in particular those allowing for an immediate understanding of the interplay between genotype diversity and usage. We hence believe that the proposed visualization may serve as a valuable tool for EA users, practitioners, and researchers.

The remainder of this article is organized as follows. Section 2 lists the tasks, defined according to an established taxonomy, that can be performed with the DU maps, motivating the need for this tool. Section 3 briefly surveys the state-of-the-art in the visualization of EAs. Section 4 presents the general definition of the DU map, discuss its design choices, and provides a brief tutorial of its usage. Section 5 describes how the DU map is applied to each of the aforementioned EAs. Section 6 shows the experiments on a number of benchmark optimization problems. Section 7 illustrates the user study which we conducted in the form of a questionnaire and presents the results. Finally, we draw the conclusions in Section 8.

2 Supported visualization tasks

This section specifies which visualization tasks are supported by the DU maps following the visualization task taxonomy from Munzner's book [39]. We consider the three levels of actions that define user goals.

At the highest level, the DU maps were conceived to consume information. More specifically, the DU maps can be used for discovering new knowledge as well as presenting it to others. For example, when researchers visualize their algorithm results through the DU maps, they are able to make new discoveries, such as finding out whether the diversity of the genotype is located at a used portion of the genotype or an unused one, and test their hypotheses, such as confirming that changing a parameter of the algorithm has the expected effect on the diversity of the genotype. After such knowledge has been discovered, the same maps can be exploited for presenting this knowledge to the scientific community.

At the middle level, the DU maps support the following search tasks: locate (target known, location unknown), browse (target unknown, location known) and explore (target and location unknown). An example of a locate search task is finding the generation at which the evolution stagnates, or alternatively, finding the location of the genes that are used and have a high diversity. As an illustration of the browse task, consider the task of determining the usage and diversity of a specific portion of the genotype. Next, the task of finding out whether the genotype has different degrees of usage is an example of the explore task.

At the lowest level, the DU maps can be used to query at all three scopes, i.e., identify, compare, and summarize. For example, users can identify the diversity and usage for each genotype position and generation. They can compare the diversity and usage between different portions of the genotype, different generations and even different DU maps. Finally, the DU map summarizes diversity and usage of the entire genotype across all generations.

While the three research questions posed in the Introduction are concerned mostly the high-level task of discovering knowledge, the experimental-based validation in Section 6 and the user-based validation in Section 7 comprise all three levels of actions.

3 Related work

This section presents a short overview of visualization methods designed to facilitate the understanding of the workings of EAs. More attention is devoted to techniques that are especially related to DU maps, either because they showcase the diversity of individuals within the population [3,15,45,61] or use color to convey multiple pieces of information [19].

Being population-based, EAs can be visualized on two levels—on the level of single individuals (such as the currently best solution) and that of entire populations. Regardless of the perspective, the choice of the visualization method heavily depends on the solution encoding. Some often used representations are simple "zebras" and Gonzo's search space view for binary encodings [11,22,71], parallel coordinates and heat maps (also called matrix charts or density plots) for real-valued genotypes [23], graphs and trees for discrete optimization problems [12, 61], radial trees for genetic programming [13,26], and domain-specific representations for some real-world problems [21,38,50,63]. In multi- and many-objective problems, the focus of the analysis, and consequently visualization, shifts to the objective space and the challenge of visualizing high-dimensional Pareto front approximations [62].

To ease visualization, populations of multidimensional individuals are often projected onto a 2-D space [29,30,51]. Although projections inevitably cause some loss of information, they enable to trace the improvement of individuals during evolution in the form of trajectories [43,44]—a technique predominantly used in visualization of swarm algorithms [20,24]. While trajectories show the evolution of individuals from a global viewpoint, some visualization tools support in-depth analysis of the relations between parent and offspring individuals, and exploration of the ancestry of the chosen (usually best) solution [8,12,18,31]. Other research focuses on visualizing the balance between exploration and exploitation during the optimization [1,27], which is closely related to the diversity of individuals.

In most visualization tools, diversity is not specifically tackled, but can be inferred from concurrently visualizing all individuals in the population (the user is able to see that a population has a high or low diversity by visual inspection). Only a few studies have explicitly defined and visualized diversity [3,15,45,61]. When specified as a measure on the population, diversity can be visualized with the same techniques as the fitness of the best solution, for example, with a simple line graph showing how it changes during the evolution [3]. Diversity can also be computed by measuring the distances between pairs of individuals and then visualized with line graphs and heat maps [45], or more elaborately, using a similarity-preserving mapping to 2-D that positions similar individuals in clusters [15]. Another option is to define diversity based on the occurrence of symbols in the genes, in which case a heat map can be used to illustrate how diversity evolves for each gene separately [61]. This approach is used in our diversity map.

The DU map presented in this paper is a population-level visualization method that uses color to convey both diversity and usage in a single heat map. To the best of our knowledge, the only other approach that employs a similar idea of combining two measures in one color (the so-called bivariate map) is the visualization with pseudo-color [19]. There, individuals are encoded as binary strings and the population is presented in a heat map with one individual per row. The color of each gene is determined depending on its value and the objective and fitness values of the solutions—a distinction is made between the objective and fitness values to accommodate for problems where the fitness value contains some other information in addition to the objective value, such as a penalty determined through expert knowledge of the problem. First, a gene is colored either in blue (value 0) or red (value 1). Then, these two basic colors are modified for the whole row in hue and brightness depending on the solutions' objective and fitness values, respectively. Our approach differs from the one proposed by Ito et al. [19] in two aspects: (1) a single DU map visualizes the entire evolution, not just one population; and (2) color is assigned for each gene separately and is based on its diversity and usage rather than on its value and the fitness and objective of the solution.

4 DU map: overview

In this section, we describe how a DU map is built from a run of an EA. To this end, we first define the class of EAs to which the DU map can be applied. Then, in Section 5.1 and following sections, we show how to apply the DU map to some considerably different EAs: GE in three variants, GSGP, GOMEA, and NEAT. A prototype implementation of the procedures needed to obtain the DU map out of a run of these EAs is publicly available at `https://github.com/ericmedvet/evolved-ge`.

The DU map applies to EAs in which a population of individuals, each one described by a fixed-length genotype of length l, evolves for a number n_{gen} of generations. We denote by P_x the *population* of the xth generation and by $\mathbf{g} = (g_1, \dots, g_l) \in \mathcal{G}_1 \times \dots \times \mathcal{G}_l$ the *genotype* of individual $\mathbf{g} \in P_x$. Each *gene* g_y of $\mathbf{g}$ takes a value from a set $\mathcal{G}_y$, the gene *domain*. The actual domain $\mathcal{G}_1 \times \dots \times \mathcal{G}_l$ of the entire genotype depends on the specific EA to which the DU map is applied (possibly $\mathcal{G}_1 = \dots = \mathcal{G}_l$), whereas the values for n_{gen} and l depend on the specific EA run—they are, usually, part of the EA parameters.

The DU map may also apply to EAs for which the genotype length is not fixed, by considering the longest length observed during the run. In Section 5.4 we show an example of such an approach with NEAT. Note also, that the population size can be variable.

Two functions have to be defined for applying the DU map to an EA: the diversity function and the usage function. The *diversity function* $d_y : \mathbb{N}^{\mathcal{G}_y} \to [0, 1]$, where $\mathbb{N}^{\mathcal{G}_y}$ is the set of all the multisets of cardinality $|P_x|$ built from values in $\mathcal{G}_y$, measures how diverse is the set $G_y = \{g_y^1, \dots, g_y^{|P_x|}\} \in \mathbb{N}^{\mathcal{G}_y}$ of the values of the yth gene in the population P_x: the closer the value of $d_y(G_y)$ to 1, the greater the diversity of G_y. In general, up to l different diversity functions can be used, in order to accommodate the possibly different domains for the l genes. The *usage function* $\mathbf{u} : \mathcal{G}_1 \times \dots \times \mathcal{G}_l \to [0, 1]^l$ measures the degree $u_y(\mathbf{g}) \in [0, 1]$ to which the yth gene g_y contributes to the solution represented by $\mathbf{g}$. The closer the value of $u_y(\mathbf{g})$ to 1, the larger the contribution of the gene g_y to the solution and, from another point of view, the greater the usage of g_y by the EA in building the solution represented by $\mathbf{g}$. The actual form of the functions d_y and $\mathbf{u}$ depends on the specific EA and on the specific aspects of the run which one wants to investigate using the DU map.

The application of the DU map to the results of an EA run consists in building a rectangular heat map of size $n_{\text{gen}} \times l$ out of sequential populations $P_1, \dots, P_{n_{\text{gen}}}$ using the functions $\mathbf{u}$ and d_y (see Figure 1). We remark that only the genotypes of the individuals of each population P_x are needed: no other data (e.g., fitness values, ancestry) is required. Each point (x, y) in the map takes a color $(i_{\text{red}}, i_{\text{green}}, i_{\text{blue}})$ in the RGB space depending on its coordinates x and y, as follows:

$$i_{\text{red}} = d_y(\{g_y; \mathbf{g} \in P_x\}) \tag{1}$$

$$i_{\text{green}} = \frac{1}{|P_x|} \sum_{\mathbf{g} \in P_x} u_y(\mathbf{g}) \tag{2}$$

$$i_{\text{blue}} = 0 \tag{3}$$

where $\mathbf{g}$ is the genotype of the individual in the population P_x of the xth generation, g_y is the yth gene of $\mathbf{g}$, and $u_y(\mathbf{g})$ is the yth component of $\mathbf{u}(\mathbf{g})$.

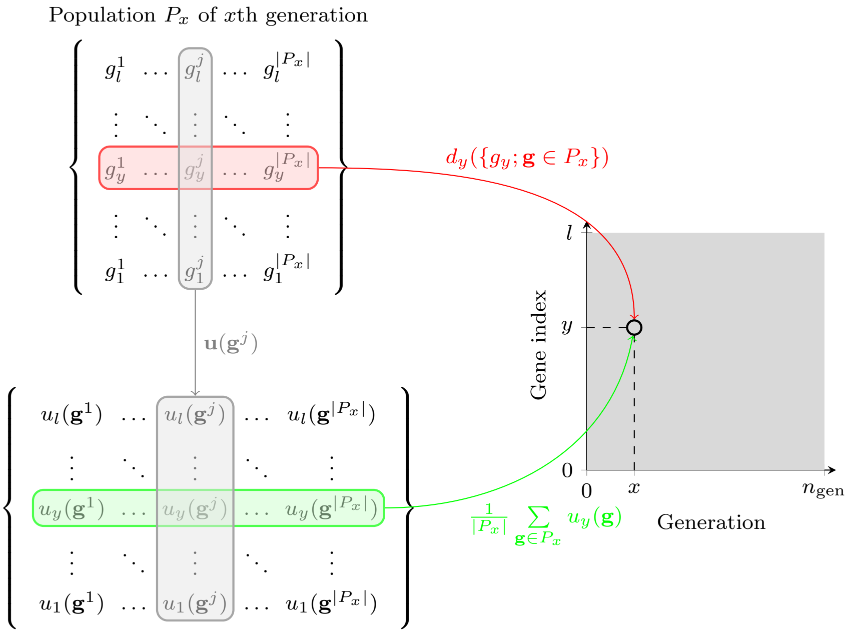
Population P_x of xth generation
$g_l^1 \dots g_l^j \dots g_l^{|P_x|}$
$g_y^1 \dots g_y^j \dots g_y^{|P_x|}$
$g_1^1 \dots g_1^j \dots g_1^{|P_x|}$
$\mathbf{u}(\mathbf{g}^j)$
$u_l(\mathbf{g}^1) \dots u_l(\mathbf{g}^j) \dots u_l(\mathbf{g}^{|P_x|})$
$u_y(\mathbf{g}^1) \dots u_y(\mathbf{g}^j) \dots u_y(\mathbf{g}^{|P_x|})$
$u_1(\mathbf{g}^1) \dots u_1(\mathbf{g}^j) \dots u_1(\mathbf{g}^{|P_x|})$
$d_y(\{g_y; \mathbf{g} \in P_x\})$
$\frac{1}{|P_x|} \sum_{\mathbf{g} \in P_x} u_y(\mathbf{g})$
Gene index
l
y
0
0
x
n_{gen}
Generation

Fig. 1 A schematic representation of how the DU map is obtained: red and green intensities for the point at (x, y) are computed by considering the yth gene of the individuals at the xth generation through the expressions of Eq. 1 and 2, respectively.

In other words, each row of the map (points with the same y coordinate) is related to exactly one gene index in the genotype: the red and green color intensities of the points in the row show how diversity and average usage, respectively, of the yth gene varied during the evolution. Each column in the map (points with the same x coordinate) is related to a generation of the evolution: the red and green color intensities of the points in the column show how the diversity and average usage, respectively, vary along the genotypes of the individuals of the xth generation. Figure 1 summarizes how the color intensities for each point are computed.

