

32nd International Conference on Principles and Practice of Constraint Programming

CP 2026, July 20–23, 2026, Lisbon, Portugal

Edited by

Nicolas Beldiceanu



Editors

Nicolas Beldiceanu 

IMT Atlantique, LS2N, CNRS, Nantes, France
nicolas.beldiceanu@imt-atlantique.fr

ACM Classification 2012

Theory of computation → Constraint and logic programming; Applied computing → Operations research; Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Combinatorial optimization; Mathematics of computing → Solvers; Computing methodologies → Combinatorial algorithms; Computing methodologies → Optimization algorithms; Computing methodologies → Planning and scheduling; Computing methodologies → Theorem proving algorithms

ISBN 978-3-95977-432-1

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-432-1>.

Publication date

July, 2026

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists all publications of this volume in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.CP.2026.0

ISBN 978-3-95977-432-1

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université Paris Cité, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Holger Hermanns (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Daniel Král' (Leipzig University, DE and Max Planck Institute for Mathematics in the Sciences, Leipzig, DE – Chair)
- Sławomir Lasota (University of Warsaw, PL)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Chih-Hao Luke Ong (Nanyang Technological University, SG)
- Eva Rotenberg (IT University of Copenhagen, DK)
- Pierre Senellart (ENS, Université PSL, Paris, France)
- Alexandra Silva (Cornell University, Ithaca, US)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Nicolas Beldiceanu</i>	0:xi
List of Authors	
.....	0:xv
Senior Program Committee	
.....	0:xxi
Program Committee	
.....	0:xxiii
Additional Reviewers	
.....	0:xxvii

Invited Talks

Decision Diagrams for Constraint Reasoning and Optimization	
<i>Willem-Jan van Hoeve</i>	1:1–1:1
Bridging the Gap: Foundedness, Defaults, and Expressivity in Constraint Answer Set Programming	
<i>Torsten Schaub</i>	2:1–2:1

Regular Papers

Approximating Time-Dependent Transition Times in Constraint Programming for an Earth Observation Mission	
<i>Romain Barrault, Cédric Pralet, Gauthier Picard, and Eric Sawyer</i>	3:1–3:17
SCENAGEN: A CP Model for Grounding Qualitative Driving Scenarios	
<i>Nassim Belmecheri, Arnaud Gotlieb, Nadjib Lazaar, and Helge Spieker</i>	4:1–4:18
Satisfiability for Large Weight Syndrome Decoding	
<i>Carl Berton, Sami Cherif, and Claire Delaplace</i>	5:1–5:17
Table Constraints for Integer Programming	
<i>Hendrik Bierlee, Wout Piessens, Tias Guns, and Peter J. Stuckey</i>	6:1–6:19
Revisiting Optional Variables in Lazy Clause Generation Solvers for Flexible Scheduling	
<i>Arthur Bit-Monnot</i>	7:1–7:23
Classification of Non-Redundancy of Boolean Predicates of Arity 4	
<i>Joshua Brakensiek, Venkatesan Guruswami, and Aaron Putterman</i>	8:1–8:24
Equal-Sized Partition Problem: Application in Spinning and Yarn Production	
<i>Quoc-Trung Bui</i>	9:1–9:14



Modelling and Optimizing HVAC Systems for Early-Stage Building Design <i>Victor Calixto, Camilo Cruz Gambardella, Amin Karimi, Pierre Le Bodic, and Allen Z. Zhong</i>	10:1–10:18
Solving the Multiple Constant Multiplication Problem with Constraint Programming <i>Théo Cantaloube, Xiao Peng, Christine Solnon, and Anastasia Volkova</i>	11:1–11:19
Linear-Memory Beam Search Algorithms in Domain-Independent Dynamic Programming <i>Yuxiao Chen and J. Christopher Beck</i>	12:1–12:21
Augmenting the Cumulative Overload Check with Integral Resource Usage Reasoning <i>Samuel Cloutier and Claude-Guy Quimper</i>	13:1–13:17
Combining an ϵ -Constraint Method with the Pareto Global Constraint <i>Manuel Combarro Simón, Pierre Talbot, and Pascal Bouvry</i>	14:1–14:18
Fairness of Classifiers in the Presence of Constraints Between Features <i>Martin C. Cooper and Imane Bousdira</i>	15:1–15:18
An Offline Neuro-Symbolic Football Pattern Retrieval Approach Using Constraint Programming <i>Augustin Crespin and Pierre Schaus</i>	16:1–16:22
From LLM Suggestions to Lean Proofs: Verified Redundant Constraints for MiniZinc <i>Philipp Danzinger and Nysret Musliu</i>	17:1–17:19
Neurosymbolic Large Neighbourhood Search <i>Arnaud Delage-Reid, Gilles Pesant, and Amal Zouaq</i>	18:1–18:18
A Scheduling-Based Approach to Railway Capacity Estimation for Industrial Freight Junctions <i>Luca Di Gaspero, Alessia Grosso, and Giovanni Longo</i>	19:1–19:17
Scaling Industrial Logistics: Tackling Multi-Batching Problems via Sequential Solving <i>Emmanuelle Dietz, Guillaume Pováda, Karl Henning, and Clara Buire</i>	20:1–20:16
End-to-End Certified Graph Colouring <i>Simon Dold, George Katsirelos, Wietze Koops, Magnus O. Myreen, Jakob Nordström, Andy Oertel, and Yong Kiam Tan</i>	21:1–21:27
Constraint Programming for Curriculum-Based High School Timetabling with Half-Blocks <i>Bérénice Dubois, Stephen Walsh, and Quentin Cappart</i>	22:1–22:17
Assembly Line Balancing with Parallel Stations and Shared Resources: A Cycle-Relative Constraint Programming Approach <i>Diego Olivier Fernandez Pons and Pierre Schaus</i>	23:1–23:16
Formally Verified Certification of Constraint Programming Proofs <i>Maarten Flippo, Konstantin Sidorov[1], Tip ten Brink, Clément Pit-Claudel, and Emir Demirović</i>	24:1–24:23

From CP Modeling to Preference Elicitation in HMLV Assembly Problems <i>Marco Foschini, Emilio Gamba, Lucas Kletzander, and Tias Guns</i>	25:1–25:20
GRID: Graph-Based Modelling Interface for Domain-Independent Dynamic Programming <i>Fabio Giordana, Zeynep Kiziltan, and Ryo Kuroiwa</i>	26:1–26:23
Utility-Peak Itemset Mining with Constraint Programming <i>Chaima Hamdi, Nadjib Lazaar, Nassim Belmecheri, Djawad Bekkoucha, Said Jabbour, and Lakhdar Saïs</i>	27:1–27:23
Accelerating Constraint Programming Solver with Parallel External Heuristics: Experiments on Scheduling and Routing Problems <i>Vilém Heinz, Šimon Zvára, Vít Knobloch, Zdeněk Hanzálek, and Petr Vilím</i>	28:1–28:19
Efficient Explanations for Rule Ensembles <i>Hao Hu, Alexey Ignatiev, and Joao Marques-Silva</i>	29:1–29:20
Modeling the p-Dispersion Problem with Distance Constraints <i>Panteleimon Iosif, Nikolaos Ploskas, Kostas Stergiou, and Dimos Tsouros</i>	30:1–30:20
Paratransit Optimization with Constraint Programming: A Case Study in Savannah, Georgia <i>Liam Jagrowski, Kevin Dalmeijer, Tinghan Ye, and Pascal Van Hentenryck</i>	31:1–31:16
On the Self-Stabilization of Dijkstra’s Asynchronous Token Circulation <i>Asma Khoualdia, Sami Cherif, Stéphane Devismes, and Léo Robert</i>	32:1–32:20
Smart Cubing for Graph Search: A Comparative Study <i>Markus Kirchweber, Tomáš Peitl, Stefan Szeider, and Hai Xia</i>	33:1–33:19
CREWAID: Interactive Optimisation for Human-In-The-Loop Crew Rostering and Rerostering <i>Matthias Klapperstueck, Frits de Nijs, Ilankaikone Senthooran, Matteo Miceli, and Michael Wybrow</i>	34:1–34:23
Automatic Relaxation and Multi-Armed Bandit Learning for Large Neighbourhood Search <i>Frej Knutar Lewander, Pierre Flener, Justin Pearson, and Peter J. Stuckey</i>	35:1–35:17
Optimizing a Multi-Commodity Home-Delivery and Pickup Service in Depopulated Rural Areas with Constraint Programming <i>Ryo Kuroiwa, Tomoki Hasegawa, Eiji Ueda, Naoki Akiyama, and Akira Yoshioka</i>	36:1–36:22
Column Generation with Domain-Independent Dynamic Programming <i>Ryo Kuroiwa and Edward Lam</i>	37:1–37:24
On the Effect of Training Data Selection in Automated Algorithm Selection <i>Erdem Kuş, Özgür Akgün, Nguyen Dang, Lars Kotthoff, and Ian Miguel</i>	38:1–38:22
Not All Restarts Are Equal: MAB-Learning at the Right Time Scale for SAT <i>Jinghu Liang, Sami Cherif, and Chu-Min Li</i>	39:1–39:18
Configuration of Heterogeneous Agent Fleets for Cognitively Demanding Missions <i>Lucas Ligny, Stéphanie Roussel, Gauthier Picard, and Élise Vareilles</i>	40:1–40:22

Learning Unified Graph and Language Representations for SMT Algorithm Selection <i>Zhengyang Lu, Paul Sarnighausen-Cahn, Jiahao Chen, Arie Gurfinkel, Florin Manea, and Vijay Ganesh</i>	41:1–41:23
From Literals to Atomic Constraints: Generalising Conflict-Driven Clause Learning for Constraint Programming <i>Imko Marijnissen, Maarten Flippo, and Emir Demirović</i>	42:1–42:21
Proof Logging for Projected Enumeration (and Counting?) Problems in VeriPB <i>Ciaran McCreesh, Jakob Nordström, Andy Oertel, and Yong Kiam Tan</i>	43:1–43:21
The Voice of the Flow: A Graph-Based Approach for Step-Wise Explanations of Constraint Satisfaction Problems <i>Minh Anh Nguyen, Tien Mai, and Hoong Chuin Lau</i>	44:1–44:23
Instance Space Analysis and Complexity Estimation for Scheduling Problems <i>Christian Pérez, Isabel Catalá, Unai López, and Miguel A. Salido</i>	45:1–45:19
Constraint Programming for Mixed-Model Assembly Line Scheduling with Complex Industrial Constraints <i>Guillaume Pováda, Javier Buil Tejero, and Tamara Borreguero Sanchidrian</i>	46:1–46:21
Revisiting Integer Programming Encodings of Acyclicity <i>Masood Feyzbakhsh Rankooh and Matti Järvisalo</i>	47:1–47:20
A Constraint Programming Approach for n -Day Lookahead Playoff Clinching in the NHL <i>Gili Rosenberg, Kyle E. C. Booth, J. Kyle Brubaker, and Ruben S. Andrist</i>	48:1–48:18
The Distance Constraint on Sequence Variables <i>Margaux Schmied, Augustin Delecluse, Jean-Charles Régin, and Pierre Schaus</i> ...	49:1–49:19
On Inferring Cumulative Constraints <i>Konstantin Sidorov</i>	50:1–50:23
Computing Short SAT Implicants via Ising/QUBO Encodings <i>Giuseppe Spallitta, Leonardo Duenas-Osorio, and Moshe Y. Vardi</i>	51:1–51:21
VIPR Certificate Construction from Black-Box ILP Solvers <i>Stefan Szeider</i>	52:1–52:14
GPU-Accelerated Relaxed Decision Diagrams for Branch-and-Bound Optimization <i>Fabio Tardivo, Laurent Michel, and Willem-Jan van Hoeve</i>	53:1–53:16
On Symmetries and Transformations <i>Daimy Van Caudenberg, Markus Anders, and Bart Bogaerts</i>	54:1–54:22
An Automata-Based Constraint Programming Framework for Optimal Classical Planning <i>Damien Van Meerbeeck, Arnaud Lequen, Gilles Pesant, and Jendrik Seipp</i>	55:1–55:18
Constraint Solving and Particle Swarm Optimization for Fixture Layout Optimization <i>Anna Vitali, Roberto Amadini, Vittorio Maniezzo, and Maurizio Gabbrielli</i>	56:1–56:18

