Report from Dagstuhl Seminar 24441

Machine Learning Augmented Algorithms for Combinatorial Optimization Problems

Deepak Ajwani*[1], Bistra Dilkina*[2], Tias Guns*[3], and Ulrich Carsten Meyer*[4]

1 **University College Dublin, IE.** deepak.ajwani@ucd.ie
2 **USC – Los Angeles, US.** dilkina@usc.edu
3 **KU Leuven, BE.** tias.guns@kuleuven.be
4 **Goethe University – Frankfurt am Main, DE.** umeyer@ae.cs.uni-frankfurt.de

Abstract

Combinatorial optimization problems are pervasive across critical domains, including business analytics, engineering, supply chain management, transportation, and bioinformatics. Many of these problems are NP-hard, posing significant challenges for even moderately sized instances. Moreover, even when polynomial-time algorithms exist, their practical implementation can be computationally expensive for real-world applications.

This has driven decades of research across diverse fields, encompassing exact and approximation algorithms, parameterized algorithms, algorithm engineering, operations research, optimization solvers (such as mixed-integer linear programming and constraint programming solvers), and nature-inspired metaheuristics.

Recently, there has been a surge in research exploring the synergistic integration of machine learning techniques with algorithmic insights and optimization solvers to enhance the scalability of solving combinatorial optimization problems. However, research efforts in this area are currently fragmented across several distinct communities, including those focused on "Learning to scale optimization solvers," "Algorithm Engineering," "Algorithms with predictions," and "Decision-focused learning."

This seminar brought together researchers from these diverse communities, fostering a dialogue on effectively combining algorithm engineering techniques, optimization solvers, and machine learning to address these challenging problems. The seminar facilitated the development of a shared vocabulary, clarifying similarities and distinctions between concepts across different research areas. Furthermore, significant progress was made in identifying key research directions for the future advancement of this field. We anticipate that these outcomes will serve as a valuable roadmap for advancing this exciting research area.

Seminar October 27–31, 2024 – https://www.dagstuhl.de/24441

2012 ACM Subject Classification Computing methodologies → Discrete space search; Theory of computation → Design and analysis of algorithms; Theory of computation → Discrete optimization

Keywords and phrases Algorithm Engineering, Combinatorial Optimization, Constraint Solvers, Machine Learning

Digital Object Identifier 10.4230/DagRep.14.10.76

* Editor / Organizer

Machine Learning Augmented Algorithms for Combinatorial Optimization Problems, *Dagstuhl Reports*, Vol. 14, Issue 10, pp. 76–100
Editors: Deepak Ajwani, Bistra Dilkina, Tias Guns, and Ulrich Carsten Meyer

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

1 Executive Summary

Deepak Ajwani (University College Dublin, IE)
Bistra Dilkina (USC – Los Angeles, US)
Tias Guns (KU Leuven, BE)
Ulrich Carsten Meyer (Goethe University – Frankfurt am Main, DE)

Combinatorial optimization problems are pervasive across critical domains, driving decades of research across diverse fields, including exact and approximation algorithms, algorithm engineering, operations research, and optimization solvers (such as mixed-integer linear programming and constraint programming solvers). In recent years, there has been a surge in research at the intersection between machine learning, optimization solvers, and algorithm engineering. While significant progress has been made, research efforts in this area remain fragmented across several distinct communities:

- "Learning to scale optimization solvers": This community focuses on leveraging learning techniques to improve the performance of solvers, such as mixed-integer linear programming, constraint programming, and satisfiability solvers, often using generic, problem-agnostic features. This includes optimizing solver components and exploring "end-to-end" approaches that replace traditional solvers entirely.
- "Algorithm Engineering": This community utilizes learning techniques to select the most effective components within algorithms, fill in the underspecified parts of the algorithms, and develop novel heuristics, often leveraging problem-specific features and algorithmic insights.
- "Algorithms with predictions": This community develops algorithms that leverage predictions to improve performance, often treating the prediction system as a black box. These algorithms aim for good performance with accurate predictions while maintaining bounded worst-case performance with inaccurate predictions.
- "Decision-focused learning": This research area focuses on systems where optimization decisions are made based on machine learning models, necessitating an integrated approach to avoid suboptimal solutions.