All of the DU maps in this work are built in the so-called offline mode—after the algorithms have stopped and produced all the required data. Note however, that since computing diversity and usage depends only on the data of the current population, this could be easily adjusted to enable online analysis of algorithm performance.

4.1 Design choices

In a preliminary phase, we explored some alternatives of the main design choices concerning the DU map. In particular, we focused on the orientation of the map and on the color encoding of the information.

Concerning the orientation, an alternative option is to swap the two axes, i.e., visualize the generation along the y-axis and the gene index along the x-axis. DU maps built according to this alternative design deliver exactly the same information of their swapped counterparts. Yet, we opted for visualizing the generation along

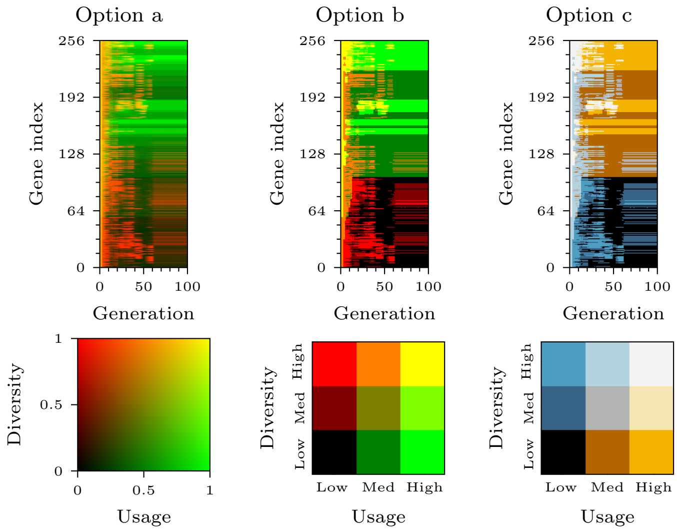

Fig. 2 The three alternative options for the color encoding: example DU map (above) and color legend (below). The three DU maps were obtained from the same run.

the x-axis, because the x-axis usually represents time and the EC practitioners are in general familiar with this way of visualizing how the population changes during the evolution (i.e., *plotting the evolution*). The most prominent example is the simple line plot of the fitness of the best individual at a given generation (on the y-axis) vs. the generation (on the x-axis), a plot which is proposed as a debugging tool by many introductory EC texts, e.g., [14,16,32].

Concerning the color, we experimented with some alternative ways of encoding the diversity and usage values through colors, i.e., alternative ways of setting the values of $(i_\text{red}, i_\text{green}, i_\text{blue})$ for each point. In particular, we considered (a) the design choice presented above, (b) a variant in which the red and green channels are discretized (on three levels each) and (c) a variant using a hue-based, color-blind-safe discretized color scheme [4]. Figure 2 shows an example of a DU map and the corresponding legend for the color encoding for the three options. We verified, by means of an online user study involving 34 users (see Section 7 for a detailed discussion about the questionnaire and the corresponding findings), that the DU map can deliver useful information regardless of the color encoding. The respondents expressed preference for the two discretized options, with the option b being slightly favored over option option c. However, in order to support users with color vision deficiencies, we show the maps using option c in the reminder of the paper.

Finally, we also considered the possibility of adding interactivity to the DU map. A first option would be to interactively display information that is already present in the map, e.g., showing the generation number, the gene index, and the corresponding diversity and usage values as the user moves the pointing device over the map. A second option would be to interactively change the discretization of colors (either going from less colors to more or vice versa). A third option would be to use interactivity for gaining additional information at the granularity level

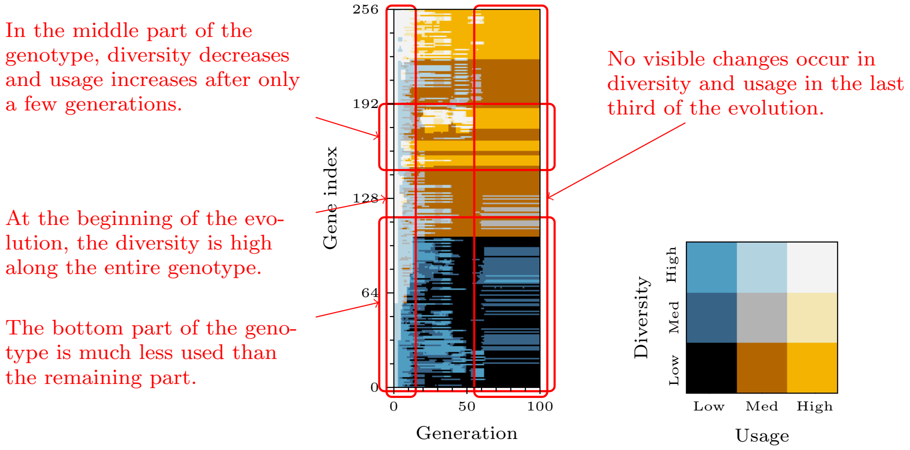

Fig. 3 An example DU map with some possible qualitative interpretations.

of single individuals. For instance, when the user clicks on a map cell in the xth generation, an additional visualization would present the usage across the entire population at that generation. We argue that similar proposals could possibly further extend the usefulness of the DU map, but we did not explore them in this work. On the other hand, the fact that the DU map is not interactive might enable its wider adoption among scholars and practitioners because static images are more easily embedded in scientific documents and technical reports.

4.2 Example of how to read a DU map

We provide an example of how a DU map could be used for gaining insights about a run of an EA. Figure 3 shows an example DU map: four regions of the map are highlighted by means of red boxes and a possible interpretation of each of those regions of the map is shown in the form of a short informal text.

The texts in Figure 3 mention all four information elements contained in a DU map: the gene index, the generation, the diversity, and the usage. In particular, the bottom right text concerns both diversity and usage and suggests that some interplay among them could be inferred. Obviously, deeper insights can be obtained only upon more profound knowledge of the underlying EA.

5 DU maps: application to EAs

5.1 DU map on GE

Grammatical Evolution (GE) [48] is an EA which operates based on an indirect representation. Individuals are represented as bit strings which are then translated, according to a genotype-phenotype mapping function, into strings of a user-provided language defined by means of a context-free grammar (CFG). Because of its mapping function, GE can be applied to a wide class of problems—namely all the problems whose solutions can be described using a CFG—without requiring the user to choose problem specific genetic operators. On the other hand, the same mapping function has been shown to scarcely adhere to the variational inheritance principle [70], which states that offspring should closely resemble, but not be identical to their parents [14]. As a consequence, many improvements of GE have been proposed: in this study we consider the original GE and two recent improvements, Structured Grammatical Evolution (SGE) [28] and Weighted Hierarchical Grammatical Evolution (WHGE) [34]. The DU map is particularly suited to investigate the different mappings of GE variants and the related impact on the diversity of the population. We briefly describe the three variants and how we defined the diversity and usage functions for them.

5.1.1 GE

The genotype-phenotype mapping function of the original GE consumes the genes in the genotype (i.e., bits in the bit string) in groups of 8 (each called codon) in order to choose one of the options in the grammar derivation rule for the leftmost non-terminal in the phenotype. If the end of the genotype has been reached and at least one non-terminal still exists in the phenotype, genes are reused starting from the beginning of the genotype—this operation being called wrapping. A maximum number n_{wrap} of wrappings is allowed: if exceeded, the mapping is aborted and the worst possible fitness is associated with the corresponding individual.

With respect to the DU map, in GE $\mathcal{G}_1 = \cdots = \mathcal{G}_l = \{0,1\}$ (i.e., all genes take values from the binary alphabet), and we define the diversity and usage functions as follows:

$$d_y(G_y) = 1 - 2\left|\frac{1}{2} - \frac{|\{b \in G_y : b = 0\}|}{|G_y|}\right| \tag{4}$$

$$\mathbf{u}(\mathbf{g}) = \frac{1}{\max_{y \in \{1,\dots,l\}} c_y}\mathbf{c} \tag{5}$$

where $c_y \in \mathbb{N}_0^+$ is the number of times the yth bit has been used during the mapping. Concerning the diversity function, it can be observed that $d_y(G_y) = 0$ if and only if all bits in G_y are 0 (or 1) and in $d_y(G_y) = 1$ if and only if exactly half of the bits are 0. The usage function captures the fact that the gene was involved in the genotype-phenotype mapping: in GE, it can be observed that $0 \le c_y \le n_{\text{wrap}}$ and $\forall y, y' : |c_y - c_{y'}| \le 1$.

5.1.2 WHGE

WHGE differs from GE only in the mapping function which consists, in the former, in a recursive function which takes a non-terminal symbol and a bit string and returns a derivation tree. The function is firstly invoked with the genotype and the grammar starting symbol: internally, it uses the input bit string for choosing a derivation rule and then divides the bit string in a number of chunks equal to the number of symbols in the chosen derivation; finally, it recursively calls itself for each chunk and corresponding symbol—for more details, we refer the readers to the study of Medvet [34].

With respect to the DU map, the gene domains and the diversity and usage functions for WHGE are the same of those of GE. However, differently than in GE, conditions on the number c_y of times the yth bit has been used do not hold, i.e., the upper bound is not n_{wrap} and there can be a large difference between values of c_y for different gene indexes.

5.1.3 SGE

In SGE, the genotype consists of an integer string, instead of a bit string. Genes (i.e., integers) are grouped together. Each group corresponds to a non-terminal symbol and each gene in the group corresponds to a possible derivation option of that non-terminal. Genes in the same group have the same domain, which is related to the number of possible derivation options for the corresponding non-terminal. SGE lacks a mechanism for reusing the genotype: instead, the ability of coping with infinite languages is obtained by working with a non-recursive grammar derived automatically from the input grammar using a user-provided parameter $d_{\max}$ representing the maximum level of recursion of derivation rules. Differently than GE and WHGE, SGE works with ad hoc genetic operators which are built considering the structure of the genotype—for more details, we refer the reader to the work of Lourenço et al. [28].

With respect to the DU map, a number n_{NT} of different gene domains exist, each one being a subset $\{0, \dots, n_h^{\text{opt}} - 1\}$ of $\mathbb{N}$, where n_{NT} is the number of non-terminal symbols in the grammar and n_h^{opt} is the number of derivation options for the hth non-terminal symbol. The diversity function is defined as:

$$d_y(G_y) = 1 - \text{NV}(\mathbf{f}_{G_y, \mathcal{G}_y}) \tag{6}$$

where it measures the diversity of a multiset G_y as one minus the normalized variance $\text{NV}(\mathbf{f}_{G_y, \mathcal{G}_y})$ of the relative frequencies $\mathbf{f}_{G_y, \mathcal{G}_y} = (f_1, \dots, f_{|\mathcal{G}_y|})$ of the $|\mathcal{G}_y|$ possible elements in G_y. The normalized variance $\text{NV} : [0,1]^m \to [0,1]$ is defined as:

$$\text{NV}(\mathbf{f}) = \frac{m \frac{\sum_i f_i^2}{\sum_i f_i} - 1}{m - 1} \tag{7}$$

In other words, $d_y(G_y) = 0$ if G_y contains only repetitions of the same element among many possible elements (i.e., no diversity) and $d_y(G_y) = 1$ if all possible elements appear in G_y for the same number of times (i.e., maximal diversity). The usage function for SGE is the same of GE and WHGE, but in SGE, $c_y \in \{0, 1\}$, since a gene is never reused.

5.2 DU map on GSGP

Geometric Semantic Genetic Programming (GSGP) was introduced recently by Moraglio et al. [37]. It is one of the numerous GP techniques that try to exploit the concept of semantics [67]. Even though the term semantics can have several interpretations, it is a common trend in the GP community to define the semantics of a solution T as the vector $\mathbf{s}(T) = (T(x_1), \dots, T(x_n))$ of its output values on the training data. According to this definition, it is possible to identify a GP individual as a point $\mathbf{s}(T)$ in a multidimensional space called the *semantic space* (where the number of dimensions is equal to the cardinality of the training set).

In GSGP, traditional crossover and mutation are replaced by so-called Geometric Semantic Operators (GSOs), which exploit semantic awareness and induce precise geometric properties on the semantic space. In particular, GSOs induce a unimodal error surface on any problem consisting of matching sets of input data into known targets (like supervised learning problems such as symbolic regression and classification) [37]. Here, we report the definition of the GSOs as given by Moraglio et al. for real functions domains (i.e., $T : \mathbb{R}^m \to \mathbb{R}$), since these are the operators we will use in this study. For applications that consider other types of data, we refer the reader to the work of Moraglio et al. [37].