Identifying Multi-Hit Cancer Drivers Without Massive Parallelization: A CP, MIP, and Column Generation Framework <i>Rick S. H. Willemsen, Tenindra Abeywickrama, and Ramu Anandakrishnan</i>	57:1–57:21
A Two-Stage Constraint Programming Approach for Artificial Teeth Scheduling <i>Felix Winter</i>	58:1–58:23
Lightweight Look-Ahead-Based Value Heuristics for Constraint Optimization Problems <i>Ziyang Yu and Hongbo Li</i>	59:1–59:16
Enhanced Lower Bound Computation in Branch-and-Bound for MaxSAT <i>Jialu Zhang, Chu-Min Li, Sami Cherif, and Shuolin Li</i>	60:1–60:18
Constraint-Aware Self-Supervised Learning for Edge Selection <i>Xinda Zheng, Frits de Nijs, and Edward Lam</i>	61:1–61:18

Short Papers

Towards Step-Wise Explanations of Large Search Trees <i>Ignace Bleukx, Peter J. Stuckey, and Tias Guns</i>	62:1–62:11
The CP Shortcut: Solving the High-Power Pump Activation Problem Without the Overkill <i>Mohamed-Anass Gallass, Philippe Greiner, Antoine Tuerlinckx, and H��l��ne Verhaeghe</i>	63:1–63:13
An Iterative Constraint Programming Approach to Integrate Maximum Workload Constraints in Preemptive Jobshop Scheduling <i>Tanguy Terrien and Cyrille Briand</i>	64:1–64:11

■ Preface

This volume contains the proceedings of the *32nd International Conference on Principles and Practice of Constraint Programming* (CP 2026), which took place at the University of Lisbon, Portugal, from the 20th to the 23rd of July 2026. Held annually, CP is the premier international conference on constraint programming. This year CP was part of the *Federated Logic Conference* (FLoC) which is held every four years and brings together leading international conferences related to logic for computer science. FLoC 2026 included 10 conferences, and CP was colocated with the *11th International Conference on Formal Structures for Computation and Deduction* (FSCD), the *42nd International Conference on Logic Programming* (ICLP), the *23rd International Conference on Principles of Knowledge Representation and Reasoning* (KR), the *41st Annual Symposium on Logic in Computer Science* (LICS), and the *29th International Conference on Theory and Applications of Satisfiability Testing* (SAT). More details about the CP 2026 conference can be found at <https://cp2026.a4cp.org/>.

CP 2026 received a total of 138 submissions of original unpublished research papers. Table 1 provides the number of submitted and accepted papers per keyword.

■ **Table 1** Overview of number of submitted and accepted papers for the different keywords

Keyword	Submitted	Accepted
Application	46	25
Operations Research & Mathematical Optimisation	40	20
Modelling & Modelling Languages	32	14
Machine Learning & Generative AI	31	10
Search (Complete & Heuristic)	30	14
Solver	28	11
SAT	21	7
Constraint Propagation	18	9
Verification	15	4
Theory	14	3
Explanation & Inference	13	7
Hybrid CP/SAT	11	5
Hybrid CP/LS	8	5
Certification	8	4
Symmetries	8	3
Filtering algorithms	7	4
Decision Diagrams	7	3
Model/Instance Explainability	7	3
Dynamic Programming	6	4
MaxSAT	6	1
Logic Programming	3	1
Quantum	3	1
Security & Parallelism	3	1
SMT	3	1
Continuous & Numerical Constraints	3	0
Testing	3	0
Model Counting	2	0
Dominance	1	1
Hardware	1	0

Following an initial screening that filtered out two papers for violating anonymity guidelines or falling outside the scope of the conference, 136 submissions were sent out for double-blind peer review. Each paper received three or four reviews from members of the Program Committee. During this detailed review process, it was discovered that one

32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).

Editor: Nicolas Beldiceanu



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

submission failed to meet critical submission criteria—specifically regarding bibliographic integrity, as it contained AI-fabricated references—and was subsequently rejected. Authors had the opportunity to answer the questions posed by the reviewers in their reviews during an author response phase. This was followed by a discussion of each paper among the reviewers, the member of the Senior Program Committee assigned to the paper, and in some cases the Program Chair when an additional review was required to reach a consensus.

After that, the Senior Program Committee member wrote a meta-review which summarised the discussion, the pros and cons of the paper, and either made a recommendation to accept/reject the paper or marked it as borderline. The latter generated extra discussions with me as Program Chair. At the end of this process, 62 papers were accepted for presentation at the conference and publication in these proceedings. Of these, 59 are regular length papers while 3 are short ones.

Two subcommittees from the Senior Program Committee were formed to select the best technical and application papers.

- The first committee awarded the Best Technical Paper Award to Fabio Tardivo, Laurent Michel, and Willem-Jan van Hoeve for their paper “*GPU-Accelerated Relaxed Decision Diagrams for Branch-and-Bound Optimization*”, and a runner-up mention to Martin C. Cooper and Imane Bousdira for their paper “*Fairness of Classifiers in the Presence of Constraints between Features*”. It also awarded the Best Student Paper Award to Théo Cantaloube, Xiao Peng, Christine Solnon, and Anastasia Volkova for their paper “*Solving the Multiple Constant Multiplication Problem with Constraint*”, and a runner-up mention to Tanguy Terrien and Cyrille Briand for their paper “*An Iterative Constraint Programming Approach to Integrate Maximum Workload Constraints in Preemptive Jobshop Scheduling*”.
- The second committee awarded the Best Application Paper Award to Matthias Klapperstueck, Frits de Nijs, Ilankaikone Senthooran, Matteo Miceli, and Michael Wybrow for their paper “*CrewAid: Interactive Optimisation for Human-In-The-Loop Crew Rostering and Rerostering*”, and a runner-up mention to Luca Di Gaspero, Alessia Grosso, and Giovanni Longo for their paper “*A Scheduling-Based Approach to Railway Capacity Estimation for Industrial Freight Junctions*”.

It is traditional for CP to dedicate an entire day before the accepted papers are presented to a series of workshops where the latest ideas on emerging/focus areas are discussed, and to the doctoral program where PhD students come together to exchange ideas and network both with their peers and with senior researchers. The co-location gave us an opportunity to have a CP/SAT doctoral program co-organised by André Schidler and Mohamed Siala. The selection and organisation of the workshops and panels was chaired by Emmanuel Hebrard, while the selection of the invited tutorials was handled by Arnaud Malapert. My thanks to all of them for their wonderful work.

A conference like CP relies on the support of many people to make it a success. First, I would like to thank everybody who contributed to the submitted papers, as without them there would be no conference. Then, of course, I would like to thank the Program Committee, the additional reviewers, and the Senior Program Committee for their careful reviews, meta-reviews and subsequent discussions. I was deeply impressed by the commitment of our reviewers; with very few exceptions due to unforeseen circumstances, reports were submitted promptly, and we were able to mobilise fifteen additional reviews at short notice. It was a pleasure to see engaging, respectful and insightful discussions where reviewers were willing to consider other points of view and even change their mind when presented with the right argument. A special thanks goes to the members of the Senior Program Committee for

■ List of Authors

Tenindra Abeywickrama  (57)

RIKEN Center for Computational Science,
Kobe, Japan

Özgür Akgün  (38)

School of Computer Science, University of St
Andrews, Scotland, UK

Naoki Akiyama (36)

Toyota Motor Corporation, Tokyo, Japan

Roberto Amadini  (56)

Alma Mater Studiorum University of Bologna,
Italy

Ramu Anandakrishnan  (57)

Virginia Tech, Edward Via College of
Osteopathic Medicine, Blacksburg, VA, USA

Markus Anders  (54)

RPTU Kaiserslautern-Landau, Kaiserslautern,
Germany

Ruben S. Andrist  (48)

Amazon Advanced Solutions Lab, Seattle, WA,
98170, USA

Romain Barrault  (3)

DTIS, ONERA, Université de Toulouse, 31000,
Toulouse, France

J. Christopher Beck  (12)

University of Toronto, Canada

Djawad Bekkoucha  (27)

LISN, CNRS, Paris-Saclay University, France

Nassim Belmecheri  (4, 27)

Simula Research Laboratory, Oslo, Norway

Carl Berton  (5)

MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Hendrik Bierlee  (6)

KU Leuven, Belgium

Arthur Bit-Monnot  (7)

LAAS-CNRS, Université de Toulouse, INSA,
CNRS, France

Ignace Bleux  (62)

Department of Computer Science, KU Leuven,
Belgium

Bart Bogaerts  (54)

KU Leuven, Dept. Computer Science, Leuven,
Belgium; Vrije Universiteit Brussel, Brussels,
Belgium

Kyle E. C. Booth  (48)

Amazon Advanced Solutions Lab, Seattle, WA,
98170, USA

Imane Bousdira  (15)

IRIT, Toulouse INP, France

Pascal Bouvry  (14)

University of Luxembourg, Esch-sur-Alzette,
Luxembourg

Joshua Brakensiek  (8)

Department of Electrical Engineering and
Computer Sciences, University of California,
Berkeley, Berkeley, CA, USA

Cyrille Briand  (64)

LAAS-CNRS, Université de Toulouse, France

J. Kyle Brubaker  (48)

Amazon Advanced Solutions Lab, Seattle, WA,
98170, USA

Quoc-Trung Bui  (9)

SoICT, Hanoi University of Science and
Technology, Vietnam

Clara Buire  (20)

Airbus Central R&T, Toulouse, France

Victor Calixto  (10)

Building 4.0 CRC, Caulfield East, Victoria,
Australia; Future Building Initiative, Monash
Art, Design and Architecture, Monash
University, Caulfield, Australia

Théo Cantaloube  (11)

INSA Lyon, CITI, UR3720, 69621 Villeurbanne,
France

Quentin Cappart  (22)

UCLouvain, Louvain-la-Neuve, Belgium;
Polytechnique Montréal, Montreal, Canada;
CIRRELT, Montreal, Canada

Isabel Catalá  (45)

Universitat Politècnica de València, Spain

Jiahao Chen  (41)

Georgia Institute of Technology, Atlanta, GA,
USA

32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).

Editor: Nicolas Beldiceanu



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Yuxiao Chen  (12)
University of Toronto, Canada

Sami Cherif  (5, 32, 39, 60)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Samuel Cloutier  (13)
Université Laval, Québec, Canada

Manuel Combarro Simón  (14)
University of Luxembourg, Esch-sur-Alzette,
Luxembourg; Interdisciplinary Centre for
Security, Reliability and Trust (SnT),
Esch-sur-Alzette, Luxembourg

Martin C. Cooper  (15)
IRIT, University of Toulouse, France

Augustin Crespin  (16)
ICTEAM, UCLouvain, Louvain-la-Neuve,
Belgium

Kevin Dalmeijer  (31)
H. Milton Stewart School of Industrial and
Systems Engineering, Georgia Tech, Atlanta,
GA, USA

Nguyen Dang  (38)
School of Computer Science, University of St
Andrews, Scotland, UK

Philipp Danzinger  (17)
Institute of Logic and Computation, TU Wien,
Favoritenstraße 9-11, 1040 Vienna, Austria

Frits de Nijs  (34, 61)
Department of Data Science and AI, Faculty of
IT, Monash University, Clayton, Australia; ARC
Industrial Training and Transformation Centre
OPTIMA, Clayton, Australia

Arnaud Delage-Reid  (18)
Polytechnique Montréal, Canada

Claire Delaplace  (5)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Augustin Delecluse  (49)
KU Leuven, Belgium

Emir Demirović  (24, 42)
Delft University of Technology, The Netherlands

Stéphane Devismes  (32)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Luca Di Gaspero  (19)
DPIA – Università degli Studi di Udine, Italy

Emmanuelle Dietz  (20)
Airbus Central R&T, Hamburg, Germany

Simon Dold  (21)
University of Basel, Switzerland

Bérénice Dubois (22)
Polytechnique Montréal, Montreal, Canada

Leonardo Duenas-Osorio  (51)
Rice University, Houston, TX, USA

Diego Olivier Fernandez Pons  (23)
OptalCP, Paris, France

Pierre Flener  (35)
Uppsala University, Department of Information
Technology, Uppsala, Sweden