This seminar brought together researchers from these diverse communities, fostering a dialogue on effectively combining algorithm engineering techniques, optimization solvers, and machine learning. The seminar facilitated the development of a shared vocabulary, clarifying similarities and distinctions between concepts across different research areas. Notably, many concepts with different names (e.g., "learning to prune," "predict then search", "finding kernel", "finding a backbone", "learning to branch", "learning constraints") were found to represent similar ideas. Key differences often stem from the source of features, with "learning to prune" typically relying on problem-specific, algorithm-based features, while "learning to branch" often utilizes problem-agnostic features derived from the solver run. This shared understanding fostered significant synergy and cross-fertilization across research areas.

The seminar featured discussions on recent advancements, at the intersection of machine learning, algorithm engineering, and combinatorial optimization. For instance, the tutorial on "ML-Guided Solvers for MIP" focused on recent advances in using contrastive learning for MILP solvers, the talk on "Neural Combinatorial Optimization – One Model to Solve Them All?" described recent advances in foundation models for sequencing combinatorial optimization and the talk on "Graph Learning for Planning" described very impressive results

on the usage of learning techniques for planning problems. In addition, many participant talks such as "Progress and Open Questions on DFL and Constrained NNs" and "Machine Learning for Edge Contractions in Multilevel Graph Partitinioning" highlighted interesting open problems in the area.

Furthermore, the seminar identified key research directions, including:

- Theoretical frameworks for machine learning augmented combinatorial optimization algorithms
- Learning to fill the underspecified parts of the algorithms
- Reinforcement learning for combinatorial optimization
- Contextual Stochastic Optimization and DFL
- Fast and/or interpretable machine learning techniques for combinatorial problems
- Scalability and Datasets for Decision Focused Learning
- Finding worst-case instances for algorithms using machine learning techniques

These directions are expected to serve as a valuable roadmap for advancing this exciting research area.

Seminar Overview

The seminar commenced with a vibrant round of introductions, allowing participants to share their research interests and backgrounds. The slides for this round of introductions were collected in advance. To foster interdisciplinary connections, participant order was randomized, encouraging interaction across the diverse research communities represented.

Following this, four insightful tutorials were presented, each from a different research community. These tutorials served to familiarize all participants with the unique vocabulary, key techniques, and recent advancements within each area, providing a common ground for subsequent discussions.

A series of engaging participant talks followed, showcasing cutting-edge research from across the field. The talk schedule was carefully randomized to maximize interaction and foster cross-pollination of ideas among the diverse research communities. This dynamic approach encouraged participants to step outside their immediate research areas and engage with novel perspectives.

Recognizing the importance of collective vision, participants were invited to submit key research directions and open challenges within the field. These valuable contributions were then clustered, enabling the identification of key research themes and areas of significant opportunity.

To delve deeper into these themes, three parallel discussion sessions were held. Each session featured three distinct discussion groups, allowing participants to select the group most aligned with their interests. This dynamic group composition, which changed daily, encouraged participants to engage with a variety of perspectives and fostered a truly inter-community exchange of ideas. It was particularly encouraging to observe that many discussion groups comprised members from all four research communities, demonstrating the successful integration of diverse perspectives throughout the seminar.

2 Table of Contents

3 Tutorials

3.1 Decision-focused learning: Foundations, State of the Art, Benchmark and Future Opportunities

Tias Guns (KU Leuven, BE)

Increasingly, combinatorial optimisation problems follow a predict + optimize paradigm, where part of the parameters (costs, volumes, capacities) are predicted from data, and those predictions are fed into a downstream combinatorial optimisation problem. How to best train these predictive models? Decision-focused learning (DFL) is an emerging paradigm in machine learning which trains a model to optimize decisions, integrating prediction and optimization in an end-to-end system. This paradigm holds the promise to revolutionize decision-making in many real-world applications which operate under uncertainty, where the estimation of unknown parameters within these decision models often becomes a substantial roadblock to high-quality solutions. This talk presents a comprehensive review of DFL. It provides an in-depth analysis of the various techniques devised to integrate machine learning and optimization models, introduces a taxonomy of DFL methods distinguished by their unique characteristics, and conducts an extensive empirical evaluation of these methods proposing suitable benchmark dataset and tasks for DFL. Finally, we'll provide valuable insights into current and potential future avenues in DFL research.