Geometric Semantic Crossover (GSXO) generates, as the unique offspring of parents T_1, T_2, the individual T_{XO}:

$$T_{\text{XO}} = T_1 T_R + T_2(1 - T_R) \tag{8}$$

where T_R is a random real function whose output values range in the interval $[0, 1]$.

Geometric Semantic Mutation (GSM) returns, as the result of the mutation of a individual T, the individual T_{mut}:

$$T_{\text{mut}} = T + \lambda(T_{R_1} - T_{R_2}) \tag{9}$$

where T_{R_1} and T_{R_2} are random real functions whose output values range in the interval $[0, 1]$ and λ is a parameter called mutation step.

As shown in the work of Moraglio et al. [37], GSXO corresponds to geometric crossover in the semantic space (i.e., the point representing the offspring lies on the segment joining the points representing the parents) and GSM corresponds to box mutation on the semantic space (i.e., the point representing the offspring lies within a box of radius λ, centered in the point representing the parent).

Despite the property of inducing a unimodal fitness landscape, GSOs present an important drawback that is related to the fast growth of the size of the individuals in the population. This makes the fitness evaluation unbearably slow (thus making the system unusable), but the issue was successfully addressed by Castelli et al. [9] with an implementation of Moraglio's operators that makes them not only usable in practice, but also very efficient. With this implementation, the size of the evolved individuals is still very large, but they are represented in a particular way that makes their evaluation faster than standard syntax-based GP. For the details, we refer the reader to the work of Castelli et al. [9, 66], but the main idea is presented here. Upon the evaluation of the individuals created after the initialization, their semantics is stored in a data structure. In the following generations, the newly created individuals are built by plugging the individuals of the initial generation in the structure defined by the GSOs. Hence, a new individual consists of the parent individual(s) and one or more random trees. That is, we do not need to effectively build the new individuals by swapping the subtrees of the parents, but we can compute the offspring by only using the information related to the semantics of the parent(s) and the semantics of the random tree(s). Hence, individual created in the subsequent generations are created and evaluated very efficiently, and their evaluation on a particular training case is performed in constant time.

GSGP is an interesting case study for DU maps, because of the geometric properties of GSOs. More specifically, with respect to standard syntax-based GP crossover, GSXO was shown to be quite ineffective on a large set of applications. The fact that GSXO generates an offspring whose semantics stands in the segment joining the semantics of the parents is an important property to ensure that the fitness landscape is unimodal but, on the other hand, it represents a limitation when population diversity must be guaranteed. In particular, if we imagine a GP population as a cloud of points in the semantic space, GSXO is only able to generate points that are "inside" the cloud. The consequences are twofold: (1) if the target (that is also a known point in the semantic space) is not contained inside the cloud, GSXO will never be able to generate it; (2) the individuals created by crossover present a low degree of diversity. Hence, in GSGP, a greater (with respect to standard GP) mutation rate is commonly employed because mutation is the only operator able to explore points of the semantic space that are "outside" the cloud. However, the fact that GSGP uses mutation to explore the search space has a negative impact on the convergence speed of GP: typically a GP with GSOs requires a greater number of generations to converge towards a good quality solutions when compared to standard GP. Using a DU map we expect to capture features that characterize GSOs.

With respect to the application of the DU map to GSGP, two key observations can be made: (1) individuals are represented as trees, whose size is, in principle, unbounded; however (2) each individual may also viewed as the result of the recombination of many other individuals through the application of GSOs—namely, the individuals of the initial population. We hence consider the yth gene of a GSGP genotype as the number of times the yth individual of the initial population is used: this way, the genotype has a length l equals to the population size and the gene domains are $\mathcal{G}_1 = \cdots = \mathcal{G}_l = \mathbb{N}_0^+$. We define the diversity and usage functions as follows:

$$d_y(G_y) = \text{NV}\left(\left\{\frac{g}{\max_{g' \in G_y} g'}, g \in G_y\right\}\right) \tag{10}$$

$$\mathbf{u}\left(\mathbf{g}\right) = \frac{1}{\max_{y \in \{1,\dots,l\}} g_y}\mathbf{g} \tag{11}$$

where the normalized variance defined in Eq. 7 is applied to the multiset G_y of gene values after having rescaled the values to $[0,1]$.

In other words, initial GP individuals are considered as building blocks for the construction of individuals in the subsequent generations. If the number of times the yth building block has been used widely varies across the individuals of a generation, then the corresponding diversity d_y will be large, and the opposite. Concerning usage, if in an individual T, the yth building block is used many times with respect to other building blocks, then the corresponding usage u_y will be large, and the opposite.

5.3 DU map on GOMEA

The Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) [58] is a framework which has been recently applied to GP [68]. GOMEA has been shown to outperform standard GP and compete with state-of-the-art methods on deceptive synthetic problems and benchmark problems of binary circuit generation. Thanks to its capability to learn and exploit problem structure, and together with a new

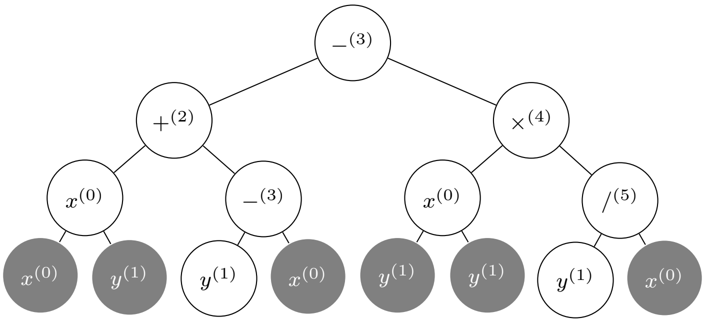

Fig. 4 Example of full binary tree for symbolic regression. The gray nodes are inactive. By parsing the tree with level-order traversal, the equivalent string of symbols is obtained: $(-, +, \times, x, -, x, /, x, y, y, x, y, y, y, x)$.

method to identify and re-use important building blocks, GOMEA reached excellent scalability on the Even parity problem [68].

Individuals evolved in GOMEA are represented as trees of a predefined shape and, hence, with a fixed size. This allows typical linkage learning techniques (i.e., techniques to measure interdependencies in the genotype) to be applied, resulting in valuable information that can be exploited during variation [59,68]. At the same time, a fixed-size genotype makes it straightforward to compute diversity and usage.

In detail, the GOMEA representation is as follows. Let $\mathcal{T}$ be the set of terminal nodes (e.g., input variables or constants), and let $\mathcal{F}$ be the set of function nodes. Let r be the maximum *arity*, i.e., the maximum number of arguments required by any function in $\mathcal{F}$. Given a user-defined height h, the genotype is a full r-ary tree of height h, where each node with a lower depth than h has exactly r children. The fixed-length l of the genotype corresponds to the number of nodes in the tree, i.e., $l = \sum_{i=0}^{i=h} r^i$. Nodes can be *inactive*: inactive nodes are not parsed during evaluation. All the children of a terminal or inactive node are inactive; the rightmost $r - r'$ children of a function node with r' arguments are inactive. Figure 5.3 shows an example of the representation used by GOMEA.

A key component of GOMEA is the *Family of Subsets* (FOS), a model which is learned at each generation and used by the variation operator GOM to generate the offspring. A FOS $F = \{F_1, \dots, F_k\}$, with $F_i \subset \{1, \dots, l\}$, is a set of crossover masks, where each mask is a set of indices that essentially represent positions in the genotype—note that F_i is strictly a subset of $\{1, \dots, l\}$ because swapping the entire genotype typically leads to premature convergence. In this work, four types of FOS are considered: the Linkage Tree (LT), the Random Tree (RT), and two of their variants which use a simple diversity preservation mechanism (LTd and RTd). Aiming to model the structure of the problem, the LT represents hierarchical interdependency between gene positions, and is learned by measuring the mutual information between all pairs of nodes, and by performing hierarchical clustering based on these measurements [68]. The RT is built in a similar way to the LT, but randomly. This FOS may be preferred to LT for problems where the learning structure does not necessarily improve the search, but using (random) hierarchical crossover masks may still be useful. Finally, LTd and RTd are variants of LT and RT with a basic diversity preservation mechanism: $1 \notin F_i$, i.e., the root node is not present in any crossover mask, thus will not be changed during variation. In the rest of the paper, the notation $\mathrm{GOMEA_F}$ refers to GOMEA using a specific FOS F (e.g., $\mathrm{GOMEA_{LT}}$ for LT).

In GOMEA, variation and selection are both handled by the Gene-pool Optimal Mixing (GOM) operator, which is applied to every individual in the population. GOM uses the crossover masks contained in the FOS to generate one offspring that is guaranteed to be at least as fit as the parent. Because of this, a separate selection phase is not needed. The GOM operator works as follows. To begin, a backup $\mathbf{g}'_p$ and the offspring $\mathbf{g}'_c$ of the parent $\mathbf{g}_p$ are created. After shuffling the FOS (operation which is done to diversify the effect of mixing when using hierarchic crossover masks), for each set F_i a crossover operation is applied whereby the nodes of $\mathbf{g}_c$ are replaced by the ones of a random donor $\mathbf{g} \in P$ in the same positions, namely those in F_i. If this results in a syntactical change, then $\mathbf{g}_c$ is evaluated and the new fitness is compared with the previous one: if $\mathbf{g}_c$ becomes less fit than $\mathbf{g}'_p$, then the change is reverted, otherwise the change is accepted and $\mathbf{g}'_p$ is updated.

With respect to the DU map, the GOMEA genotype is viewed as a fixed-length string of symbols where the first symbol corresponds to the root of the tree, the following r symbols correspond to the root children, the following r^2 symbols to the root grandchildren, and so on (*level-order parsing*). As mentioned before, the length of the resulting genotype $\mathbf{g}$ is $l = \sum_{i=0}^{i=h} r^i$. The gene domains $\mathcal{G}_y$ depend on the position, i.e., on whether the corresponding nodes in the tree can be function nodes, terminal nodes, or both; moreover, they allow to describe active and inactive nodes. Since trees are initialized with a minimum height of 2 (specifically, with the Half-and-Half method [25]), the domains are defined as follows:

$$\mathcal{G}_y = \begin{cases} \mathcal{F} \times \{1,0\} & \text{for } y \leq 1 + r \text{ (root and its children)} \\ \mathcal{T} \times \{1,0\} & \text{for } y > l - r^h \text{ (leaves at maximum depth)} \\ \mathcal{F} \cup \mathcal{T} \times \{1,0\} & \text{otherwise (other tree nodes)} \end{cases} \tag{12}$$

In other words, each gene is a pair $g_y = (s_y, a_y)$ where the first element s_y corresponds to the tree node and the second element a_y is 0 for inactive nodes and 1 otherwise. The diversity function is defined as in Eq. 6, while we define the usage function as follows:

$$\mathbf{u}(\mathbf{g}) = (a_1, \ldots, a_l) \tag{13}$$

Because GOMEA encodes a tree with level-order parsing, the bottom of the DU map (small gene indices) represents the evolution of diversity and usage for the top of the trees. More specifically, the bottom row of the map (i.e., G_1) corresponds to the root, and the top half of the map (i.e., the top r^h rows) corresponds to the leaves at maximum depth.

5.4 DU map on NEAT

Neuro-Evolution of Augmenting Topologies (NEAT) [56] is an EA which simultaneously evolves the topology and the weights of a Recurrent Neural Network (RNN). NEAT has been widely used in many applications, e.g., in evolutionary robotics [17] to generate RNN-based controllers able to address complex tasks. Moreover, NEAT constitutes the foundations on which other more recent and sophisticated approaches for neuro-evolution are built, e.g., HyperNEAT [55] and odNEAT [52].

Stanley and Miikkulainen [56], inventors of NEAT, showed that the simultaneous evolution of the RNN topology and weights enabled a faster convergence to better solutions with respect to fixed-topology-based methods. In essence, the improvement was motivated by the three main components of NEAT: (1) a principled method of crossover of different topologies, (2) the protection of structural innovations through speciation, and (3) incremental growing from minimal structure.

The key idea behind NEAT, which essentially enabled the first two components, is in the individual representation. Each individual is represented by a variable-length genotype containing two kinds of genes: node genes and connection genes. *Node genes* are related to nodes of the RNN (input, output, and hidden nodes): each gene consists of an innovation number (see below). *Connection genes* are related to edges between nodes of the RNN: each gene consists of the in-node innovation number, the out-node innovation number, the weight, a binary value specifying if the edge is enabled or not (the enable bit), and an innovation number.

Innovation numbers are unique (across the entire run) positive integer identifiers and are the salient feature of the NEAT representation. NEAT genetic operators never change the value of an innovation number, nor in-node and out-node innovation numbers in connection genes—as a consequence, the gene with a given innovation number is the same in all the individuals of all the generations. Moreover, whenever a new hidden node or a new edge is inserted in an individual, a new innovation number is assigned to the corresponding gene whose value is set using an evolution-wise global counter. For further details, we refer the reader to the work of Stanley and Miikkulainen [56].