Maarten Flippo  (24, 42)
Delft University of Technology, The Netherlands

Marco Foschini  (25)
DTAI, KU Leuven, Belgium

Maurizio Gabbrielli  (56)
Alma Mater Studiorum University of Bologna,
Italy

Mohamed-Anass Gallass (63)
UCLouvain, Louvain-la-Neuve, Belgium

Emilio Gamba  (25)
Productions, Flanders Make VZW, Lommel,
Belgium

Camilo Cruz Gambardella  (10)
Building 4.0 CRC, Caulfield East, Victoria,
Australia; Future Building Initiative, Monash
Art, Design and Architecture, Monash
University, Caulfield, Australia; Department of
Architecture, Monash University, Caulfield,
Australia

Vijay Ganesh  (41)
Georgia Institute of Technology, Atlanta, GA,
USA

Fabio Giordana  (26)
Department of Computer Science and
Engineering, University of Bologna, Bologna,
Italy

Arnaud Gotlieb  (4)
Simula Research Laboratory, Oslo, Norway

Philippe Greiner (63)
Technord, Belgium

Alessia Grosso  (19)
DEAMS – Università degli Studi di Trieste, Italy

Tias Guns  (6, 25, 62)
KU Leuven, Belgium

Arie Gurfinkel  (41)
University of Waterloo, Canada

Venkatesan Guruswami  (8)
Simons Institute for the Theory of Computing
and the University of California, Berkeley,
Berkeley, CA, USA

Chaima Hamdi  (27)
LISN, CNRS, Paris-Saclay University, France

Zdeněk Hanzálek  (28)
Czech Institute of Informatics, Robotics and
Cybernetics, Czech Technical University in
Prague, Czech Republic

Tomoki Hasegawa (36)
Future Technology Laboratories INC., Nagoya,
Japan

Vilém Heinz  (28)
Faculty of Electrical Engineering, Czech
Technical University in Prague, Czech Republic;
Czech Institute of Informatics, Robotics and
Cybernetics, Czech Technical University in
Prague, Czech Republic

Karl Henning  (20)
Airbus Central R&T, Hamburg, Germany

Hao Hu  (29)
University of Lleida, Lleida, Spain

Alexey Ignatiev  (29)
Monash University, Melbourne, Australia

Panteleimon Iosif  (30)
University of Western Macedonia, Kozani,
Greece

Said Jabbour  (27)
CRIL, CNRS - Université d'Artois, Lens, France

Liam Jagrowski  (31)
H. Milton Stewart School of Industrial and
Systems Engineering, Georgia Tech, Atlanta,
GA, USA

Matti Järvisalo  (47)
HIIT, Department of Computer Science,
University of Helsinki, Finland

Amin Karimi  (10)
Building 4.0 CRC, Caulfield East, Victoria,
Australia; Department of Data Science and
Artificial Intelligence, Monash University,
Clayton, Australia; ARC Training Centre in
Optimisation Technologies, Integrated
Methodologies, and Applications (OPTIMA),
Melbourne, Australia

George Katsirelos  (21)
MIA Paris, INRAE, Université Paris-Saclay,
France; Université de Toulouse, ANITI, France

Asma Khoualdia  (32)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Markus Kirchweger  (33)
Algorithms and Complexity Group, TU Wien,
Austria

Zeynep Kiziltan  (26)
Department of Computer Science and
Engineering, University of Bologna, Bologna,
Italy

Matthias Klapperstueck  (34)
Department of Human-Centred Computing,
Faculty of IT, Monash University, Clayton,
Australia

Lucas Kletzander  (25)
DBAI, TU Wien, Austria

Vít Knobloch (28)
Faculty of Electrical Engineering, Czech
Technical University in Prague, Czech Republic

Frej Knutar Lewander  (35)
Uppsala University, Department of Information
Technology, Uppsala, Sweden

Wietze Koops  (21)
Lund University, Sweden; University of
Copenhagen, Denmark

Lars Kotthoff  (38)
School of Computer Science, University of St
Andrews, Scotland, UK

Ryo Kuroiwa  (26, 36, 37)
Principles of Informatics Division, National
Institute of Informatics, Tokyo, Japan; The
Graduate University for Advanced Studies,
SOKENDAI, Kanagawa, Japan

Erdem Kuş  (38)
School of Computer Science, University of St
Andrews, Scotland, UK

- Edward Lam  (37, 61)
Department of Data Science and Artificial Intelligence, Monash University, Melbourne, Australia
- Hoong Chuin Lau  (44)
School of Computing and Information Systems, Singapore Management University, Singapore
- Nadjib Lazaar  (4, 27)
LISN, Université Paris-Saclay, Paris, France
- Pierre Le Bodic  (10)
Building 4.0 CRC, Caulfield East, Victoria, Australia; Department of Data Science and Artificial Intelligence, Monash University, Clayton, Australia
- Arnaud Lequen  (55)
Linköping University, Sweden
- Chu-Min Li  (39, 60)
MIS UR 4290, Université de Picardie Jules Verne, Amiens, France
- Hongbo Li  (59)
School of Computing and Artificial Intelligence, Shanghai University of Finance and Economics, Shanghai, China
- Shuolin Li  (60)
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
- Jinghu Liang  (39)
MIS UR 4290, Université de Picardie Jules Verne, Amiens, France
- Lucas Ligny  (40)
DTIS, ONERA, Université de Toulouse, Toulouse, France; CGI, IMT Mines Albi, Université de Toulouse, Albi, France
- Giovanni Longo  (19)
DEAMS – Università degli Studi di Trieste, Italy
- Zhengyang Lu  (41)
University of Waterloo, Canada
- Unai López  (45)
Universitat Politècnica de València, Spain
- Tien Mai  (44)
School of Computing and Information Systems, Singapore Management University, Singapore
- Florin Manea  (41)
University of Göttingen and CIDAS, Göttingen, Germany
- Vittorio Maniezzo  (56)
Alma Mater Studiorum University of Bologna, Italy
- Imko Marijnissen  (42)
Delft University of Technology, Netherlands
- Joao Marques-Silva  (29)
ICREA, University of Lleida, Lleida, Spain
- Ciaran McCreesh  (43)
University of Glasgow, Scotland, UK
- Matteo Miceli (34)
Australian Operations, Woodside Energy Ltd., Perth, Australia
- Laurent Michel  (53)
Synchrony Chair in Cybersecurity, U. of Connecticut, School of Computing, Storrs, CT, USA
- Ian Miguel  (38)
School of Computer Science, University of St Andrews, Scotland, UK
- Nysret Musliu  (17)
Institute of Logic and Computation, TU Wien, Favoritenstraße 9-11, 1040 Vienna, Austria
- Magnus O. Myreen  (21)
Chalmers University of Technology, Gothenburg, Sweden; University of Gothenburg, Sweden
- Minh Anh Nguyen  (44)
School of Computing and Information Systems, Singapore Management University, Singapore
- Jakob Nordström  (21, 43)
University of Copenhagen, Denmark; Lund University, Sweden
- Andy Oertel  (21, 43)
Lund University, Sweden; University of Copenhagen, Denmark
- Justin Pearson  (35)
Uppsala University, Department of Information Technology, Uppsala, Sweden
- Tomáš Peitl  (33)
Algorithms and Complexity Group, TU Wien, Austria
- Xiao Peng  (11)
LAAS-CNRS, University of Toulouse, Toulouse, France
- Gilles Pesant  (18, 55)
Polytechnique Montréal, Canada; IVADO, Montréal, Canada

Gauthier Picard  (3, 40)
DTIS, ONERA, Université de Toulouse, 31000,
Toulouse, France

Wout Piessens  (6)
KU Leuven, Belgium

Clément Pit-Claudel  (24)
EPFL, Lausanne, Switzerland

Nikolaos Ploskas  (30)
University of Western Macedonia, Kozani,
Greece

Guillaume Povéda  (20, 46)
Airbus Central R&T, Toulouse, France

Cédric Pralet  (3)
DTIS, ONERA, Université de Toulouse, 31000,
Toulouse, France

Aaron Putterman  (8)
School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Christian Pérez  (45)
Universitat Politècnica de València, Spain

Claude-Guy Quimper  (13)
Université Laval, Québec, Canada

Masood Feyzbakhsh Rankooh  (47)
HIIT, Department of Computer Science,
University of Helsinki, Finland

Léo Robert  (32)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Gili Rosenberg  (48)
Amazon Advanced Solutions Lab, Seattle, WA,
98170, USA

Stéphanie Roussel  (40)
DTIS, ONERA, Université de Toulouse,
Toulouse, France

Jean-Charles Régis  (49)
Université Côte d'Azur, France

Miguel A. Salido  (45)
Universitat Politècnica de València, Spain

Tamara Borreguero Sanchidrian  (46)
Airbus DS, Getafe, Spain

Paul Sarnighausen-Cahn  (41)
University of Göttingen and CIDAS, Göttingen,
Germany

Eric Sawyer  (3)
CNES, Toulouse, France

Lakhdar Saïs  (27)
CRIL, CNRS - Université d'Artois, Lens, France

Torsten Schaub  (2)
Potsdam University, Germany

Pierre Schaus  (16, 23, 49)
ICTEAM, UCLouvain, Louvain-la-Neuve,
Belgium

Margaux Schmied  (49)
Université Côte d'Azur, France

Jendrik Seipp  (55)
Linköping University, Sweden

Ilankaikone Senthooan  (34)
Department of Data Science and AI, Faculty of
IT, Monash University, Clayton, Australia; ARC
Industrial Training and Transformation Centre
OPTIMA, Clayton, Australia

Konstantin Sidorov  (24, 50)
Delft University of Technology, The Netherlands

Christine Solnon  (11)
INSA Lyon, Inria, CITI, UR3720, 69621
Villeurbanne, France

Giuseppe Spallitta  (51)
Rice University, Houston, TX, USA

Helge Spieker  (4)
Simula Research Laboratory, Oslo, Norway

Kostas Stergiou  (30)
University of Western Macedonia, Kozani,
Greece

Peter J. Stuckey  (6, 35, 62)
Monash University, Melbourne, Australia;
OPTIMA ITTC, Melbourne, Australia

Stefan Szeider  (33, 52)
Algorithms and Complexity Group, TU Wien,
Austria

Pierre Talbot  (14)
University of Luxembourg, Esch-sur-Alzette,
Luxembourg

Yong Kiam Tan  (21, 43)
Nanyang Technological University, Singapore;
Institute for Infocomm Research (I²R), A*STAR,
Singapore

Fabio Tardivo  (53)
University of Connecticut, School of Computing,
Storrs, CT, USA

Javier Buil Tejero (46)
ALTEN GmbH, Manching, Germany

Tip ten Brink (24)
Delft University of Technology, The Netherlands

Tanguy Terrien (ID) (64)
LAAS-CNRS, Université de Toulouse, France

Dimos Tsouros (ID) (30)
University of Western Macedonia, Kozani,
Greece

Antoine Tuerlinckx (63)
Technord, Belgium

Eiji Ueda (36)
Future Technology Laboratories INC., Nagoya,
Japan

Daimy Van Caudenberg (ID) (54)
KU Leuven, Dept. Computer Science, Leuven,
Belgium

Pascal Van Hentenryck (ID) (31)
H. Milton Stewart School of Industrial and
Systems Engineering, Georgia Tech, Atlanta,
GA, USA

Willem-Jan van Hoeve (ID) (1, 53)
Carnegie Mellon University, Pittsburgh, PA,
USA

Damien Van Meerbeeck (ID) (55)
Linköping University, Sweden

Moshe Y. Vardi (ID) (51)
Rice University, Houston, TX, USA

Élise Vareilles (ID) (40)
CGI, IMT Mines Albi, Université de Toulouse,
Albi, France

Hélène Verhaeghe (ID) (63)
UCLouvain, Louvain-la-Neuve, Belgium

Petr Vilím (ID) (28)
ScheduleOpt, Novy Knin, Czech Republic

Anna Vitali (ID) (56)
Alma Mater Studiorum University of Bologna,
Italy

Anastasia Volkova (ID) (11)
Inria, INSA Lyon, CITI, UR3720, 69621
Villeurbanne, France