3.2 Approximation Algorithms with Predictions

Adam Polak (Bocconi University – Milan, IT)

Joint work of Adam Polak, Antonios Antoniadis, Marek Eliáš, Moritz Venzin

I will talk about utilizing predictions to improve over approximation guarantees of classic algorithms, without increasing the running time. I will cover prior results going in that direction, and then I will talk about our recent work in which we propose a generic method for a wide class of optimization problems that ask to select a feasible subset of input items of minimal (or maximal) total weight. This gives simple (near-)linear-time algorithms for, e.g., Vertex Cover, Steiner Tree, Min-Weight Perfect Matching, Knapsack, and Clique. Our algorithms produce an optimal solution when provided with perfect predictions and their approximation ratio smoothly degrades with increasing prediction error. With small enough prediction error we achieve approximation guarantees that are beyond the reach without predictions in given time bounds, as exemplified by the NP-hardness and APX-hardness of many of the above problems. Although we show our approach to be optimal for this class of problems as a whole, there is a potential for exploiting specific structural properties of individual problems to obtain improved bounds; we demonstrate this on the Steiner Tree problem. I will conclude with an empirical evaluation of our approach. This is based on joint work with Antonios Antoniadis, Marek Eliáš, and Moritz Venzin.

3.3 ML-Guided Solvers for MIP

Bistra Dilkina (USC – Los Angeles, US)

In this tutorial talk, I present recent advances on (i) augmenting discrete optimization algorithms with learning components and (ii) learning methods that incorporate the combinatorial decisions they inform. While the first set of techniques focus on infusing discrete optimization with machine learning, the second set of techniques focus on infusing machine learning with constrained decision making. The talk is based on my recent publications [1, 2, 3].

References

1 Taoan Huang, Aaron M. Ferber, Yuandong Tian, Bistra Dilkina and Benoit Steiner. Searching Large Neighborhoods for Integer Linear Programs with Contrastive Learning abs/2302.01578 2023.
2 Taoan Huang, Aaron M. Ferber, Arman Zharmagambetov, Yuandong Tian and Bistra Dilkina. Contrastive Predict-and-Search for Mixed Integer Linear Programs. Forty-first International Conference on Machine Learning, ICML 2024,
3 Junyang Cai, Taoan Huang and Bistra Dilkina. Learning Backdoors for Mixed Integer Linear Programs with Contrastive Learning 27th European Conference on Artificial Intelligence ECAI, 2024,

3.4 Learning problems in Algorithm design and engineering

Deepak Ajwani (University College Dublin, IE)

Joint work of Dena Tayebi, Saurabh Ray, Deepak Ajwani
Main reference Dena Tayebi, Saurabh Ray, Deepak Ajwani: "Learning to Prune Instances of k-median and Related Problems", in Proc. of the Symposium on Algorithm Engineering and Experiments, ALENEX 2022, Alexandria, VA, USA, January 9-10, 2022, pp. 184–194, SIAM, 2022.
URL http://dx.doi.org/10.1137/1.9781611977042.15

In this talk, we explored the learning problems arising in algorithm design, engineering and analysis. We will discuss (i) ways in which the insights from the algorithm engineering literature can be used in problem-specific learning solutions, (ii) how learning techniques can be applied to fill the underspecified parts in algorithms (e.g., ordering in Bellman-Ford algorithm), (iii) how machine learning can help in identifying and estimating the various parameters used in the design of parameterised algorithms, (iv) how learning techniques can be used for the design of interpretable heuristics and (v) if learning techniques can be used for finding combinatorial structures and worst-case inputs for algorithm analysis.

References

1 Dena Tayebi, Saurabh Ray and Deepak Ajwani. Learning to Prune Instances of k-median and Related Problems Proceedings of the Symposium on Algorithm Engineering and Experiments, ALENEX 2022,

4 Overview of Talks

4.1 On LLM prompt optimization and amortization

Brandon Amos (Meta – New York, US)

Prompting LLMs is a challenging art where different ways of expressing the same idea can lead to drastically different responses. Not prompting in the right way gives suboptimal performance and has led to the budding space of prompt engineering and optimization techniques. A standard example here is that appending the string "let's think step by step" to the prompt may significantly improve the quality on few-shot classification tasks. In this session, we'll first cover how prompt optimization can be expressed as a combinatorial optimization problem (over the token sequence space) and overview the standard methods here. Beyond this, prompt optimization problems are often not solved a single time in isolation, but are repeatedly solved for every new task or piece of information we want to extract from the language model. So, we'll conclude with an overview of learned optimization, or amortization, to share the information between the repeatedly-solved optimization problems.