NEAT is of particular interest in this study because of two reasons. First, we show that the DU map can be applied also to EAs where the genotype length is not fixed and known a priori before the run (differently than in GE and variants, GSGP, and GOMEA). Second, it radically differs from the other considered EAs both in the nature of the evolved artifacts and in the kinds of problems it is most suited for.

With respect to the application of the DU map to NEAT, two key observations can be made: (1) the innovation number $i \in \mathbb{N}^+$, due to its uniqueness and semantics, may be used as an index of the position of a gene in the genotype; (2) if an hidden node of an individual is connected only to edges for which the enable bit is not set, then it is irrelevant to the RNN. We hence consider the yth gene domain as $\mathcal{G}_y = \{1, 0\}$, if genes associated with y are node genes, or as $\mathcal{G}_y = [-1, 1] \times \{1, 0\}$, otherwise—in the former case, a gene is 0 if the corresponding hidden node is connected only to disabled edges; in the latter, the gene consists of the weight and the enable bit. The length l of the genotype is, for what concerns the application of the DU map, equal to the value of the global counter at the end of the run. We define the diversity and usage functions as follows. Let denote by $g_y = (w_y, e_y)$ the

Table 1 Summary of the EA representations for the considered EAs.

EA	Description	Genotype domain $\mathcal{G}_y$
GE	Bit string	$\{0,1\}$
WHGE	Bit string	$\{0,1\}$
SGE	Integer string	$\{0,\dots,n_h^{\mathrm{opt}}-1\}$
GSGP	Weights of initial trees	$\mathbb{N}_0^+$
GOMEA	Predefined shape tree	$\begin{cases} \mathcal{F}\times\{1,0\} & \text{for } y \leq 1+r \\ \mathcal{T}\times\{1,0\} & \text{for } y > l-r^h \\ \mathcal{F}\cup\mathcal{T}\times\{1,0\} & \text{otherwise} \end{cases}$
NEAT	Neural network	$\begin{cases} [-1,1]\times\{1,0\} & \text{for connections} \\ \{1,0\} & \text{for nodes} \end{cases}$

Table 2 Summary of the diversity and usage functions for the considered EAs.

EA	Diversity $d_y(G_y)$	Usage $\mathbf{u}(\mathbf{g})$
GE	$1-2\left\lvert\frac{1}{2}-\frac{\lvert\{b\in G_y : b=0\}\rvert}{\lvert G_y\rvert}\right\rvert$	$\frac{1}{\max_{y\in\{1,\dots,l\}} c_y}\mathbf{c}$
WHGE	$1-2\left\lvert\frac{1}{2}-\frac{\lvert\{b\in G_y : b=0\}\rvert}{\lvert G_y\rvert}\right\rvert$	$\frac{1}{\max_{y\in\{1,\dots,l\}} c_y}\mathbf{c}$
SGE	$1-\mathrm{NV}(\mathbf{f}_{G_y,\mathcal{G}_y})$	$\frac{1}{\max_{y\in\{1,\dots,l\}} c_y}\mathbf{c}$
GSGP	$\mathrm{NV}\left(\left\{\frac{g}{\max_{g'\in G_y} g'}, g\in G_y\right\}\right)$	$\frac{1}{\max_{y\in\{1,\dots,l\}} g_y}\mathbf{g}$
GOMEA	$1-\mathrm{NV}(\mathbf{f}_{G_y,\mathcal{G}_y})$	$(a_1,\dots,a_l)$
NEAT	$\begin{cases} \mathrm{NV}\left(\left\{\frac{1+w}{2},(w,e)\in G_y\right\}\right) & \text{for connections} \\ \mathrm{NV}(G_y) & \text{for nodes} \end{cases}$	$\begin{cases} (e_1,\dots,e_l) & \text{for connections} \\ (g_1,\dots,g_l) & \text{for nodes} \end{cases}$

connection genes; then:

$$d_y(G_y) = \begin{cases} \mathrm{NV}\left(\left\{\frac{1+w}{2},(w,e)\in G_y\right\}\right) & \text{for connection genes} \\ \mathrm{NV}(G_y) & \text{for node genes} \end{cases} \tag{14}$$

$$\mathbf{u}(\mathbf{g}) = (u_1,\dots,u_l) \quad \text{where } u_y = \begin{cases} e_y & \text{for connection genes} \\ g_y & \text{for node genes} \end{cases} \tag{15}$$

In other words, the diversity is determined by the normalized variance of weights, for connection genes, or of enable bits, for node genes, while the usage of a gene is determined by its enable bit.

Finally, Tables 1 and 2 summarize the representations, and the diversity and usage functions for the six considered EAs.

6 Experimental-based validation

In this section, we describe the experimental evaluation we performed in order to validate the usefulness of the DU map. In particular, we aimed at answering the following research questions (RQs):

RQ1: Is the DU map useful for unveiling the properties of an EA representation?

RQ2: Is the DU map useful for gaining insight into the behavior of an EA run on a problem, and for comparing problems?
RQ3: Is the DU map useful for choosing EA parameters or components?

To this end, we performed several runs of the six EAs (GE, WHGE, SGE, GSGP, GOMEA, and NEAT) on various problems and with different parameter settings.

Overall, we considered several benchmark problems, including synthetic and real-life symbolic regression problems, Boolean problems, and alike, for the five GP varinats (GE, WHGE, SGE, GSGP, and GOMEA), and a single evolutionary robotics problem for NEAT. We here briefly describe the problems—we refer the reader to cited papers for more details:

- *7T4* [49]: a deceptive synthetic problem of seven concatenated trap functions of size four;
- *Airfoil* [6]: real-life regression of airfoil self-noise data with five observations and 1502 features;
- *Car* [57]: evolution of a NN-based controller for a driverless car (with 15 inputs and two outputs) aimed at maximizing traffic efficiency and safety;
- *Concrete* [10]: real-life regression of concrete compressing strength data with eight observations and 1029 features;
- *EParity-6* [25]: synthesis of the Even parity 6 Boolean function;
- *KLandscapes-4* [65]: a synthetic GP benchmark with tunable hardness (we set $k = 4$);
- *MOPM-2* [69]: synthesis of the multiple outputs parallel 2-bit multiplier Boolean function;
- *Nguyen7* [64]: symbolic regression of $f(x) = \log(x+1) + \log(x^2+1)$, with a training set of 20 points in $[0, 2]$;
- *Pagie1* [42]: symbolic regression of $f(x, y) = \frac{1}{1+x^{-4}} + \frac{1}{1+y^{-4}}$, with a training set of 125 points evenly spaced in $[-5, 5] \times [-5, 5]$;
- *Slump* [72]: real-life regression of concrete slump test data with eight observations and 1029 features.
- *Text* [33]: evolution of the target string `Hello world!` using a predefined grammar with the fitness given by the edit distance to the target string;
- *Yacht* [40]: real-life regression of yacht hydrodynamics data with eight observations and 1029 features.

The algorithm parameters that are most relevant for the following discussion, i.e., the genotype length l, the number of generations n_{gen}, and the size of the population n_{pop}, are shown for each EA in Table 3. Recall that l and n_{gen} determine the size of the DU map and its aspect ratio. In SGE, l is not set directly but is determined by the grammar and the parameter $d_{\max}$, which we set to 6; similarly, l in GOMEA is determined by the tree height h, which we set to 6. In GSGP, l is by design equal to n_{pop}. In NEAT, l depends on the specific run (see Section 5.4).

6.1 RQ1: Is the DU map useful for unveiling the properties of an EA representation?

This section shows the DU maps obtained by five different algorithms on the same synthetic symbolic regression problem (Nguyen7). Considering the different nature

Table 3 Most relevant parameters of the considered EAs (see text for more information).

EA	l	n_{gen}	n_{pop}
GE	256	100	500
WHGE	256	100	500
SGE	92	100	500
GSGP	100	50	100
GOMEA	127	100	1024
NEAT	142	300	100

of NEAT with respect to all the other algorithms, a different benchmark was used for this EA, namely the Car problem.

Figures 5 and 6 show Diversity maps, Usage maps and DU maps for the aforementioned algorithms. Diversity and Usage maps essentially correspond to visualizing only the corresponding color channel: we show them here to better explain how the DU map works.

6.1.1 Non-EA-specific observations

It must be stressed that the considered EAs are based on very different individual representations. Moreover, the considered EAs differ also in other components (e.g., NEAT incorporates a innovation preservation mechanism, GSGP employs genetic operators with geometric properties). Nevertheless, the DU maps shown in Figures 5 and 6 share some traits which allow to do some general, high-level observations. This is because the diversity and usage functions can be adapted to the specific EA to maintain consistency in the semantics of the DU map.

The foremost observation is related to the interplay between diversity and usage: in general, the DU maps highlight the fact that the portions of the genotype in which the population exhibits significant diversity are those for which the usage is low (i.e., blue[1] regions are larger than white regions)—the phenomenon being particularly evident for GE, WHGE, SGE, and GOMEA. It is worth noting that it would be hard to spot the interaction between diversity and usage by looking only at the values of a diversity measure during the evolution.

Another interesting observation concerns the relative "amount of used genotype", which roughly corresponds to, in a given generation (i.e., column of the DU map), the average intensity of yellow. Leaving aside any consideration about possible premature convergence to local minima, it can be seen that different EAs actually use different amounts of the genotype. For instance, GE uses the full genotype, whereas SGE and GSGP only use a small fraction. From another point of view, this finding might be an indication that (parameters of) the representation need to be fine-tuned (e.g., increase or reduce the number of bits in GE [35] or the depth in SGE) in order to allow for a more effective or more efficient search.

The final observation we make, is that the DU maps visualize the rate of change in the evolution dynamic, giving an indication of when the dynamic is reduced to a negligible amount. Essentially, that moment can be inferred by seeing when the

[1] For readability, we use the terms "black", "yellow", "white", and "blue" for the four colors at the corners of the color legend (see Figure 2) corresponding to, respectively, low diversity and low usage, low diversity and high usage, high diversity and high usage, and high diversity and low usage.

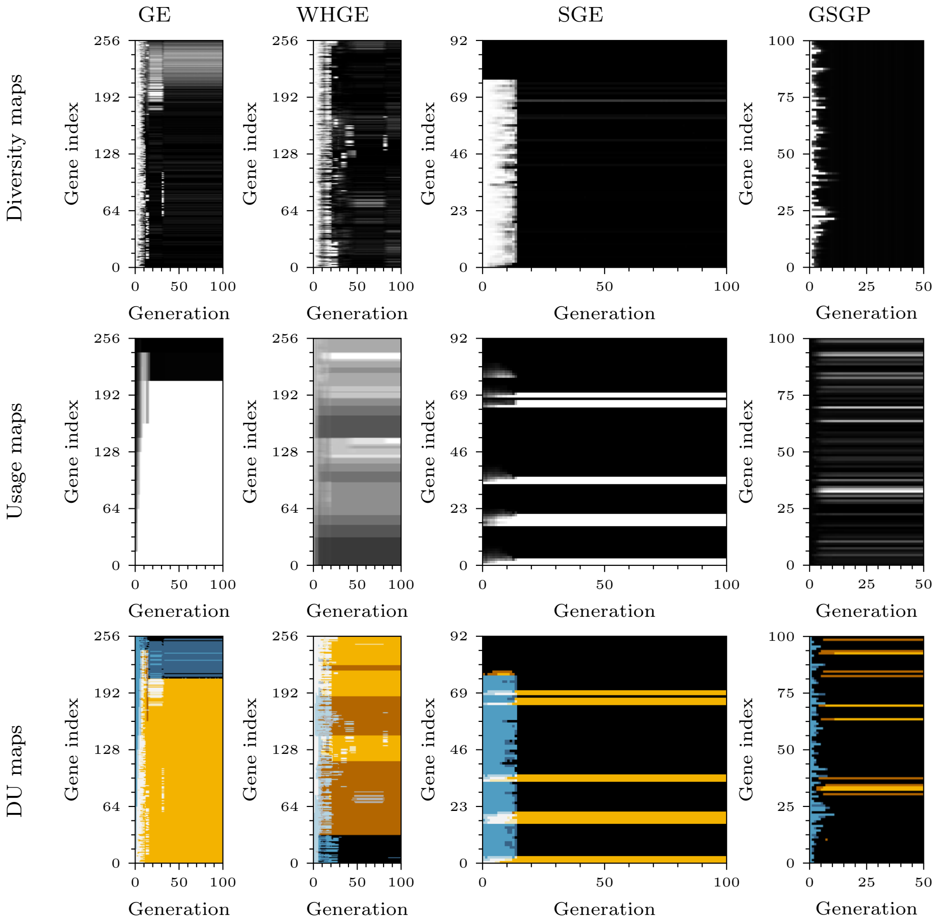

Fig. 5 Diversity maps, Usage maps, and DU maps obtained from one run of GE, WHGE and SGE on the Nguyen7 problem.