Stephen Walsh (22)
Dash Computer Solutions, Montreal, Canada

Rick S. H. Willemsen (ID) (57)
Singapore University of Technology and Design,
Engineering Systems and Design, Singapore

Felix Winter (ID) (58)
TU Wien, Vienna, Austria

Michael Wybrow (ID) (34)
Department of Human-Centred Computing,
Faculty of IT, Monash University, Clayton,
Australia

Hai Xia (ID) (33)
Algorithms and Complexity Group, TU Wien,
Austria

Tinghan Ye (ID) (31)
H. Milton Stewart School of Industrial and
Systems Engineering, Georgia Tech, Atlanta,
GA, USA

Akira Yoshioka (36)
Toyota Motor Corporation, Tokyo, Japan

Ziyang Yu (ID) (59)
School of Information Science and Technology,
Northeast Normal University, Changchun, China

Jialu Zhang (ID) (60)
MIS UR 4290, Université de Picardie Jules
Verne, Amiens, France

Xinda Zheng (ID) (61)
Department of Data Science and AI, Faculty of
IT, Monash University, Clayton, Australia;
OPTIMA ARC Training Centre, Melbourne,
Australia

Allen Z. Zhong (ID) (10)
Building 4.0 CRC, Caulfield East, Victoria,
Australia; Department of Data Science and
Artificial Intelligence, Monash University,
Clayton, Australia; ARC Training Centre in
Optimisation Technologies, Integrated
Methodologies, and Applications (OPTIMA),
Melbourne, Australia

Amal Zouaq (ID) (18)
Polytechnique Montréal, LAMA-WeST Lab,
Canada; MILA, Montréal, Canada

Šimon Zvára (28)
Faculty of Electrical Engineering, Czech
Technical University in Prague, Czech Republic

■ Senior Program Committee

- Arnaud Gotlieb (Simula Research Laboratory)
- Chris Beck (University of Toronto)
- Ciaran McCreesh (University of Glasgow)
- Christian Bessiere (University of Montpellier, CNRS)
- Christine Solmon (INSA Lyon / INRIA)
- Christophe Lecoutre (CRIL-CNRS, University of Artois)
- Claude-Guy Quimper (Université Laval)
- George Katsirelos (INRAE)
- Gilles Pesant (Polytechnique Montreal)
- Guido Tack (Monash University)
- Ian Gent (University of St Andrews)
- Justin Pearson (Uppsala University)
- Laurent Michel (University of Connecticut)
- Laurent Simon (University of Bordeaux, LaBRI)
- Louis-Martin Rousseau (Polytechnique Montreal)
- Nadjib Lazaar (University of Paris-Saclay)
- Roland H. C. Yap (National University of Singapore)
- Simon de Givry (INRAE)
- Tias Guns (KU Leuven)
- Willem-Jan van Hoeve (Carnegie Mellon University)
- Zeynep Kiziltan (University of Bologna)



■ Program Committee

- Agostino Dovier (University of Udine)
- Alejandro Arbelaez (Universitat Politècnica de València)
- Alexander Schiendorfer (Technische Hochschule Ingolstadt)
- Alexandre Goldsztejn (CNRS, University of Nantes, LS2N)
- Alexey Ignatiev (Monash University)
- Allen Z. Zhong (Monash University)
- Amel Hidouri (University Burgundy Franche-Comté)
- Anastasia Paparrizou (University of Montpellier, CNRS)
- Andre Augusto Cire (University of Toronto)
- Andrea Formisano (University of Udine)
- Andreas Podelski (University of Freiburg)
- Andrei A. Bulatov (Simon Fraser University)
- Armin Biere (University Freiburg)
- Armin Wolf (Fraunhofer FOKUS, Berlin)
- Arnaud Lallouet (RTE R&D)
- Arnaud Malapert (Université Côte d'Azur)
- Arnold Hien (University of Orleans, LIFO)
- Augustin Delecluse (KU Leuven)
- Barry O'Sullivan (University College Cork)
- Berthe Y. Choueiry (University of Nebraska-Lincoln)
- Bertrand Neveu (Gustave Eiffel University)
- Catherine Dubois (National School of Computer Science for Industry and Business)
- Cédric Pralet (ONERA)
- Cemalettin Ozturk (Munster Technological University and University College Cork)
- Charles Prud'homme (IMT Atlantique, LS2N)
- Christian Artigues (LAAS-CNRS)
- Chu-Min Li (University of Picardie Jules Verne)
- Clément Carbonnel (University of Montpellier, CNRS)
- Dimosthenis Tsouros (University of Western Macedonia)
- Edward Lam (Monash University)
- Elise Vareilles (IMT Mines Albi)
- Emir Demirović (Delft University of Technology)
- Emmanuel Hebrard (LAAS-CNRS)
- Enrico Pontelli (New Mexico State University)
- Feifei Ma (Institute of Software, Chinese Academy of Sciences)
- Felix Ulrich-Oltean (University of York)
- Felix Winter (TU Wien)
- Florentina Voboril (TU Wien)
- François Fages (INRIA)
- Frédéric Saubion (University of Angers)
- Gaël Reynal (Polytechnique Montreal)
- Gilles Simonin (IMT Atlantique, LS2N)
- Gilles Trombetti (University of Montpellier, LIRMM)
- Guillaume Derval (ULiège)
- Guillaume Escamocher (University College Cork)

32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).

Editor: Nicolas Beldiceanu



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Hadrien Cambazard (Grenoble INP)
- H  l  ne Verhaeghe (UCLouvain)
- Helge Spieker (Simula Research Laboratory)
- Helmut Simonis (University College Cork)
- Ian Miguel (University of St Andrews)
- Ilankaikone Senthoooran (Monash University)
- Jakob Nordstr  m (University of Copenhagen and Lund University)
- Jean-Charles R  gin (Universit   C  te d'Azur)
- Jeremias Berg (University of Helsinki)
- Jip J. Dekker (Monash University)
- Joan Espasa (University of St Andrews)
- Johannes Klaus Fichte (Link  ping University)
- John N. Hooker (Carnegie Mellon University)
- Kevin Leo (Monash University)
- Kostas Stergiou (University of Western Macedonia)
- Kyle E. C. Booth (Amazon)
- Lars Kotthoff (University of St Andrews)
- Laura Climent (University Polytechnic of Valencia)
- Laurent Perron (Google)
- Luis Quesada (University College Cork)
- Mar  a Andre  na Francisco Rodr  guez (Uppsala University)
- Marie-Jos   Huguet (LAAS-CNRS)
- Martin C. Cooper (University of Toulouse, IRIT)
- Mateu Villaret (University of Girona)
- Mats Carlsson (RISE)
- Michael Codish (Ben-Gurion University)
- Michael J. Maher (Reasoning Research Institute, Canberra)
- Michele Lombardi (University of Bologna)
- Mikael Zayenz Lagerkvist (Sambanova)
- Mikolas Janota (Czech Institute of Informatics, Robotics and Cybernetics)
- Mohamed Siala (INSA Toulouse and LAAS-CNRS)
- Nassim Belmecheri (Simula Research Laboratory)
- Neil Yorke-Smith (Delft University of Technology)
- Neng-Fa Zhou (Brooklyn College)
- Nguyen Dang (University of St Andrews)
- Nikolaj S. Bj  rner (Microsoft Research)
- Nysret Musliu (Vienna University of Technology)
- Olivier Lhomme (IBM)
-   zg  r Akg  n (University of St Andrews)
- Patrick Prosser (University of Glasgow)
- Paul Shaw (IBM)
- Peter Nightingale (University of York)
- Peter Stuckey (Monash University)
- Petr Vil  m (OptalCP)
- Phebe Vayanos (University of Southern California)
- Philippe Codognot (Sorbonne University / CNRS / University of Tokyo)
- Philippe Refalo (IBM)
- Philippe Vismara (Institut Agro Montpellier, MISTEA, LIRMM)

- Pierre Flener (Uppsala University)
- Pierre Schaus (UCLouvain)
- Quentin Cappart (UCLouvain & Polytechnique Montréal)
- Radu Marinescu (IBM)
- Roger Kameugne (Maroua University)
- Roman Barták (Charles University)
- Ruiwei Wang (National University of Singapore)
- Ruth Hoffmann (University of St Andrews)
- Samba Ndiaye (Claude Bernard University, LIRIS)
- Samir Loudni (IMT Atlantique, LS2N)
- Sebastian Ordyniak (University of Leeds)
- Serdar Kadioglu (Brown University & Fidelity Investments)
- Stefano Bistarelli (Perugia University)
- Stéphanie Roussel (ONERA)
- Steve Prestwich (University College Cork)
- Thi-Bich-Hanh Dao (University of Orleans, LIFO)
- Thierry Petit (Emotia)
- Vasco Manquinho (INESC-ID, Instituto Superior Técnico - Universidade de Lisboa)
- Vijay Ganesh (Georgia Institute of Technology)
- Vinasétan Ratheil Houndji (University of Abomey-Calavi)
- Xavier Lorca (IMT Mines Albi)
- Xiao Peng (LAAS-CNRS)

■ Additional Reviewers

- Aicha Boukhari (CEA)
- Amira Mastouri (IMT Atlantique, LS2N)
- Carlos March Moya (Delft University of Technology)
- Carlo Taticchi (Perugia University)
- Chiara Luchini (Perugia University)
- Christoph Jabs (University of Helsinki)
- Clotilde Brayé (Enovacom)
- Dimitri Watel (ENSIIE)
- Fabien Lehuédé (IMT Atlantique, LS2N)
- Frej Knutar (Uppsala University)
- Imane Bousdira (IRIT, Toulouse INP)
- Ivan Mercanti (Perugia University)
- Luca Maria Tutino (Perugia University)
- Marie Pelleau (Université Côte d'Azur)
- Mathijs de Weerd (Delft University of Technology)
- Ngan Ha Duong (Delft University of Technology)
- Sulian Le Bozec-Chiffolleau (IMT Atlantique, LS2N)
- Xueying Ou (Delft University of Technology)



A Scheduling-Based Approach to Railway Capacity Estimation for Industrial Freight Junctions

Luca Di Gaspero ✉ 

DPIA – Università degli Studi di Udine, Italy

Alessia Grosso ✉ 

DEAMS – Università degli Studi di Trieste, Italy

Giovanni Longo ✉ 

DEAMS – Università degli Studi di Trieste, Italy

Abstract

This work proposes a Constraint Programming (CP)-based methodology for estimating the maximum freight capacity of railway junctions serving industrial areas with multiple terminals, where minimum per-destination service frequencies must be guaranteed as hard constraints to preserve commercial competitiveness among terminal operators. The problem is formulated as a scheduling problem with optional interval variables: the solver decides which candidate services to execute and when, maximising completed round-trips within a 24-hour horizon while satisfying track-occupancy exclusivity, turnout conflicts, buffer-capacity limits, a mandatory maintenance window, and a fixed passenger timetable. The approach is implemented in OptalCP and evaluated on five scenarios covering the current and a planned upgraded infrastructure of the Port of Trieste, Italy. Satisfiability queries are answered in under 13 seconds; maximisation queries establish that the as-is network sustains at least 35 freight services per day against the 11 currently operated, rising to at least 38 after the planned upgrade, with both authority service targets confirmed feasible on their respective infrastructures.

2012 ACM Subject Classification Applied computing → Transportation; Theory of computation → Constraint and logic programming; Computing methodologies → Temporal reasoning

Keywords and phrases Freight Train Scheduling, Infrastructure Capacity Assessment, Optional Activities, OptalCP

Digital Object Identifier 10.4230/LIPIcs.CP.2026.19

Supplementary Material *Dataset (Dataset)*: <https://doi.org/10.5281/zenodo.20080975>

1 Introduction

Railway capacity estimation is a fundamental challenge in both infrastructure planning and operational management, as networks become increasingly congested and the pressure to accommodate heterogeneous traffic grows. Capacity, however, cannot be reduced to a single fixed value: it depends on infrastructure layout, signalling systems, traffic composition, and operational rules, making its estimation inherently complex [4, 6]. Methods developed to address this problem are generally classified into three categories: analytical approaches, simulation-based models, and optimisation techniques [1, 7].