DU map columns stop varying in colors. For instance, Figures 5 and 6 show that for SGE, GSGP, and GOMEA the evolution dynamic becomes negligible after few generations, whereas in GE and WHGE it lasts longer—roughly to the first third of the evolution. Differently, NEAT (whose DU map is obtained on a very different problem, see Figure 6) apparently never stops evolving.

6.1.2 EA-specific observations

Concerning GE, WHGE, and SGE, Figure 5 shows that the way in which usage varies along the genotype is consistent with the respective genotype-phenotype mapping functions. In GE, the genotype is used starting from the beginning up to a given position (note that in the specific DU map of Figure 5 no wrapping can be seen). In WHGE, the usage of a single gene (bit) is not binary: a wide range of usage values can be observed in the corresponding usage map; this is reflected in three different colors (yellow, dark yellow and black) in the DU map. In SGE, the

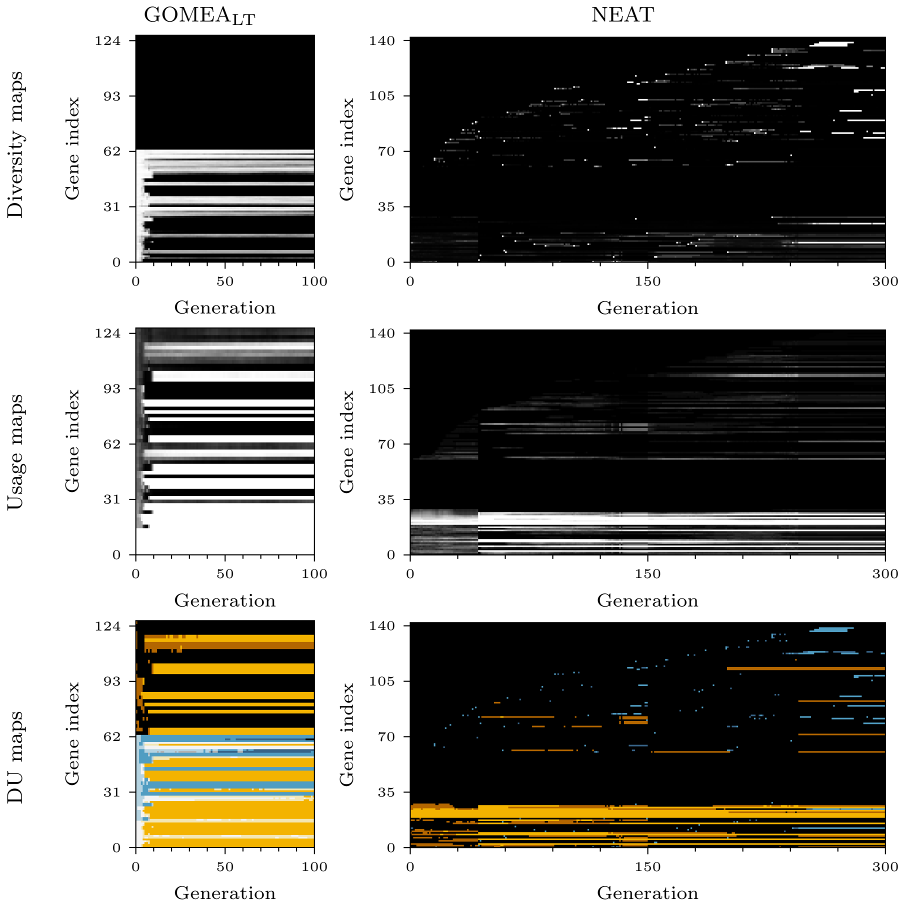

Fig. 6 Diversity map, Usage map, and DU map obtained from one run of GOMEA$_{LT}$ on the Nguyen7 problem and one run of NEAT on the Car problem.

genotype is partitioned in a number of portions corresponding to grammar non-terminal symbols (see Section 5.1.3): the DU map reflects this structure because yellow stripes highlight the fact that for each possible genotype portion (i.e., non-terminal symbol) only a small part is actually used.

With respect to GSGP, the Diversity map of Figure 5 shows that at in the early stages of evolution, individuals present a certain degree of diversity that is completely lost after (approximately) the first 10 generations. Considering the definition of the diversity function, and the behaviour of the GSOs, this finding is somewhat expected. In fact, each application of the GSOs creates a new individual that is a linear combination of the original parent individuals. In particular the offspring stands in the middle of the parents in the semantic space. With respect to the genotype, the offspring contains by construction the whole structure of the parents plus a random tree. Hence, after the application of GSOs, the defined diversity function is able to capture this aspect of the evolution. With respect to usage, only a subset of the individuals created after initialization is actually used during evolution (those corresponding to the yellow stripes). This is something that can be explained considering the constructions of the individuals in GSGP and their fitness. More in detail, each application of the GSOs produces a new individual that contains the whole structure of the parents (hence the initial genes). Moreover, the offspring of the crossover cannot be worse than the worse of its parents [37], while the mutation operator has a probability of 0.5 to improve or maintain the fitness of the current individual. Hence, generation by generation, there is an increasing number of the initial genes associated to good-quality individuals and the selection process will favour these individuals.

The DU map of a run of GOMEA$_{\text{LT}}$ on Nguyen-7 (see Figure 6) shows a dichotomy between diversity and usage: with few exceptions, active nodes quickly converge, while inactive nodes maintain diversity due to a lack of fitness contribution and thus a lack of selection bias. This can also be noticed by looking at the diversity and usage maps separately. Because the genotype of GOMEA is a level-order encoding of a tree, we can tell which parts of the tree are active in which generation, and when they converge. For instance, the evolution of the root is represented in the bottom row of the map. After a few generations, the whole population has the same node as root, and will not diverge from it. This is due to the definition of the GOM variation operator which does not allow offspring to become worse during variation. Moreover, GOM is known to induce high selection pressure, which, in combination with a FOS that fits the problem at hand well, enables the high performance of GOMEA [68]. The top half of the map shows the evolution of diversity and usage of the leaves. Because the single input variable can only be a leaf, diversity is absent. On usage, it is possible to see that in the beginning of the evolution few leaves are active, and after some generations new leaves are activated. This means that the size of individuals quickly grows and then stabilizes, together with the loss of diversity in the upper part of the tree.

Finally, concerning NEAT, the DU map in Figure 6 clearly reflects that this EA is based on a very different individual representation. In particular, the fact that NEAT genotypes are variable in size shows in the DU map by means of the peculiar shape of the non-black region. The DU map for this NEAT run shows that, on the one hand, new genes appear over the time (i.e., complexification of the NN topology happens); on the other hand, few of the innovations introduced during the evolution are actually used. We recall that in NEAT, individuals of the initial population correspond to NNs with only the input layer, the output layer, and few connections between them. The yellow stripes in the bottom-most region of the DU map suggests that, in this particular run, the initial topology essentially remains the same.

6.2 RQ2: Is the DU map useful for gaining insight into the behavior of an EA run on a problem, and for comparing problems?

In this section we show how the DU map can be used to investigate the behavior of an EA across different runs on the same problem or on different problems.

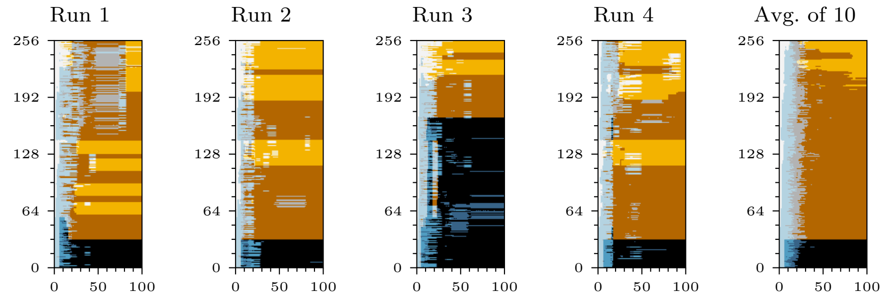

Fig. 7 DU maps obtained from different evolutions of WHGE on the Nguyen7 symbolic regression problem: the four leftmost maps are related to four different runs; the rightmost map is the average map of 10 runs.

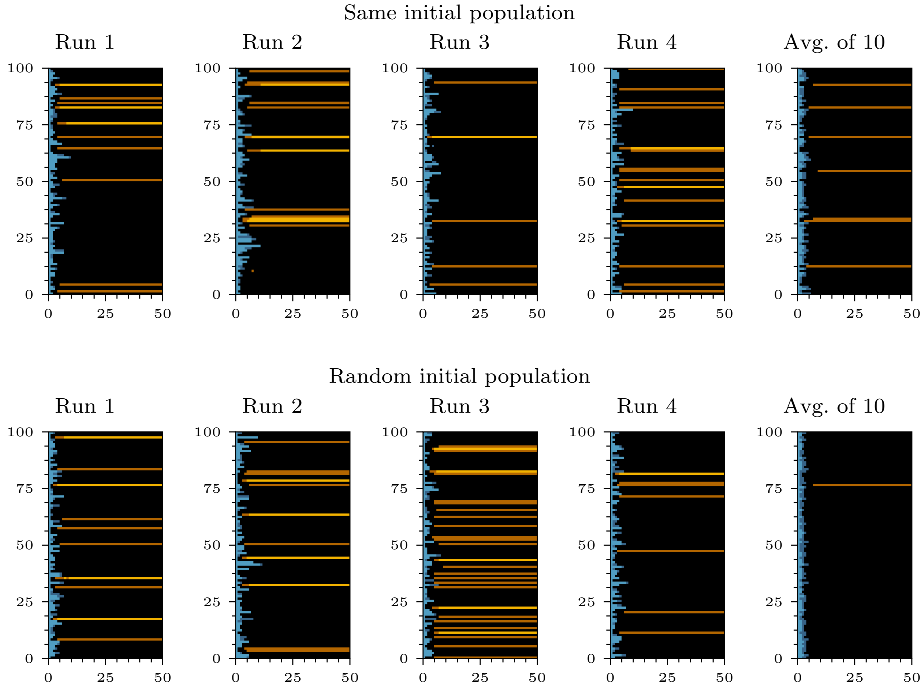

Fig. 8 DU maps obtained from different evolutions of GSGP on the Nguyen7 symbolic regression problem, with the same or random initial population: the four leftmost maps are related to four different runs; the rightmost map is the average map of 10 runs.

6.2.1 Comparing different runs

Figures 7, 8 and 9 present the results on the Nguyen7 symbolic regression problem for multiple runs of WHGE, GSGP, and GOMEA, respectively.

Figure 7 shows the DU maps for different runs of WHGE. Different from GSGP and NEAT, but similar to GOMEA (see below), in WHGE the gene index matters. Hence, by observing how the usage is distributed along the genotype one can reason

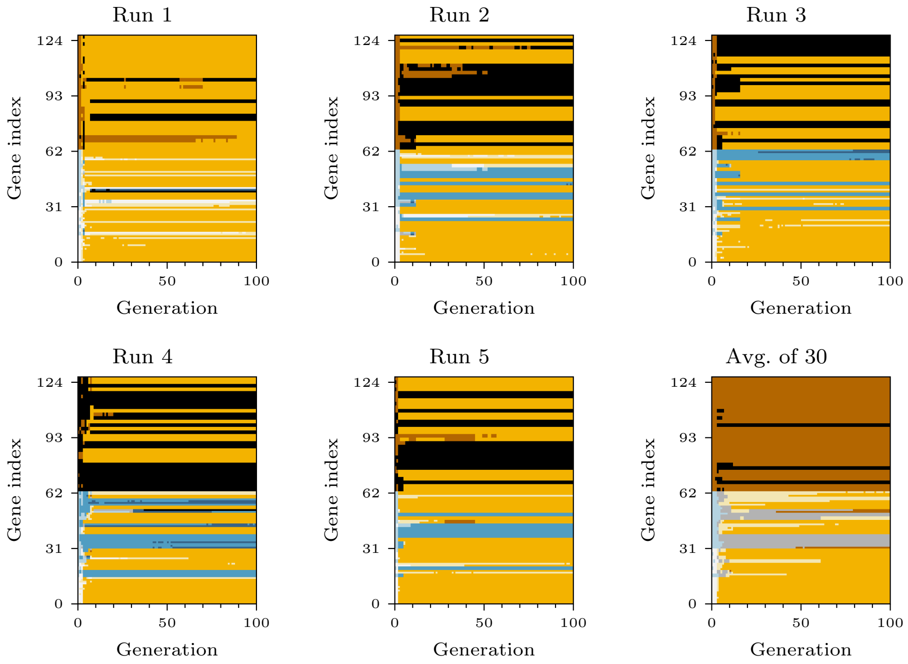

Fig. 9 DU maps obtained from different runs of GOMEA$_{\mathrm{LT}}$ on the Nguyen7 symbolic regression problem: the three maps of the top row and the first two maps of the second row are related to five different runs; the last map of the second row is the average map of 30 runs.

about the existence of local optima. For instance, Figure 7 shows that the usage in the last generations of Runs 2 and 4 is similar and close to the usage obtained by the average DU map; usage of Runs 1 and 3 is different. This may corresponds to different (local) optima (1 and 3).