Analytical methods such as UIC 406 and the CUI method estimate capacity by compressing the railway timetable to measure infrastructure utilisation [6]. While computationally inexpensive, these methods rely on restrictive assumptions, most notably traffic homogeneity, that limit their applicability to heterogeneous systems [1, 2]. Microscopic simulation overcomes these limitations by providing a detailed representation of infrastructure, signalling, and rolling stock dynamics, and is frequently used to verify the feasibility of proposed timetables or infrastructure modifications [1]. However, its high computational cost makes it ill-suited to the systematic generation and exploration of optimal scenarios. Optimisation



© Luca Di Gaspero, Alessia Grosso, and Giovanni Longo;
licensed under Creative Commons License CC-BY 4.0

32nd International Conference on Principles and Practice of Constraint Programming (CP 2026).

Editor: Nicolas Beldiceanu; Article No. 19; pp. 19:1–19:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

approaches, typically based on Mixed Integer Linear Programming (MILP), allow operational constraints to be modelled explicitly and support the identification of timetables that maximise infrastructure utilisation [3, 12, 19], though they are more commonly applied to train scheduling or rescheduling than to capacity estimation directly [7].

The limitations of all three approaches become particularly significant at complex railway nodes, including junctions, freight yards, and intermodal terminals, which concentrate the interactions between multiple routes and competing operational processes and represent the primary bottlenecks of railway networks. Analytical methods often fail to capture interaction effects between a node and its surrounding network [8]; simulation is costly to apply at node level for scenario exploration; and optimisation-based saturation approaches, while more suitable, remain limited in their ability to model differentiated operational requirements. Pascariu et al. [14] addressed this challenge by applying a MILP-based timetable saturation approach to the Novara freight node (Italy), iteratively adding train paths until infrastructure saturation was reached. Similarly, Russo et al. [17] proposed a MILP-based saturation model for evaluating the practical capacity of a transshipment yard, accounting for physical resources, operational time windows, and timetable constraints. Kianinejadoshah and Ricci [8] compared several analytical and simulation methods at the mixed passenger-freight node of the Port of Trieste, Italy, demonstrating that analytical methods systematically overestimate capacity by failing to capture network interaction effects, and that simulation remains necessary for in-depth analyses while analytical methods are appropriate only for strategic assessments.

In nodes serving industrial or logistics areas with multiple demand points, complexity increases further due to the need to guarantee minimum service levels to each destination, a requirement that existing capacity models do not explicitly address. Ensuring a minimum train frequency per terminal is a prerequisite for the commercial competitiveness of the node, yet no methodology has, to our knowledge, incorporated such constraints within a capacity estimation framework.

Constraint Programming (CP) offers a natural framework to address this gap. Its ability to represent complex combinatorial constraints flexibly, combined with the expressive power of optional interval variables, makes it possible to model the activation of each candidate train path as a direct scheduling decision. In the railway sector, CP has been applied to junction regulation by Rodriguez [16], to real-time traffic management via conditional interval variables by Marlière et al. [13], and to network-level capacity modelling for large freight networks by Harabor and Stuckey [5]. More recently, Schutt et al. [18] addressed in-station train dispatching using a CP-based approach, demonstrating the superiority of this approach over planning-based methods, and highlighted the flexibility of CP in accommodating multiple optimisation criteria. In complex logistics settings, Perez et al. [15] demonstrated the effectiveness of CP for scheduling train unloading operations at a bulk terminal. However, the application of CP to railway node capacity estimation with minimum per-destination service constraints remains unexplored.

This paper proposes a CP-based methodology for estimating the maximum capacity of a railway node serving an industrial area with multiple terminals, incorporating minimum train frequencies per destination as hard constraints. The model accounts for track-occupancy exclusivity, intersection conflicts, buffer-capacity limits, and two classes of fixed trains: timetabled passenger services enforced as hard temporal constraints, and pre-committed freight trains representing an established operational baseline. The output is twofold: a capacity estimate measured as the maximum number of schedulable trains, and a concrete operational schedule demonstrating practical feasibility, directly supporting what-if analyses for infrastructure and policy decisions. The methodology is validated on the railway network of the Port of Trieste, Italy.

The remainder of the paper is organised as follows. Section 2 describes the CP model, covering the infrastructure representation, decision variables, constraints, and objective function. Section 3 presents the experimental setup, the five evaluation scenarios, and the results obtained on the railway network of the Port of Trieste, Italy. Section 4 draws conclusions and outlines directions for future work.

2 Problem Model

In this work, the railway capacity estimation problem is formulated as a scheduling problem with optional activities and solved with OptaCP [20]. A **train** is defined as a single path traversal, either inbound (from an external endpoint to a terminal) or outbound (from a terminal back to an external endpoint); a **service** is defined as a complete freight round-trip consisting of an inbound train, a loading interval at the terminal, and an outbound train. Trains are assumed to operate on left-hand running, following the convention of the case study network; adapting the model to right-hand running would require only minor modifications. The model takes as input a description of the railway infrastructure (specified in a JSON configuration file) and a set of candidate freight train services, some of which are optional; the solver decides which optional services to execute and when, subject to operational constraints, while maximising the number of executed services.

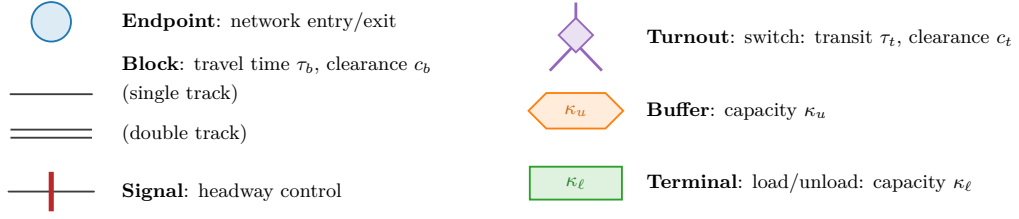
2.1 Infrastructure

The physical network is represented as an undirected graph $G = (N, E)$ whose nodes N are infrastructure objects and whose edges E are blocks (see Figure 1). Although the graph is undirected, meaning that each block can be traversed in either direction, the direction of traversal is determined by the train path π_θ and is therefore a property of the service, not of the infrastructure itself. The nodes N comprise five types of infrastructure objects; the edges E are blocks, described below together with the node types for completeness.

Endpoint ($\mathcal{EP}$). An entry or exit point of the network (e.g., a connection to the main line or to a railway station) that acts as a source or sink for services. Each endpoint is assigned a fixed, nominally small transit duration d^{ep} , introduced to enforce sequentiality of movements through the node.

Block ($\mathcal{B}$). A track segment connecting two adjacent infrastructure objects, traversable in either direction; the direction of traversal is determined by the train path π_θ and is therefore a property of the service, not of the infrastructure itself. The physical layout of the railway is reflected by the graph structure: each turnout node is incident to exactly three blocks, and signal nodes are always inserted between two blocks. Each block $b \in \mathcal{B}$ is characterised by its travel time $\tau_b = \lceil (d_b/v_b) \cdot 60 \rceil$ (minutes), where d_b is the block length in km and v_b the permitted speed in km/h. It is further characterised by the number of parallel tracks $k_b \in \{1, 2\}$, and an optional clearance time $c_b \geq 0$ that must elapse between two consecutive train passages in the same direction on a single-track segment. Long connections (distance $> 2 \cdot \ell_{\min}$, where $\ell_{\min}$ is a minimum block length, usually 1.35 km) are automatically subdivided into shorter segments separated by intermediate *signals*.

Signal ($\mathcal{S}$). A subdivision point inserted within a long block to enable finer-grained mutual exclusion. Signals carry no explicit dwell time and are transparent to the timetable representation.



■ **Figure 1** Node types (Endpoint, Signal, Turnout, Buffer, Terminal) and edge type (Block, shown as track segments) of the infrastructure graph G .

Turnout ($\mathcal{T}$). A railway switch connecting one *main* block to exactly two *diverging* blocks.

Each turnout $t \in \mathcal{T}$ has a transit time τ_t and a clearance time c_t . Two simultaneous train movements through the same turnout are *compatible* if and only if they traverse disjoint blocks and do not use the same physical rail (formally characterised by the `compatible_at_turnout` predicate in the implementation, see Section 2.4).

Buffer ($\mathcal{U}$). A set of holding tracks (sidings) adjacent to a terminal, with capacity κ_u (number of trains it can hold simultaneously).

Terminal ($\mathcal{L}$). A loading/unloading facility with capacity κ_ℓ (number of trains that can be served simultaneously).

2.2 Decision Variables

The core modelling primitive is the *interval variable* [9]: a variable x representing a contiguous time window $[s(x), e(x))$ of size $p(x) = e(x) - s(x)$, used here to model the occupation of a resource r by a train θ . We write $x_{\theta,r}$ for the occupation slot of train θ on resource r , decorated with a type superscript when needed. Optional interval variables additionally carry a *presence* logical variable: if absent ($\neg \text{pres}(x)$), all temporal constraints involving x are trivially satisfied [11]. This modelling choice is deliberate: by jointly deciding which services to schedule and when, the solver can exploit constraint propagation more effectively than if selection and timing were decoupled via binary variables, which would weaken propagation and potentially miss interactions between scheduling decisions and resource constraints. The structural primitives used throughout this section, namely `span`, `noOverlap`, `alternative`, sequence variables, and the cumulative `pulse( $x, h$ )` function, which contributes h units to a resource profile over $[s(x), e(x))$, are standard constructs of the interval-variable framework; we refer the reader to [10] for a comprehensive treatment. Interval variable domains are initially unbounded, with the exception of service span variables, which are constrained to lie within the generation horizon H' ; domain tightening is then performed automatically by the solver's constraint propagation.

Train paths and occupation slots

Each (candidate) train θ is associated with a simple *path* $\pi_\theta = (o_1, o_2, \dots, o_m)$ in G , where consecutive nodes are connected by a block. For every node o_i and every block $b_i = (o_i, o_{i+1})$ on the path, an *occupation slot* os_i is introduced, defined as a set of one or more interval variables that together capture the time during which train θ occupies that infrastructure object. For a path of n nodes, this yields $2n - 1$ occupation slots in total: n slots for the nodes and $n - 1$ slots for the connecting blocks. The following slot types are distinguished:

- **Block interval** $x_{\theta,b}^B$: size in $[\tau_b, \tau_b + \text{MAX_WAITING_TIME}]$. The size is thus flexible, allowing the solver to model brief halts or slow downs. MAX_WAITING_TIME is the maximum time a train may wait at a signal or before a turnout; it reflects operational practice, as extended waiting is not permitted since it would block the infrastructure for other movements. It is exposed as a configurable parameter in the JSON infrastructure file.
- **Turnout interval** $x_{\theta,t}^T$: fixed size τ_t , encoding the total time from the moment train θ begins traversing turnout t until the turnout has been reset to its neutral position and is available for the next movement.
- **Buffer stay interval** $x_{\theta,u}^U$: free size (the dwell time at a buffer is determined by the solver).
- **Endpoint interval** $x_{\theta,p}^{EP}$: fixed size d^{EP} (a sentinel marking network entry/exit).
- **Terminal marker** $x_{\theta,\ell}^L$: zero-size marker (used as a synchronisation anchor between inbound and outbound legs).

When a long connection is subdivided by signals, multiple block intervals are created and grouped into an *occupation slot span* $\hat{x}_{\theta,b}^B$ via the span constraint, which enforces that the span interval covers all sub-intervals exactly. Sub-intervals are further forced to be *consecutive* ($e(x_i) = s(x_{i+1})$) and, for optional trains, co-present ($\text{pres}(x_i) \iff \text{pres}(x_{i+1})$).

The overall occupation of train θ is captured by a global span

$$\text{span}(\hat{x}_\theta, \{\text{os}_i \mid o_i \in \pi_\theta\}),$$

which enforces $s(\hat{x}_\theta) = \min_i s(\text{os}_i)$ and $e(\hat{x}_\theta) = \max_i e(\text{os}_i)$.