Next, consider the results of GSGP presented in Figure 8. The first row shows the DU maps for GSGP runs with the same initial population, while the second row depicts the runs with different initial populations. Starting the analysis from the first row, it is possible to see that each run presents a similar degree of diversity. Moreover, diversity is lost quickly after a few generations. With respect to usage, a subset of the initial individuals is selected to build new, fitter, individuals. It is important to underline that this subset varies in a significant way in each run: both which individuals are selected as well as how many are selected, differs. Initially, all individuals have similar fitness. Hence, tournament selection initially selects different individuals with (approximately) the same probability, resulting in the observed similarity in diversity. Considering the usage and taking into account, for example, Runs 3 and 4, it is possible to see a difference in terms of usage, where in Run 3 only a few individuals are largely used, while in Run 4 there are a lot of individuals that contribute to the creation of new individuals. This behaviour is a side effect of the loss of diversity. When this problem is particularly relevant (i.e., equal semantics for a vast amount of individuals) a DU map similar to the one achieved for Run 3 is obtained. On the other hand, when the individuals present

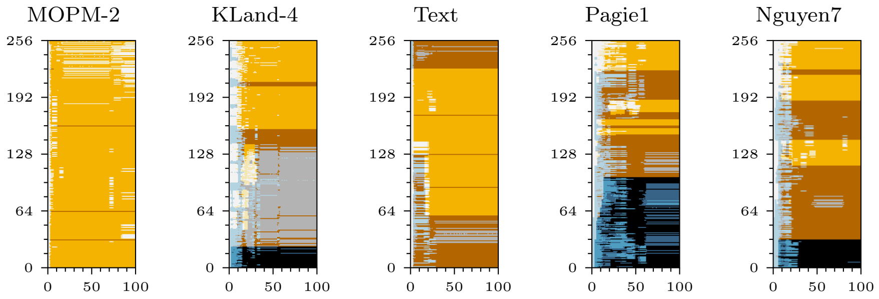

Fig. 10 DU maps obtained from evolutions with WHGE on five different problems.

different semantics a DU map similar to the one shown in Figure 8 for Run 4 is obtained.

The same considerations are valid for the second experimental setting, where different initial populations are considered (see the bottom row of Figure 8). Again, diversity is lost after a few generations, and the usage is different in each run. Also in this case, there are runs (for instance Run 3) where the majority of the individuals is used to create new individuals by applying the GSOs, and there are other runs (like Run 4) where a small subset of the initial individuals is used to drive the evolutionary process towards fitter individuals. As previously explained, this relates with the selection process and with the definition of the GSOs, where there is an increasing number of the initial genes associated to good-quality individuals.

The behavior of different runs of GOMEA$_{\text{LT}}$ on Nguyen7 is shown in Figure 9. The most evident difference among the runs is in terms of usage, with Run 1 exhibiting considerably more active nodes than the others. Despite these differences, all runs found a good approximating function (average mean squared error of 0.0002, standard deviation of 0.0003 on the test set). This suggests that the function can be efficiently approximated both by small and large trees. Considering the average DU map, common patterns of usage can be spotted by looking at yellow bands. In particular, the top of the tree is always used, followed by an intermittent use of lower nodes. Furthermore, we can observe a growth of active nodes in all maps. As to diversity, the runs show very similar behavior. The population is initially composed by different individuals, but after a few generations, most of the active nodes converge to the same value. Interestingly, we can see that few active nodes maintain their diversity during evolution. Possibly these nodes do not have an impact on the fitness (e.g., their output is multiplied by the constant 0, generated by a sibling node), and computational effort is wasted by parsing their subtree during fitness evaluations. Lastly, note that no diversity is present for the leaves, since only one node is possible (only one input variable is used as terminal).

6.2.2 Comparing different problems

This section analyzes the DU maps produced by three of the algorithms on a set of selected benchmarks (see Figures 10, 11 and 12).

The DU maps for WHGE presented in Figure 10 show how the hierarchical representation of this EA is differently exploited in each of the considered problems.

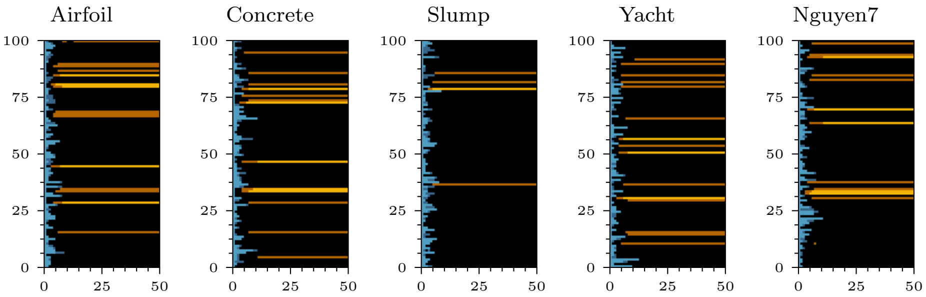

Fig. 11 DU maps obtained from evolutions with GSGP on five different problems.

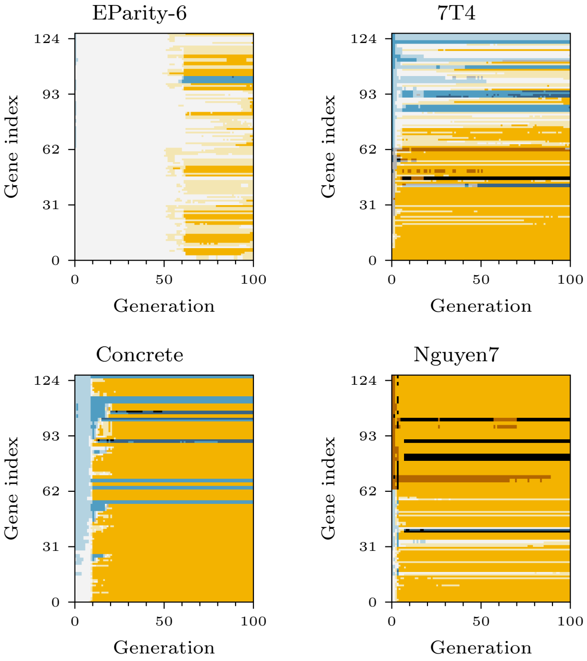

Fig. 12 DU maps obtained from evolutions with GOMEA$_{\mathrm{LT}}$ on four different problems.

We recall that in WHGE the usage of a gene (bit) is given by the number of times that bit is re-used in the genotype-phenotype mapping and this number itself depends on the depth of the corresponding part of the derivation tree. It follows that, for WHGE, the usage in the DU map gives an intuition of how balanced the derivation trees are in the population. The user might exploit this information when adapting the grammar of the problem at hand—a task for which automatic, meta-evolution approaches have already been proposed [41]. The maps of Figure 10 also show how diversity varies across the problems. E.g., in MOPM-2 there is still some diversity in highly used genes in a late stage of evolution: this is an indication that evolution is still ongoing, a finding which is consistent with the fact that MOPM-2 is an hard problem for this EA.

For GSGP, the resulting DU maps present a similar pattern with respect to both diversity and usage. In particular, it is possible to see that, over the different test problems, a subset of the individuals is used most of the times for creating newer individuals. This is a behaviour that is more evident in the yacht dataset (Figure 11) and less evident for the slump dataset. We hypothesize that this different behaviour is somehow related to the different effect that GSOs have on different problems. For instance, the effect of geometric semantic mutation is strongly dependent of the the value of the mutation step, while crossover has a different impact on the distribution of the individuals in semantic space for each problem [67]. All in all, DU maps are a suitable tool for capturing the main features of GSGP, but a future analysis aimed at investigating the effect of the GSOs on the search process is needed for a better comprehension of the semantics of DU maps for GSGP.

DU maps for one GOMEA_{LT} run on four selected problems are shown in Figure 12. Despite the considered problems being very different from each other, the algorithm shows similar behavior on 7T4, Concrete, and Nguyen7. Compared to the other experiments, in the EParity-6 map more diversity on the used portion of the genotype can be seen (a larger portion of the DU map is white). This aspect corroborates the hypothesis that it is possible to discover equally fit individuals with different genotypes due to a high redundancy in the genotype-phenotype mapping. After roughly half of the evolution, however, the individuals start to converge to the same genotype (the diversity is decreased), as shown by the white color turning into yellow. In 7T4, only one function node is possible, thus diversity in the top of the tree is only related to presence of terminal nodes, and is mostly low. For the leaves, which are depicted in the top half of he map, diversity is instead moderate to high. This is due to the fact that the fitness is based on the order of the terminal nodes, but not on their exact position. The symbolic regressions of Concrete and Nguyen7 have a similar evolution, with a brief first phase with high diversity followed by a quick convergence to the same genotype, for the majority of the nodes. Despite the fact that the Concrete dataset has eight input variables (thus eight possible terminal nodes), the evolution is very similar to the one of Nguyen7, where only one variable is possible. This observation suggests that the convergence behavior of GOMEA_{LT} is similar on different symbolic regression problems.

6.3 RQ3: Is the DU map useful for choosing EA parameters or components?

To answer this research question, we consider the behavior of GOMEA in tackling EParity-6 using different FOS choices. Picking the right FOS can be crucial for the effectiveness of GOMEA. If a DU map could provide insight into the effect of a particular FOS in terms of evolution of diversity and usage, such insights could then be used to determine which FOS to use for what problem, or how to design an improved, problem-specific FOS. A similar analysis could be performed with other EAs and different parameters or components, but this is outside the scope of this paper.

We measured the performance of 30 runs of GOMEA with the four FOS variants: LT, RT, LTd, RTd. In Table 4 the percentange of successful runs is reported (a run is successful if the perfect Boolean formula is found within the last gener-

Table 4 Success rate for GOMEA on EParity-6 problem using LT, RT, LTd, and RTd.

FOS	Success rate
LT	0.40
RT	0.23
LTd	0.87
RTd	0.03

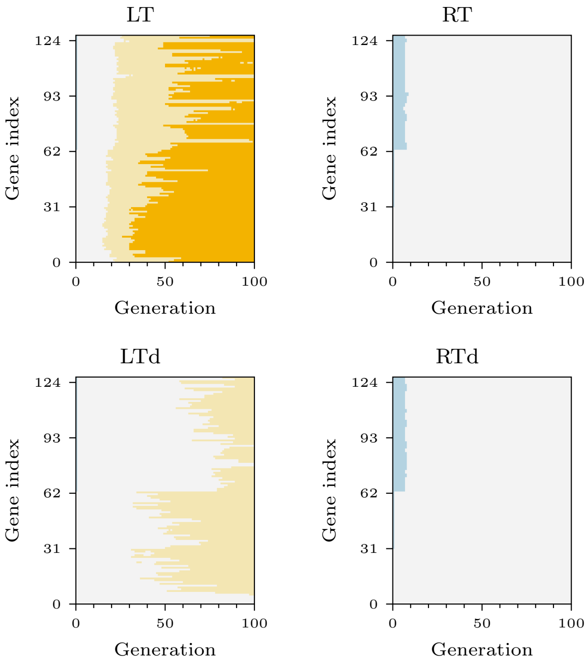

Fig. 13 Average DU maps obtained on 30 runs of GOMEA using different choices of FOS on EParity-6.

ation). These results show that using LTd leads to clearly superior performance, with a success rate of 0.87. The adoption of the LT leads to the second best performance with a success rate of 0.40, followed by the RT and RTd, with success rate of 0.23 and 0.03, respectively.

Figure 13 shows, for each FOS, a DU Map of the average of the 30 runs. The maps obtained by using RT and RTd are extremely similar. While using the RTd leads to better conservation of diversity, using the RT leads to a similar behavior. In particular, the minimum diversity of the root for the runs with RT is 0.964 in the last generation (with RTd it is 0.998, given by the initial sampling of the population). However, RT performs statistically better than RTd in terms of success rate, which is in contrast with the fact that the DU Maps are nearly identical. A possible explanation of this fact is that while not possible with RTd, with RT the root can be swapped (together with other nodes at the same time), and this allows for the discovery of more fit individuals. However, diversity can remain high, with the RT being a random set of crossover masks which are different every generation. Hence, this is a type of performance improvement that cannot necessarily be detected by studying DU maps.