For infrastructure objects that require mutual exclusion between trains (namely blocks, turnouts, and endpoints) a dedicated *sequence variable*

$$\sigma^r = \{x_{\theta,r}^{\text{type}(r)} \mid r \in \pi_\theta\}, r \in \mathcal{B} \cup \mathcal{T} \cup \mathcal{EP}$$

is defined over the occupation slot spans of all trains whose candidate path includes r . The constraint $\text{noOverlap}(\sigma^r)$ enforces that no two trains occupy r simultaneously, and that their passages are ordered consistently. Buffers and terminals are instead governed by capacity constraints (see Section 2.4) and do not require sequence variables.

All occupation slots are encapsulated in a **TrainPathModel** instance, which exposes: the global span variable $\hat{x}_\theta$, the individual occupation slot intervals indexed by infrastructure object, and the sequence variables σ^r .

In the following, to ease the notation, we will use $x_{\theta,r}$ or even x_θ when the resource is clear from the context.

Alternative trains

When the routing of a service between an endpoint p and a terminal ℓ admits multiple simple paths through G , the model constructs an *alternative train* to capture this routing choice efficiently. Rather than introducing one independent optional **TrainPathModel** per candidate route in full, the construction first identifies the longest *common prefix* π^{pre} and the longest *common suffix* π^{suf} shared by all candidate paths. A single, mandatory **TrainPathModel** is built for π^{pre} and for π^{suf} , so the occupation slots on the shared infrastructure segments are introduced only once and participate in mutual-exclusion constraints exactly once. Only the *diverging middle segments* $\{\pi_1^{\text{div}}, \dots, \pi_q^{\text{div}}\}$, i.e., the portions that differ across routes, give rise to separate optional interval variables $\hat{x}_{\theta,k}^{\text{div}}$, one per route variant. These are linked by the alternative constraint:

$$\text{alternative}(\hat{x}_\theta^{\text{alt}}, \{\hat{x}_{\theta,1}^{\text{div}}, \dots, \hat{x}_{\theta,q}^{\text{div}}\}),$$

which enforces that exactly one diverging variant is selected when the service is present. The overall span of the alternative train is then assembled as:

$$\text{span}(\hat{x}_\theta, [\hat{x}_\theta^{\text{pre}}, \hat{x}_\theta^{\text{alt}}, \hat{x}_\theta^{\text{suf}}]),$$

with the chaining constraints $e(\hat{x}_\theta^{\text{pre}}) = s(\hat{x}_\theta^{\text{alt}})$ and $e(\hat{x}_\theta^{\text{alt}}) = s(\hat{x}_\theta^{\text{suf}})$ ensuring temporal contiguity between the shared and diverging segments. This construction reduces the number of interval variables and mutual-exclusion constraints compared to a naive per-route enumeration, while preserving full modelling accuracy on the shared infrastructure.

The resulting structure is encapsulated in an **AlternativeTrainPathModel** instance, which exposes the same interface as **TrainPathModel** and can therefore be used transparently in all higher-level constructs.

2.3 Service Generation Strategy

Given the infrastructure graph G and a planning scenario, candidate services $s \in \mathcal{F}$ are generated automatically before the CP model is built. The procedure is driven by a designated set of *source/sink endpoints* $\mathcal{EP}^* \subseteq \mathcal{EP}$, which represent the external connection points relevant to freight traffic (e.g., industrial gates or connections to the main line). Note that not all endpoints need be available to freight: some may be reserved exclusively for passenger trains (e.g., a passengers station) and are therefore excluded from $\mathcal{EP}^*$.

For each pair (p, ℓ) with $p \in \mathcal{EP}^*$ and $\ell \in \mathcal{L}$, all *simple paths* from p to ℓ in G are enumerated. The enumeration remains tractable for the port railway topologies considered here, which are typically hierarchical or linear: trains flow from one or two external endpoints through a small number of buffers to reach the terminals, with limited branching. As a result, the number of simple paths between any endpoint-terminal pair is small. In more complex, reticulated topologies, the number of simple paths could grow combinatorially, requiring pruning based on operational feasibility criteria; we leave this extension to future work. Each such path becomes a candidate *inbound* train route. Because multiple simple paths may exist between the same endpoint and terminal, due to alternative turnout routing or parallel tracks, the collection of inbound routes is wrapped in an **AlternativeTrainPathModel**: the solver selects at most one concrete route per service execution, as enforced by the alternative constraint. This construction naturally captures routing flexibility without requiring the modeler to enumerate route choices manually.

For the *outbound* leg, a single route per service is generated by computing the *shortest path* (by travel time) from the terminal ℓ back to the originating endpoint p . The rationale is that, once a train has completed loading/unloading, it should vacate the terminal area as quickly as possible to minimise resource occupation; routing flexibility on the return leg is therefore less critical than on the inbound leg. In a real-world railway network an outbound path always exists by construction; nonetheless, the model performs a topological consistency check at initialisation to verify that every terminal is reachable from every freight endpoint, and raises an error if this precondition is violated.

The two legs are joined by a *service*, which pairs the inbound **AlternativeTrainPathModel** θ_s^{in} with the outbound **TrainPathModel** θ_s^{out} and introduces a **loading interval** x_s^{load} of minimum duration λ_s interposed between them:

$$e(\hat{x}_{\theta_s^{\text{in}}}) = s(x_s^{\text{load}}), \quad e(x_s^{\text{load}}) = s(\hat{x}_{\theta_s^{\text{out}}}).$$

The loading interval represents the time during which the train is stationary at terminal ℓ_s for loading or unloading operations; its duration is bounded below by λ_s and left free above, allowing the solver to delay departure as needed. A service can be declared *optional*; in

that case all constituent interval variables, including x_s^{load} , inherit the same presence status. The entire structure is encapsulated in a `ServiceModel` instance, which exposes the loading interval as the primary handle for capacity counting and terminal-capacity constraints.

The complete set of optional services, together with any mandatory (pre-scheduled) passenger or freight trains, is then handed to the `RailwaySchedulingModel`, which builds the CP model as described in Sections 2.2 and 2.4.

2.4 Constraints

Path contiguity

For any two consecutive occupation slots os_i and os_{i+1} on the path of train θ :

$$s(os_{i+1}) = e(os_i).$$

This enforces that a train moves continuously without unexplained gaps. In addition, for block-to-block transitions and for movements before a turnout, a maximum size constraint prevents excessive waiting:

$$p(os_i) \leq \tau_{os_i} + \text{MAX_WAITING_TIME}.$$

Block mutual exclusion

For every pair of trains (θ_i, θ_j) sharing an incompatible block b (i.e., $k_b = 1$ or both trains use the same track of a double-track block):

- If $c_b = 0$: $\text{noOverlap}(x_{\theta_i,b}, x_{\theta_j,b})$.
- If $c_b > 0$: a binary order variable $\delta_{b,ij} \in \{0, 1\}$ is introduced and the following conditional constraints are posted:

$$\begin{aligned} \text{pres}(x_{\theta_i,b}) \wedge \text{pres}(x_{\theta_j,b}) \wedge \delta_{b,ij} = 1 &\Rightarrow e(x_{\theta_i,b}) + c_b \leq s(x_{\theta_j,b}), \\ \text{pres}(x_{\theta_i,b}) \wedge \text{pres}(x_{\theta_j,b}) \wedge \delta_{b,ij} = 0 &\Rightarrow e(x_{\theta_j,b}) + c_b \leq s(x_{\theta_i,b}). \end{aligned}$$

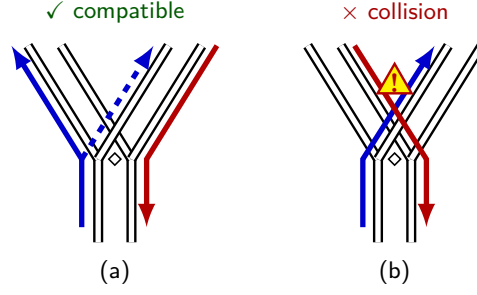
The clearance time models the *headway* required between trains using a single-track block in the same direction (e.g., the approach track to a terminal).

Turnout mutual exclusion

A turnout t connects one *main* block to one or more *diverging* blocks. When a train traverses t , it occupies a specific physical rail of the switch depending on its movement direction (main→diverging, diverging→main, or through). Two simultaneous movements through the same turnout are *compatible*, and therefore allowed to overlap in time, if and only if they use *disjoint* physical rails of the switch. This can only occur when the blocks adjacent to the turnout are *double-track*: in that case, one train may travel on track 0 of the incoming block while a second train simultaneously travels on track 1 of the outgoing block (or vice versa), without any physical conflict at the switch geometry, as illustrated in Figure 2(a). Conversely, if any of the blocks involved is single-track, the two trains would be forced onto the same rail and their occupations must be mutually exclusive (Figure 2(b)).

Formally, for every pair of trains (θ_i, θ_j) whose movements through a shared turnout t are determined to be incompatible by the `compatible_at_turnout` predicate, which checks movement directions and the track count of adjacent blocks, the following constraint is posted:

$$\text{noOverlap}(x_{\theta_i,t}, x_{\theta_j,t}).$$



■ **Figure 2** Compatible vs. incompatible movements at a double-track turnout: (a) each train takes a different branch or not sharing the stem; (b) opposing movements in the shared stem cause a head-on collision.

Pairs for which the predicate returns *compatible* are left unconstrained at the turnout level, though they may still be separated by block-level mutual exclusion constraints on the adjacent single-track segments.

Terminal capacity

Analogously, the number of trains simultaneously being loaded or unloaded at terminal ℓ is bounded:

$$\sum_{s: \ell_s = \ell} \text{pulse}(x_s^{\text{load}}, 1) \leq \kappa_\ell.$$

The $\text{pulse}(x, h)$ function contributes h units to a cumulative resource profile over the interval $[s(x), e(x))$; summing pulses and bounding the result enforces a capacity limit at every point in time [11].

Buffer capacity

The number of trains simultaneously held in buffer u is bounded by its capacity κ_u :

$$\sum_{\theta: u \in \pi_\theta} \text{pulse}(x_{\theta, u}, 1) \leq \kappa_u.$$

Interaction with passenger timetable

Scheduled passenger trains are modelled as *mandatory* (non-optional) `TrainPathModel` instances with fixed time windows. Their occupation intervals participate in the same block and turnout mutual-exclusion constraints as freight trains, so the solver automatically avoids conflicts with the pre-existing timetable.

Symmetry breaking

Since the model includes multiple optional trains sharing the same destination terminal, symmetric solutions arise naturally: any permutation of selected train paths *over a given terminal* yields a solution of equal objective value. Symmetry breaking is therefore applied independently to each group of optional trains assigned to the same terminal; trains serving different terminals are never grouped together, ensuring that no valid solution is excluded. Within each group, trains are indexed arbitrarily $\{\theta_1, \dots, \theta_n\}$ and two classes of symmetry-breaking constraints are imposed. First, presence variables are chained according to

$$\text{pres}(\hat{x}_{\theta_{i+1}}) \Rightarrow \text{pres}(\hat{x}_{\theta_i}), \quad i = 1, \dots, n-1$$

so that only the first k trains in the ordered sequence can be active in any feasible solution, effectively fixing the identity of the selected trains and removing permutation symmetry. Second, a temporal ordering constraint

$$s(\hat{x}_{\theta_i}) \leq s(\hat{x}_{\theta_{i+1}}), \quad i = 1, \dots, n-1,$$

is enforced among present trains, ensuring that the sequence reflects a canonical chronological order and preventing the solver from exploring solutions that differ only in the assignment of identical time slots to interchangeable interval variables. Together, these two constraints select a canonical ordering among structurally identical services targeting the same terminal, substantially reducing the search space without excluding any feasible solution. Correctness follows from the observation that any two feasible solutions differing only in the assignment of time slots to trains within the same group are related by a permutation of indices; the temporal ordering constraint selects exactly one canonical representative from each such equivalence class.

2.5 Objective

Let $H = 1,440$ min (24 hours) denote the *capacity horizon* and $H' = H + 360$ min (30 hours) the *generation horizon*. Services are generated with the end constraint $e(\hat{x}_s) \leq H'$. The extended horizon $H' = H + 360$ min serves two purposes. First, it captures trains entering the network within H whose outbound leg extends beyond it: truncating the horizon would prevent these services from being generated as candidates, underestimating the true capacity. Second, and more subtly, trains entering late in the day occupy shared infrastructure objects (blocks, turnouts, buffers) and may interfere with the outbound legs of other services; modeling these interactions explicitly, even beyond H , ensures that the schedule computed within H is not artificially optimistic.

Only services that are fully completed within the capacity horizon contribute to the objective:

$$\max \sum_{s \in \mathcal{F}^{\text{opt}}} \text{pres}(x_s^{\text{load}}) \cdot [e(\hat{x}_s) \leq H],$$

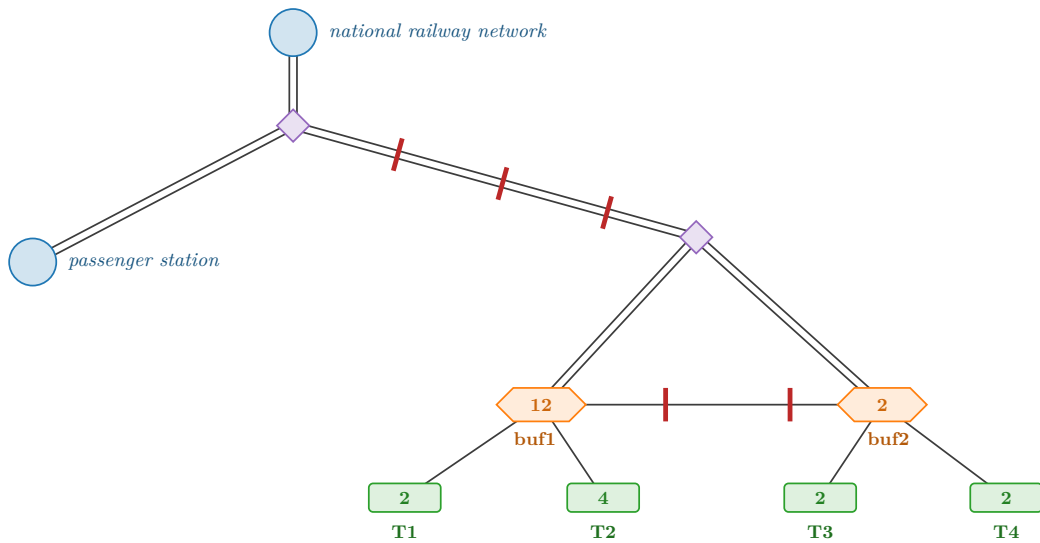
where $[\cdot]$ denotes the Iverson bracket. This ensures that the capacity estimate reflects only services whose entire round-trip falls within the operative day.

No-movement window

A mandatory *no-movement window* $[W_s, W_e]$ models the nightly maintenance closure of the external network. During this window, no inbound or outbound train movement is permitted on the shared infrastructure (blocks, turnouts, and the external endpoint); however, trains that have already reached a buffer or terminal may continue their loading or unloading activity uninterrupted. Formally, a fixed interval variable x^{nw} with $s(x^{\text{nw}}) = W_{\text{start}}$ and $e(x^{\text{nw}}) = W_{\text{end}}$ is introduced, and for every service $s \in \mathcal{F}$ the following non-overlap constraints are imposed:

$$\text{noOverlap}(\hat{x}_{\theta_s^{\text{in}}}, x^{\text{nw}}), \quad \text{noOverlap}(\hat{x}_{\theta_s^{\text{out}}}, x^{\text{nw}}).$$

These constraints ensure that the inbound and outbound legs of every service are entirely disjoint from the maintenance window, while leaving the loading interval x_s^{load} unconstrained with respect to it, since loading operations at a terminal are not affected by track closures on the external network.



■ **Figure 3** Railway Infrastructure of the Case Study.

The model is implemented in Python using the OptaCP API [20]; the output is a *concrete, feasible timetable* for the selected services, which can be inspected directly without the need for post-hoc simulation.

3 Case study

The proposed model is validated on a railway network serving the Port of Trieste, Italy, located along a core TEN-T corridor connecting Central Europe to the Mediterranean basin. The port is one of the principal freight gateways in the region, with rapidly growing traffic volumes and an increasing rail modal share; rail connectivity is a critical enabler for future port development, as the surrounding road network is already heavily congested.

Given the complexity of the full port infrastructure, which encompasses nine terminal areas operated by distinct companies, the study focuses on a representative portion of the network. From an infrastructure perspective, this sub-network branches off from the main national line into a dedicated freight sub-network comprising two operational buffers that regulate train flows between the main line and the terminal areas, and four terminals handling intermodal, rolling highway, and conventional freight services (see Figure 3).

The coexistence of frequent passenger traffic on the shared main line and heterogeneous freight movements creates a complex scheduling environment. The multi-terminal structure introduces a distinctive requirement: since terminals are managed by competing operators, a minimum number of rail services must be guaranteed to each, ensuring fair access and market competitiveness. This requirement, combined with the bottleneck effect of frequent passenger services at the junction, makes this case study both representative for the proposed methodology and practically relevant for infrastructure planning.

3.1 Experimental Setup

All experiments were conducted on a server equipped with two Intel® Xeon® Platinum 8368 processors (76 physical cores, 152 logical threads, 2.40 GHz base clock, 3.40 GHz max boost) and 512 GB of RAM. OptaCP was configured with `nbWorkers= 76` to exploit the available

parallelism. A time limit of 3600s was imposed on maximisation runs (Scenarios 3 and 5); satisfiability runs (Scenarios 1, 2, and 4) were assigned a time limit of 600s. Each scenario was solved three times independently to estimate solve-time variance, given the stochastic nature of the solver's search procedure; reported times are medians.

The planning horizon H spans 24 hours. A mandatory *unavailability window* from 00:00 to 04:00 prohibits freight movements on the external network during nightly maintenance; in practice, this window is binding only for services whose scheduling extends into the generation horizon H' beyond H . The loading time at each terminal is set to 240 minutes, representing the minimum time required for a complete train loading or unloading cycle. Passenger trains are included as mandatory, timetabled constraints in all scenarios.

Solver configuration

OptalCP supports the following search strategies, available in workers: Large Neighbourhood Search (LNS), Failure-Directed Search (FDS), Failure-Directed Search Dual (FDSDual).

LNS is a destroy-and-repair metaheuristic that iteratively fixes a subset of variables to their values in the current best solution, solves the resulting sub-problem to find an improvement, and restarts from the new incumbent; it finds good feasible solutions quickly but provides no optimality guarantee on its own.

FDS is a systematic search strategy based on the fail-first principle: it maintains a rating for each branching choice reflecting how often that choice has led to failures, and preferentially explores branches most likely to prune the search space; it is the primary driver of dual bound improvement and optimality proofs.

FDSDual is a variant of FDS focused exclusively on tightening the dual bound by iteratively proving infeasibility of successively better objective values, without searching for a complete solution.

Parallel cooperative search

When multiple workers are available, each worker runs its own independent search thread, and all workers share information about the best solution found so far: whenever a worker improves the incumbent, the new solution is immediately broadcast and the other workers restart from it. This cooperative scheme allows different search strategies to be combined across workers, typically pairing LNS workers (for rapid primal improvement) with FDS and FDSDual workers (for bound tightening), so that both solution quality and proof of optimality are pursued simultaneously.

3.2 Scenarios

Five scenarios are evaluated, differing in infrastructure configuration, minimum per-terminal service requirements, and solver objective. Table 1 summarises the instance parameters.

Scenario 1 – As-Is Baseline (Satisfiability)

The current as-is infrastructure is used, with minimum and maximum service counts set equal to the number of trains presently operated at each terminal (T1: 3, T2: 6, T3: 1, T4: 1, total 11). The solver is run in satisfiability mode: the question is whether the existing timetable is reproducible within the model, and whether the model correctly captures the current operational constraints. This scenario serves as a *correctness validation* of the CP model against known operational data.

19:12 A Scheduling-Based Approach to Railway Capacity Estimation

■ **Table 1** Scenario parameters: $N^{\max}$ are the upper bound on the allowed terminal services; κ_{b_1} , κ_{b_2} are the buffer capacities.

		T1	T2	T3	T4	κ_{b_1}	κ_{b_2}	Mode
S1	min = max	3	6	1	1	12	2	satisfiability
	$N^{\max}$	3	6	1	1			
S2	min = max	3	11	5	1	12	2	satisfiability
	$N^{\max}$	3	11	5	1			
S3	min	3	6	1	1	12	2	maximisation
	$N^{\max}$	12	24	6	6			
S4	min = max	6	16	12	2	20	10	satisfiability
	$N^{\max}$	6	16	12	2			
S5	min	3	6	3	1	20	10	maximisation
	$N^{\max}$	12	24	12	6			

Scenario 2 – As-Is, Authority Target (Satisfiability)

The authority has set a target of 20 freight services per day (T1: 3, T2: 11, T3: 5, T4: 1) on the as-is infrastructure. Minimum and maximum are set equal, so the solver must find a feasible schedule achieving exactly the target or declare infeasibility. This scenario *assesses whether the authority's short-term target is achievable without any infrastructure investment*.

Scenario 3 – As-Is, Maximisation

The same as-is infrastructure is retained. The minimum service floor equals the current traffic levels (T1: 3, T2: 6, T3: 1, T4: 1), while the upper bound on generated services is set well above any plausible infrastructure limit (T1: 12, T2: 24, T3: 6, T4: 6), ensuring that the model budget does not artificially cap the result. The solver maximises the total number of executed services within the capacity horizon. This scenario *quantifies the true residual capacity* of the existing infrastructure above the current operating point.

Scenario 4 – To-Be, Authority Target (Satisfiability)

A planned infrastructure upgrade is modelled: buffer b_1 capacity is increased from 12 to 20 and buffer b_2 from 2 to 10 trains. The minimum and maximum service counts are set equal to the authority's long-term target (T1: 6, T2: 16, T3: 12, T4: 2, total 36). The solver checks feasibility. This scenario *verifies whether the planned infrastructure upgrade is sufficient to meet the authority's future service objectives*.

Scenario 5 – To-Be, Maximisation

The same to-be infrastructure is used. The minimum service floor is set to a conservative demand estimate (T1: 3, T2: 6, T3: 3, T4: 1), and the upper bound on generated services is set well above any expected limit (T1: 12, T2: 24, T3: 12, T4: 6). The solver maximises the total number of executed services. This scenario *quantifies the potential capacity gain* brought by the infrastructure improvement.

3.3 Results

Table 2 reports the outcome of all five scenarios. All scenarios yield a feasible solution, confirming the practical tractability of the model across a range of infrastructure configurations and service requirements. In each scenario, the `min` column of Table 1 specifies the mandatory service floor: the minimum number of services per terminal that must be guaranteed to preserve commercial competitiveness among terminal operators. When `min` = `max`, all services are mandatory and the solver operates in pure satisfiability mode. The upper bound N^{max} on generated services per terminal is set to a sufficiently large value to ensure that the pool of candidates does not artificially cap the result.

Model size and solve time

The number of presolved interval variables grows from 1,097 in Scenario 1 to 2,473 in Scenario 5, reflecting the increasing number of candidate services generated. The number of constraints follows the same trend, rising from 18,061 to 54,644. Despite this growth, solve times for the three satisfiability scenarios (S1, S2, S4) are remarkably low: 0.52s, 5.22s, and 12.04s respectively, all well within the allotted 600s limit. This confirms that the CP model scales gracefully with network and service complexity for feasibility queries.

Scenario 1 – correctness validation

The solver reproduces the current operational timetable of 11 services feasibly and near-instantaneously, validating the model against known real-world data.

Scenario 2 – short-term authority target

The authority's short-term target of 20 freight services per day is confirmed feasible on the as-is infrastructure in just 5.22s. No physical investment is therefore required to meet the near-term demand target.