When using the LT, usage follows a similar pattern, but the evolution of diversity is dramatically different. Here, GOMEA_{LT} exhibits a markedly different convergence behavior compared to GOMEA_{RT} and $\text{GOMEA}_{\text{RTd}}$. Due to the high selection pressure of GOM, the pattern of nodes with a positive contribution to the fitness quickly spread in the population, and the interdependencies among those nodes are captured by mutual information. Because the LT is built from this information the hierarchical crossover masks are made to mix individuals according to these patterns. Therefore, there is a mutual, reinforced convergence of genotype and the structure of the LT itself in GOMEA_{LT}, which results in the rapid loss of diversity. This can be clearly seen in the DU map. It should be noted that this type of diversity loss is desirable as it is the result of extremely effective mixing behavior as a result of the right patterns being present in the population and being correctly modeled by the LT. Indeed, the fast convergence of GOMEA_{LT} does not compromise the success rate when compared with GOMEA_{RT} and $\text{GOMEA}_{\text{RTd}}$. Rather, using the LT results in the second-best performance. This means that, almost half of the times (0.4 success rate), the right patterns are present in the population and are correctly modeled by the FOS. We hypothesize that either the correct information is present in the population, and a perfect solution is quickly found, or the wrong information is modeled and GOMEA_{LT} quickly converges to a suboptimal solution. If a larger population size were to be chosen, the performance of GOMEA_{LT} will increase because it enables more robust learning of the salient linkage information, and the consequent propagation of the correct patterns of nodes. This result was experimentally shown on different problems by adopting a framework of multiple interleaved runs with increasing population size (and tree height) [68]. Here, we consider only a single population size however. By looking at the DU map of GOMEA_{LT}, we can see that, on average, the population almost completely converges in roughly the first quarter of the evolution. The improvement that can be achieved during the subsequent generations is likely to be minimal. This is a key insight to improve the performance of GOMEA_{LT}: given the same budget (i.e., evaluations or time), it is better to use a bigger population size for less generations.

Together with increasing the population size, another way to provide GOMEA more time to find the salient building blocks to be used in GOM before selection drives the population to convergence, is to promote diversity. The compromise in this respect that is achieved by LTd can be seen in the respective DU map. The use of LTd results in the best performance for this problem, with a success rate close to 0.9. The simple diversity mechanism consisting of excluding the root from the FOS forces $\text{GOMEA}_{\text{LTd}}$ to look for different solutions to the problem, namely one per different value that the root can have (i.e., the number of functions in the function set). The map shows that this moderates the convergence of the whole genotype. It could be said, therefore that the DU map in this case makes the interplay between selection pressure, proper mixing, and diversity maintenance clear, and a correspondence with the ultimate EA performance can be observed. It should be noted, however, that to find which type of FOS truly leads to the best possible performance (i.e., finding the optimum within the least number of evaluations), the influence of the population size should also be considered.

7 User-based validation

We performed an online user study to assess the usability and the usefulness of the DU map. The user study consisted of a questionnaire which we built guided by the literature sources [5,7,39,46]. We shared the questionnaire through our personal network of colleagues at different research institutions and through the GP mailing list[2] collecting 34 responses. We stress the fact the respondents filled the questionnaire online without any influence from our side: i.e., the study was not a controlled user study. We made the complete set of answers publicly available[3].

The questionnaire consisted of four parts. The first one introduced the DU map, provided a tutorial of the DU map (similar to Figure 3), and asked the participants about their expertise level. The second part tested the comprehension of the DU map, showing visualizations and asking whether considerations on diversity and usage are true or false. This part aimed at assessing whether the participants were effectively capable of inferring the correct information from the visualizations. Also, the second part asked the participants which of the three different color options they used to answer the questions (Section 4.1). The third part asked the participants about their opinion on the usefulness of the DU map. The fourth and last part was the System Usability Scale (SUS) [2,5], a widely applicable questionnaire which is often adopted to assess the usability of a system.

7.1 Questionnaire results

In the first part, 67.6% of the respondents declared to be experts in EC, and only 5.9% considered themselves not experts. The remaining neither agreed nor disagreed with this definition (26.5%). For the genotype-to-phenotype mappings, 23.5% of respondents stated to be experts, while the others were equally divided between not experts and in-between (38.2% each).

The summary of the results from the comprehension tests of the second part of the questionnaire are reported in Table 5. Each of the three proposed tests contained four or five statements and the users were asked whether they agree with those statements (three possible answers were given: “yes”,“no” and “I don’t know”). For instance, one statement was “Diversity of the used genes lasts longer in the DU map 1 than in the DU map 2” in a test showing two DU maps. The table shows, for each test, the percentage of answers. Overall, participants were typically able to answer correctly, and performed slightly better on Test 1, where considerations on a single map were presented. Test 2 and 3 showed two DU maps and asked questions regarding their comparison, thus they may be considered harder than Test 1. Interestingly, the percentage of 87.3% correct answers over the three tests is substantially higher compared to the number of respondents who declared to be experts in EC (67.6%), and to the number of genotype-to-phenotype mappings experts (only 23.5%).

After each of the three tests, we asked which color scheme the respondent used to answer the questions among the three presented in Figure 4.1. Multiple answers were allowed. The most used scheme was option b, which was used 50.0% of the

[2] `http://www.genetic-programming.org/gpmailinglist.html`

[3] `http://machinelearning.inginf.units.it/data-and-tools/unveiling-evolutionary-algorithm-representation-with-du-maps`

Table 5 Percentage of correct, wrong, and unanswered responses to each comprehension test of the questionnaire.

Test	Correct	Wrong	Unanswered
1	92.9	6.5	0.6
2	82.4	9.5	8.1
3	86.5	8.8	4.7

time. Slightly less preferred was option c, with 48.0%; the last one was option a, picked 38.2% of the time. We used this feedback to improve the representation of the DU map compared to its original form [35]: we opted for option c as it was only slightly less popular than option b, but it allows people with color vision deficiencies to use the tool. We remark that the maps appeared in the questionnaire in the same order for all the participants and that it is fair to assume that most of them read from left to right. This may have resulted in a bias in the choice of the preferred color schemes and is hence a limitation of this user study.

The third part of the questionnaire consisted of four statements where the respondents could pick among five options, ranging from “completely disagree” to “completely agree”, plus two open questions about the weak and strong points of the DU map. Figure 14 shows the distribution of responses of the four statements. Overall, the responses support the idea that the DU map is an useful tool.

The majority of the respondents agreed (or totally agreed) that the DU map allows to spot the interplay between diversity and usage (Statement 1, 70.6%). This is important as it is the main contribution of the DU map. When dealing with an algorithm that is well known to the respondent, the majority thought that the DU map could still give extra insight (Statement 2, 58.9%), although one third was not sure about this (35.3%). This is the weakest score among our statements. Zooming in by taking into account expertise, it shows that users with high expertise thought that the DU map can provide further insights even for familiar algorithms in a larger percentage (65.2% and 75.0% for experts in EC and genotype-to-phenotype mapping, respectively). This suggests that more experienced users can better appreciate the contribution of the DU map. Statement 3 proposed one of the practical uses that the DU map can have, i.e., understanding if the evolution is stagnating. Respondents mostly agreed that the DU map can be useful with respect to this issue (73.5%). The last statement explored the usefulness of the discrete versions of the DU map (option b and c) compared to the continuous version (option a). The users clearly felt that the discrete versions bring enough information, which supports the adoption of the option c.

Concerning the two open questions about the weak and strong points of the DU map, we collected several interesting responses. Many respondents said that being able to understand a DU map may require some practice: e.g., “As a first time user, I kept going back and forth to the legend to decipher the graphs”, and “It takes some time to understand what is going on, if you haven’t seen them before”. On the other hand, respondents appreciated the fact that the DU map can deliver a quick overview of a run: e.g., “It concisely displays information about all generations of an algorithm”, and “A nice succinct presentation of the genotype usage during the evolution”.

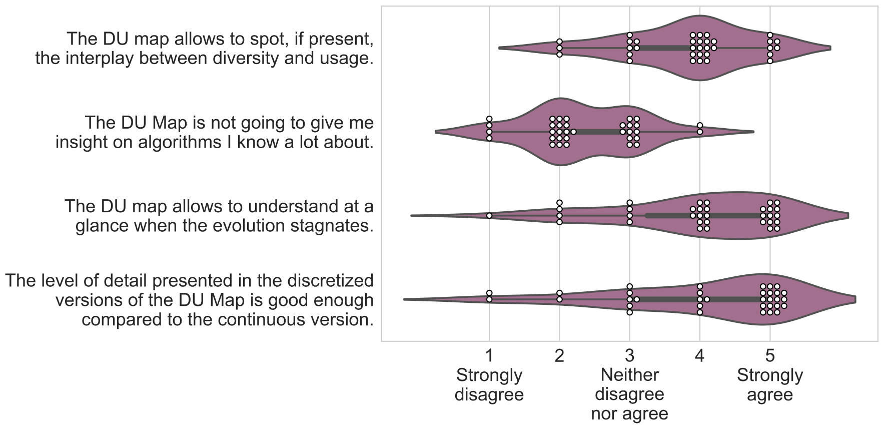

Fig. 14 Violin plot indicating the responses to the 4 statements of the third part of the questionnaire. The purple shape represents the overall distribution of the results. White points are the responses.

The fourth and last part of the questionnaire was the SUS. The SUS asked the respondents wether they agree or disagree with ten statements on the usability of a system, using the a scale of five possibilities ranging from “strongly disagree” to “strongly agree”. The SUS is used in many different domains as its questions are very general, and it can be very accurate even when a limited number of responses are collected [5]. We remark that we modified two questions of the SUS as they did not apply to the DU map, as the latter is a visualization tool rather than a system. The statement “I found the various functions in this system were well integrated” was modified to “I find different DU Maps are easy to compare”, and “I thought there was too much inconsistency in this system” was modified to “I think there is too much inconsistency to compare different DU Maps”. By using the SUS, a score ranging from 0 to 100 can be computed. The DU map scored, on average among all participants, 65.7 (with a standard deviation of 17.0). This score is considered a positive score (which start above 50), albeit not excellent [2].

8 Concluding remarks

Understanding if and why an EA is capable of (efficiently) solving a certain problem is a cornerstone research question of the field. Moreover, figuring out how to best fine-tune parameters of an EA is a time-consuming task. Tackling both of these issues could well be supported by the use of adequate and insightful visualization tools. In this article, we have considered the Diversity and Usage map (the DU map), a visualization tool that focuses specifically on representations, one of the most prominent components of any EA. In particular, the two goals targeted by the DU map, at the same time, are (1) to convey detailed information about the internal workings of an EA using key, well-understood concepts in the EA field and (2) to have the same semantics for widely varying EAs. No other visualization tools have been proposed that tackle these goals simultaneously.

We improved the DU map, hence extending the previous work by Medvet and Tušar [35], in two ways. First, we greatly extended the applicability of the DU map by making its definition more general. We also showed how to apply it to six EAs that use considerably different representations (bit and integer strings, trees, ensemble of trees, and neural networks). Second, we better grounded the DU map as a visualization tool by discussing in detail its purpose and design rationale. In particular, we performed an online user study that involved 34 EC researchers in order to assess the perceived usefulness of the DU map and validate its main design choices.

The outcome of our experimental validation and of the user study show that by inspecting DU maps users may better understand the impact of using certain representations, parameter settings, or components on the evolutionary process. As such, we believe that the DU map offers clear added value to tools that are currently available to users, practitioners, and researchers who wish to unveil key properties of the EAs that they are designing and analyzing. From a broader point of view, we have shown that visualization tools may be very useful to EC practitioners and that designing and assessing those tools based on visualization practices can improve their quality and applicability, and favor their adoption.

We also discussed possible improvements to the DU map, including those introducing some form of interaction. Motivated by the perceived usefulness of the DU map in investigating the individual representation in terms of diversity and usage, other researchers could propose similar visualization tools able to deliver in a succinct view important information about EAs and/or to aid practitioners in specific analysis tasks, e.g., fitness landscape analysis.

Acknowledgments

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 692286. This work was also financed through the Regional Operational Programme CENTRO2020 within the scope of the project CENTRO-01-0145-FEDER-000006. Marco Virgolin received financial support from the Kinderen Kankervrij foundation (project No. 187).