Scenario 3 – as-is capacity

The solver finds a best solution of $N^* = 35$ services (median over three independent runs) within the 3600s time limit, with a remaining gap of 23.4% against the best dual bound of 46. The true maximum capacity of the as-is infrastructure therefore lies in the interval [35, 46]. The current operating level of 11 services thus represents only a fraction of what the infrastructure can sustain, suggesting that significant capacity margin is available without any physical investment.

Scenario 4 – long-term authority target

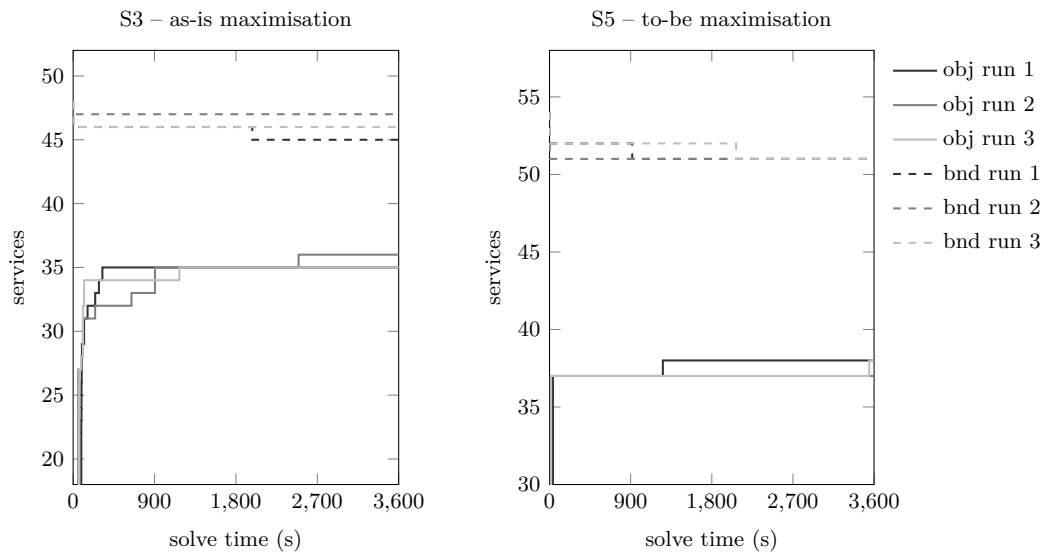
The authority's long-term target of 36 freight services per day is confirmed feasible on the to-be infrastructure in 12.04s, well within the time limit. The planned buffer upgrade is therefore sufficient to sustain the projected traffic growth.

Scenario 5 – to-be capacity

With the planned buffer expansion (κ_{b_1} raised from 12 to 20 and κ_{b_2} from 2 to 10), the best solution found is $N^* = 38$ services (median), with a remaining gap of 25.5% against a dual bound of 51. The estimated capacity range for the to-be infrastructure is [38, 51]. The upgrade thus yields a measurable increase over the as-is case, with the lower bound improving by at least 3 additional services.

■ **Table 2** Experimental results (medians over three runs). N^* : services within the 24-hour horizon; Bound/Gap: dual bound and relative gap at termination (maximisation only); Ivars/Ctrs: presolved interval variables and constraints; Mem: peak memory.

	Status	N^*	Bound	Gap (%)	Time (s)	Ivars	Ctrs	Mem (MB)
S1	Feasible	11	—	—	0.52	1,097	18,061	1,551
S2	Feasible	20	—	—	5.22	1,385	23,241	2,042
S3	Feasible	35	46	23.4	3600	2,281	48,471	5,467
S4	Feasible	36	—	—	12.04	1,897	35,017	2,653
S5	Feasible	38	51	25.5	3600	2,473	54,644	4,794



■ **Figure 4** Convergence of best objective (solid) and dual bound (dashed) over time for Scenarios 3 and 5, across three independent runs.

Convergence analysis

Figure 4 shows the evolution of the best objective value and dual bound over time for Scenarios 3 and 5, across all three independent runs.

In Scenario 3, the solver locates an initial solution of 22–24 services between 56 and 90 s, then steadily improves through LNS to reach 35–36 services by the end of the horizon. The dual bound descends slowly: it drops from 48 to 47 within the first two seconds, but then requires over half an hour to reach 45–46, leaving a gap that cannot be closed within 3600 s.

This asymmetry between primal improvement and dual bound tightening is characteristic of problems where LNS quickly saturates the timetable but global optimality requires reasoning about the entire combinatorial structure. In Scenario 5, the behaviour is markedly different: the first solution (37 services) is found much faster (14–38 s), reflecting the additional buffer capacity that relaxes the most constrained resource. However, improving beyond 38 services proves difficult, and the dual bound stagnates at 51 for most of the solve, leaving a gap of approximately 25% at termination. The inter-run variability is moderate for Scenario 3 (best solutions range from 35 to 36) and low for Scenario 5 (best solutions range from 37 to 38), consistent with the stochastic nature of parallel LNS.

4 Conclusions

This paper proposed a CP-based methodology for estimating the maximum freight capacity of a railway node serving an industrial area with multiple terminals, incorporating minimum per-destination service frequencies as hard constraints alongside track-occupancy exclusivity, intersection conflicts, buffer-capacity limits, and a mandatory passenger timetable. The model is implemented using optional interval variables and sequence variables in OptalCP, and produces as output both a capacity estimate and a concrete, verifiable timetable that can be inspected directly without any post-hoc simulation step.

Experimental evaluation on the railway network of the Port of Trieste, Italy yielded several practically relevant findings. For the satisfaction scenarios (S1, S2, S4), solve times are uniformly low (under 13 s), demonstrating that feasibility queries can be answered in near real-time, a property directly useful for interactive infrastructure planning tools. For the maximisation scenarios (S3 and S5), the upper bound on generated services was set well above any plausible infrastructure limit to prevent the model budget from artificially capping the result; the solver makes strong primal progress via LNS, establishing that the as-is infrastructure sustains at least 35 freight services per day against the 11 currently operated, rising to at least 38 after the planned buffer expansion. A dual gap of approximately 23–25% remains at the 3600 s time limit, indicating that closing the gap for this class of problems remains an open challenge that calls for tighter problem-specific bounds. The authority's short-term target of 20 services (S2) is confirmed feasible on the as-is infrastructure without any physical modification, and the long-term target of 36 services (S4) is confirmed feasible on the upgraded infrastructure, directly informing investment prioritisation and policy decisions.

A direct comparison with MILP-based approaches and microscopic simulation is not pursued in this work. To the best of our knowledge, no MILP formulation exists in the literature that operates at the same level of operational detail as the proposed model; existing formulations require simplifications, such as traffic homogeneity or relaxed infrastructure constraints, that do not guarantee equivalent solution quality. Simulation-based approaches, on the other hand, require a pre-specified timetable as input and are designed to verify feasibility rather than to optimise capacity, and are therefore not directly comparable to an optimisation-based capacity estimation framework.

Several directions for future work emerge from these results. Extending the model to the full port infrastructure, including the additional terminals and buffer areas planned in the to-be layout, would refine the capacity estimates and broaden the scope of the what-if analysis to a larger and more representative network. On the solving side, tighter dual bounds through problem-specific reasoning and additional symmetry-breaking constraints could reduce the optimality gap within the available time budget. Finally, the framework opens a natural path towards systematic sensitivity analysis, quantifying how capacity responds to changes in passenger traffic intensity, terminal loading times, or buffer configurations, thereby directly supporting infrastructure investment decisions.

References

- 1 Montserrat Abril, Federico Barber, Laura Paola Ingolotti, Miguel A. Salido, Pilar Tormos, and Antonio Luis Lova. An assessment of railway capacity. *Transp. Res. Part E: Logist. Trans. Rev.*, 44:774–806, 2008. doi:10.1016/j.tre.2007.04.001.
- 2 Nikola Bešinović and Rob M P Goverde. Stable and robust train routing in station areas with balanced infrastructure capacity occupation. *Public Transp.*, 11(2):211–236, August 2019. doi:10.1007/s12469-019-00202-3.

- 3 Valentina Cacchiani, Dennis Huisman, Martin Kidd, Leo Kroon, Paolo Toth, Lucas Veelenturf, and Joris Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Trans. Res. Part B: Methodol.*, 63:15–37, 2014. doi:10.1016/j.trb.2014.01.009.
- 4 Ingo Hansen and Joern Pachl, editors. *Railway timetabling and operations: Analysis, modelling, optimisation, simulation, performance, evaluation*. Eurail press, 2014.
- 5 Daniel Harabor and Peter J. Stuckey. Rail capacity modelling with constraint programming. In Claude-Guy Quimper, editor, *Integration of AI and OR Techniques in Constraint Programming*, volume 9676, pages 170–186. Springer International Publishing, 2016. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-319-33954-2_13.
- 6 International Union of Railways. *UIC Code 406: Capacity*. Paris, 1st edition, 2004.
- 7 Melody Khadem Sameni and Arash Moradi. Railway capacity: A review of analysis methods. *Journal of Rail Transport Planning & Management*, 24:100357, 2022. doi:10.1016/j.jrtpm.2022.100357.
- 8 Atieh Kianinejadshah and Stefano Ricci. Capacity assessment in freight-passengers complex railway nodes: Trieste case study. *Infrastructures*, 7(8):106, 2022. doi:10.3390/infrastructures7080106.
- 9 Philippe Laborie and Jérôme Rogerie. Reasoning with conditional time-intervals. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, pages 555–560. AAAI Press, 2008. URL: <https://aaai.org/papers/flairs-2008-126/>.
- 10 Philippe Laborie, Jérôme Rogerie, Paul Shaw, and Vincent Furnon. IBM ILOG CP optimizer for scheduling. *Constraints*, 23(2):210–250, 2018. doi:10.1007/s10601-018-9281-x.
- 11 Philippe Laborie, Jérôme Rogerie, Paul Shaw, Petr Vilím, and Ferenc Katai. Interval-based language for modeling scheduling problems: An extension to constraint programming. In Josef Kallrath, editor, *Algebraic Modeling Systems: Modeling and Solving Real World Optimization Problems*, volume 104 of *Applied Optimization*, pages 111–143. Springer, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-23592-4_6.
- 12 Zhengwen Liao and Ce Mu. Assessing the compatibility of railway station layouts and mixed heterogeneous traffic patterns by optimization-based capacity estimation. *Mathematics*, 11(17):3727, 2023. doi:10.3390/math11173727.
- 13 Grégory Marlière, Sonia Sobieraj Richard, Paola Pellegrini, and Joaquin Rodriguez. A conditional time-intervals formulation of the real-time railway traffic management problem. *Control Engineering Practice*, 133:105430, 2023. doi:10.1016/j.conengprac.2022.105430.
- 14 Bianca Pascariu, Nicola Coviello, and Andrea D’Ariano. Railway freight node capacity evaluation: a timetable-saturation approach and its application to the novara freight terminal. *Transportation Research Procedia*, 52:155–162, 2021. doi:10.1016/j.trpro.2021.01.017.
- 15 Guillaume Perez, Gaël Glorian, Wijnand Suijlen, and Arnaud Lallouet. A constraint programming model for scheduling the unloading of trains in ports. In *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 693–699. IEEE, 2023. doi:10.1109/ICTAI59109.2023.00108.
- 16 Joaquín Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B: Methodological*, 41(2):231–245, 2007. doi:10.1016/j.trb.2006.02.006.
- 17 Anna Russo Russo, Roberto Mancini, Gianpaolo Oriolo, Veronica Piccialli, and Davide Ussai. Optimization for evaluating the practical capacity of a transshipment yard. doi:10.48550/arXiv.2410.02968.
- 18 Andreas Schutt, Matteo Cardellini, Jip J. Dekker, Daniel Harabor, Marco Maratea, and Mauro Vallati. Constraint-Based In-Station Train Dispatching. In *31st International Conference on Principles and Practice of Constraint Programming (CP 2025)*, volume 340 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CP.2025.33.

- 19 Christopher Szymula, Nikola Bešinović, and Karl Nachtigall. Demand-based capacity assessment using mixed integer programming. *Journal of Rail Transport Planning & Management*, 33:100502, 2025. doi:10.1016/j.jrtpm.2024.100502.
- 20 Vilim, Petr. OptalCP. Accessed: 2026-03-10. URL: <http://dev.vilim.eu/>.