References

1. Amor, H.B., Rettinger, A.: Intelligent exploration for genetic algorithms: Using self-organizing maps in evolutionary computation. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '05, pp. 1531–1538. ACM (2005)
2. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. Journal of usability studies **4**(3), 114–123 (2009)
3. Barlow, M., Galloway, J., Abbass, H.A.: Mining evolution through visualization. In: Workshop Proceedings of the Eighth International Conference on Artificial Life, Alife VIII, pp. 103–110. MIT Press (2002)
4. Brewer, C.A.: Color use guidelines for data representation. In: Proceedings of the Section on Statistical Graphics, pp. 55–60. American Statistical Association (1999)
5. Brooke, J., et al.: Sus-a quick and dirty usability scale. Usability evaluation in industry **189**(194), 4–7 (1996)

6. Brooks, T.F., Pope, D.S., Marcolini, M.A.: Airfoil self-noise and prediction. Tech. Rep. NASA RP-1218, National Aeronautics and Space Administration (1989)
7. Burgess, T.F.: Guide to the design of questionnaires. A general introduction to the design of questionnaires for survey research pp. 1–27 (2001)
8. Burlacu, B., Affenzeller, M., Kommenda, M., Winkler, S.M., Kronberger, G.: Visualization of genetic lineages and inheritance information in genetic programming. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '13, pp. 1351–1358. ACM (2013)
9. Castelli, M., Silva, S., Vanneschi, L.: A C++ framework for geometric semantic genetic programming. Genetic Programming and Evolvable Machines **16**(1), 73–81 (2015)
10. Castelli, M., Vanneschi, L., Silva, S.: Prediction of high performance concrete strength using genetic programming with geometric semantic genetic operators. Expert Systems with Applications **40**(17), 6856–6862 (2013)
11. Collins, T.D.: Applying software visualization technology to support the use of evolutionary algorithms. Journal of Visual Languages and Computing **14**(2), 123–150 (2003)
12. Cruz, A., Machado, P., Assunção, F., Leitão, A.: ELICIT: Evolutionary computation visualization. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '15, pp. 949–956. ACM (2015)
13. Daida, J.M., Hilss, A.M., Ward, D.J., Long, S.L.: Visualizing tree structures in genetic programming. Genetic Programming and Evolvable Machines **6**(1), 79–110 (2005)
14. De Jong, K.A.: Evolutionary Computation: A Unified Approach. MIT Press (2006)
15. Drchal, J., Šnorek, M.: Diversity visualization in evolutionary algorithms. In: Proceedings of 41th Spring International Conference, MOSIS'07, pp. 77–84. Ostrava: MARQ (2007)
16. Eiben, A.E., Smith, J.E., et al.: Introduction to evolutionary computing, vol. 53. Springer (2003)
17. Floreano, D., Husbands, P., Nolfi, S.: Evolutionary robotics. In: Springer handbook of robotics, pp. 1423–1451. Springer (2008)
18. Hart, E., Ross, P.: GAVEL—A new tool for genetic algorithm visualization. IEEE Transactions on Evolutionary Computation **5**(4), 335–348 (2001)
19. Ito, S., Mitsukura, Y., Miyamura, H.N., Saito, T., Fukumi, M.: A visualization of genetic algorithm using the pseudo-color. In: Revised Selected Papers from the 14th International Conference on Neural Information Processing, ICONIP 2007, *Lecture Notes in Computer Science*, vol. 4985, pp. 444–452. Springer (2007)
20. Jornod, G., Mario, E.D., Navarro, I., Martinoli, A.: SwarmViz: An open-source visualization tool for particle swarm optimization. In: Congress on Evolutionary Computation, CEC 2015, pp. 179–186. IEEE (2015)
21. Keedwell, E., Johns, M.B., Savic, D.A.: Spatial and temporal visualisation of evolutionary algorithm decisions in water distribution network optimisation. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '15, pp. 941–948. ACM (2015)
22. Kerren, A., Egger, T.: EAVis: A visualization tool for evolutionary algorithms. In: 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2005), pp. 299–301. IEEE (2005)
23. Khemka, N., Jacob, C.: VISPLORE: a toolkit to explore particle swarms by visual inspection. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '09, pp. 41–48. ACM (2009)
24. Kim, Y., Lee, K.H., Yoon, Y.: Visualizing the search process of particle swarm optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '09, pp. 49–56. ACM (2009)
25. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, vol. 1. MIT press (1992)
26. Langdon, W.B.: Long-term evolution of genetic programming populations. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, pp. 235–236. ACM (2017)
27. Liu, S.H., Črepinšek, M., Mernik, M.: Analysis of VEGA and SPEA2 using exploration and exploitation measures. In: Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2012, pp. 97–108. Jožef Stefan Institute (2012)
28. Lourenço, N., Pereira, F.B., Costa, E.: Sge: A structured representation for grammatical evolution. In: Revised Selected Papers from the International Conference on Artificial Evolution (Evolution Artificielle), EA 2015, *Lecture Notes in Computer Science*, vol. 9554, pp. 136–148. Springer (2015)

29. Lutton, E., Foucquier, J., Perrot, N., Louchet, J., Fekete, J.: Visual analysis of population scatterplots. In: Revised Selected Papers from the 10th International Conference on Artificial Evolution (Evolution Artificielle), EA 2011, *Lecture Notes in Computer Science*, vol. 7401, pp. 61–72. Springer (2011)
30. Mach, M., Zetakova, Z.: Visualising genetic algorithms: A way through the labyrinth of search space, pp. 279–285. IOS Press (2002)
31. McPhee, N.F., Casale, M.M., Finzel, M., Helmuth, T., Spector, L.: Visualizing genetic programming ancestries. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '16, pp. 1419–1426. ACM (2016)
32. McPhee, N.F., Poli, R., Langdon, W.B.: Field Guide to Genetic Programming. `http://lulu.com` (2008)
33. Medvet, E.: A comparative analysis of dynamic locality and redundancy in grammatical evolution. In: Proceedings of the 20th European Conference, EuroGP 2017, pp. 326–342. Springer (2017)
34. Medvet, E.: Hierarchical grammatical evolution. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, pp. 249–250. ACM (2017)
35. Medvet, E., Tušar, T.: The DU map: A visualization to gain insights into genotype-phenotype mapping and diversity. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17, pp. 1705–1712. ACM (2017)
36. Miller, J.F., Smith, S.L.: Redundancy and computational efficiency in cartesian genetic programming. IEEE Transactions on Evolutionary Computation **10**(2), 167–174 (2006)
37. Moraglio, A., Krawiec, K., Johnson, C.: Geometric semantic genetic programming. In: Proceedings of the International Congres on Parallel Problem Solving from Nature, PPSN XII, *Lecture Notes in Computer Science*, vol. 7491, pp. 21–31. Springer (2012)
38. Mourshed, M.M., Shikder, S., Price, A.D.F.: Phi-array: A novel method for fitness visualization and decision making in evolutionary design optimization. Advanced Engineering Informatics **25**(4), 676–687 (2011)
39. Munzner, T.: Visualization analysis & design. CRC press (2014)
40. Ortigosa, I., Lopez, R., Garcia, J.: A neural networks approach to residuary resistance of sailing yachts prediction. In: Proceedings of the International Conference on Computational Methods in Marine Engineering, MARINE 2007, vol. 2007, p. 250 (2007)
41. O'Neill, M., Ryan, C.: Grammatical evolution by grammatical evolution: The evolution of grammar and genetic code. Genetic Programming pp. 138–149 (2004)
42. Pagie, L., Hogeweg, P.: Evolutionary consequences of coevolving targets. Evolutionary computation **5**(4), 401–418 (1997)
43. Parsopoulos, K.E., Georgopoulos, V.C., Vrahatis, M.N.: A technique for the visualization of population-based algorithms. In: Proceedings of the Congress on Evolutionary Computation, CEC 2008, pp. 1694–1701. IEEE (2008)
44. Pohlheim, H.: Multidimensional scaling for evolutionary algorithms - Visualization of the path through search space and solution space using Sammon mapping. Artificial Life **12**(2), 203–209 (2006)
45. Pohlheim, H.: Understanding the course and state of evolutionary optimizations using visualization: Ten years of industry experience with evolutionary algorithms. Artificial Life **12**(2), 217–227 (2006)
46. Rajanen (Marghescu), D.: Evaluating multidimensional visualization techniques in data mining tasks (2018)
47. Rothlauf, F.: Representations for genetic and evolutionary algorithms. In: Representations for Genetic and Evolutionary Algorithms, pp. 9–32. Springer Berlin Heidelberg (2006)
48. Ryan, C., Collins, J., O'Neill, M.: Grammatical evolution: Evolving programs for an arbitrary language. In: Proceedings of the First European Workshop on Genetic Programming, EuroGP'98, *Lecture Notes in Computer Science*, vol. 1391, pp. 83–96. Springer (1998)
49. Sastry, K., Goldberg, D.E.: Probabilistic model building and competent genetic programming. Genetic Programming Series **6**, 205–220 (2003)
50. Sekanina, L., Kapusta, V.: Visualisation and analysis of genetic records produced by cartesian genetic programming. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '16, pp. 1411–1418. ACM (2016)
51. Shine, W.B., Eick, C.F.: Visualizing the evolution of genetic algorithm search processes. In: International Conference on Evolutionary Computation, pp. 367–372. IEEE (1997)
52. Silva, F., Urbano, P., Correia, L., Christensen, A.L.: odNEAT: An algorithm for decentralised online evolution of robotic controllers. Evolutionary Computation **23**(3), 421–449 (2015)

53. Spector, L.: Introduction to the peer commentary special section on “On the mapping of genotype to phenotype in evolutionary algorithms” by Peter A. Whigham, Grant Dick, and James Maclaurin. Genetic Programming and Evolvable Machines **18**(3), 351–352 (2017)
54. Squillero, G., Tonda, A.: Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. Information Sciences **329**, 782–799 (2016)
55. Stanley, K.O., D’Ambrosio, D.B., Gauci, J.: A hypercube-based encoding for evolving large-scale neural networks. Artificial life **15**(2), 185–212 (2009)
56. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2), 99–127 (2002)
57. Talamini, J., Scaini, G., Medvet, E., Bartoli, A.: Selfish vs. global behavior promotion in car controller evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion (2018)
58. Thierens, D., Bosman, P.A.: Optimal mixing evolutionary algorithms. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO ’11, pp. 617–624. ACM, New York, NY, USA (2011)
59. Thierens, D., Bosman, P.A.N.: Hierarchical problem solving with the linkage tree genetic algorithm. In: Proceedings of the 15th annual conference on Genetic and evolutionary computation, pp. 877–884. ACM (2013)
60. Thorhauer, A.: On the non-uniform redundancy in grammatical evolution. In: Proceedings of the International Conference on Parallel Problem Solving from Nature, PPSN XIV, *Lecture Notes in Computer Science*, vol. 9921, pp. 292–302. Springer (2016)
61. Tóth, Z.: A graphical user interface for evolutionary algorithms. Acta Cybernetica **16**(2), 337–365 (2003)
62. Tušar, T., Filipič, B.: Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. IEEE Transactions on Evolutionary Computation **19**(2), 225–245 (2015)
63. Ulrich, T.: Pareto-set analysis: Biobjective clustering in decision and objective spaces. Journal of Multi-Criteria Decision Analysis **20**(5-6), 217–234 (2013)
64. Uy, N.Q., Hoai, N.X., O’Neill, M., McKay, R.I., Galván-López, E.: Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genetic Programming and Evolvable Machines **12**(2), 91–119 (2011)
65. Vanneschi, L., Castelli, M., Manzoni, L.: The k landscapes: A tunably difficult benchmark for genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’11, pp. 1467–1474. ACM (2011)
66. Vanneschi, L., Castelli, M., Manzoni, L., Silva, S.: A new implementation of geometric semantic GP and its application to problems in pharmacokinetics. In: Proceedings of the 16th European Conference on Genetic Programming, EuroGP 2013, *Lecture Notes in Computer Science*, vol. 7831, pp. 205–216. Springer (2013)
67. Vanneschi, L., Castelli, M., Silva, S.: A survey of semantic methods in genetic programming. Genetic Programming and Evolvable Machines **15**(2), 195–214 (2014)
68. Virgolin, M., Alderliesten, T., Witteveen, C., Bosman, P.A.N.: Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1041–1048. ACM (2017)
69. Walker, J.A., Miller, J.F.: The automatic acquisition, evolution and reuse of modules in cartesian genetic programming. IEEE Transactions on Evolutionary Computation **12**(4), 397–417 (2008)
70. Whigham, P.A., Dick, G., Maclaurin, J.: On the mapping of genotype to phenotype in evolutionary algorithms. Genetic Programming and Evolvable Machines pp. 1–9 (2017)
71. Wu, A.S., De Jong, K.A., Burke, D.S., Grefenstette, J.J., Ramsey, C.L.: Visual analysis of evolutionary algorithms. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 2, pp. 1419–1425. IEEE (1999)
72. Yeh, I.C.: Simulation of concrete slump using neural networks. Proceedings of the Institution of Civil Engineers-Construction Materials **162**(1), 11–18 (2009)