4.2 Enhancing constraint programming with machine learning

Quentin Cappart (Polytechnique Montréal, CA)

Joint work of Quentin Cappart, Louis-Martin Rousseau, Swann Bessa, Darius Dabert, Max Bourgeat
Main reference Swann Bessa, Darius Dabert, Max Bourgeat, Louis-Martin Rousseau, Quentin Cappart: "Learning Valid Dual Bounds in Constraint Programming: Boosted Lagrangian Decomposition with Self-Supervised Learning", CoRR, Vol. abs/2408.12695, 2024.
URL http://dx.doi.org/10.48550/ARXIV.2408.12695

Constraint programming (CP) is a well-established method for tackling combinatorial problems. Traditionally, CP has focused on solving isolated problem instances, often overlooking the fact that these instances frequently originate from related data distributions. In recent years, there has been a growing interest in leveraging machine learning, particularly neural networks, to enhance CP solvers by utilizing historical data. Despite this interest, it remains unclear how to effectively integrate learning into CP engines to boost overall performance. In this presentation, I will share my experience in tackling this challenge, from my initial attempts to my current research directions.

4.3 Learning to solve real-world puzzles: from Sudoku to protein design

Marianne Defresne (KU Leuven, BE)

Real-life decision making often involves reasoning on ill-defined problems, where exact constraints or parameters (such as costs) are unknown. The goal of Decision-Focused Learning (DFL) is to automatize the definition of problem parameters by using Deep

Learning to extract knowledge out of the environment. The main challenge here is to combine discrete optimization for reasoning with continuous optimization for learning. I will present one method to learn discrete graphical models, the reasoning framework we chose. We first assessed it on learning the rules of Sudoku, a popular benchmark for DFL methods. We then apply it to Computational Protein Design, a real-world problem that can be described and tackled with graphical models. We deep learned the interactions within existing proteins to better guide the design towards new proteins of interest.

4.4 The differentiable feasibility pump

Alexandre Forel (Polytechnique Montréal, CA)

Joint work of Alexandre Forel, Matteo Cacciola, Andrea Lodi, Antonio Frangioni

The feasibility pump algorithm is a widely used heuristic to find feasible primal solutions to mixed-integer linear problems. Many extensions of the algorithm have been proposed. Yet, its core algorithm remains centered around two key steps: solving the linear relaxation of the original problem to obtain a solution that respects the constraints, and rounding it to obtain an integer solution. This paper shows that the feasibility pump and many of its follow-ups can be seen as gradient-descent algorithms with specific parameters. A central aspect of this reinterpretation is observing that the traditional algorithm differentiates the solution of the linear relaxation with respect to its cost. This reinterpretation opens many opportunities for improving the performance of the original algorithm. We study how to modify the gradient-update step as well as extending its loss function. We perform extensive experiments on instances from the MIPLIB library and show that these modifications can substantially reduce the number of iterations needed to find a primal solution.

4.5 Screening for Data Reductions in Combinatorial Optimization: A Graph Neural Network Approach

Ernestine Großmann (Universität Heidelberg, DE)

Joint work of Ernestine Großmann, Kenneth Langedal, Christian Schulz

Combinatorial optimization problems play a pivotal role in various domains, challenging researchers to find optimal solutions within large solution spaces. One key strategy to tackle the complexity of these problems is data reduction, where instances are simplified to facilitate more efficient algorithms. Even though there are numerous reduction rules available, they are often not used in applications since the time needed to test them for the whole graph exhaustively becomes infeasible. With this, a new problem in the form of early reduction screening emerges. In this talk, we tackle this problem in the context of maximum weight independent sets using graph neural networks. Before checking if an expensive data reduction rule is applicable, we consult a GNN oracle to decide if the probability of successfully reducing the graph is sufficiently large. With this approach, we can use even expensive reduction rules successfully in feasible time. We introduce a new supervised learning dataset and provide first results using established graph neural network architectures.

4.6 Efficiently learning instance-optimal linear system solvers

Mikhail Khodak (Princeton University, US)

Augmenting classical algorithms with learned predictions or configurations has found many successful applications in energy management, database systems, scientific computing, and beyond. At the same time it has been theoretically challenging to go beyond the computationally inefficient learning of static predictors and configurations. We study the problem of sequentially solving linear systems, a fundamental problem in numerical computing, and introduce an algorithm that efficiently learns to do instance-optimal linear solver configuration while using only the number of iterations as feedback. Our approach points towards new directions for analyzing iterative methods, designing surrogate objectives for optimizing algorithmic costs, and incorporating tools from online learning into algorithm design.

4.7 The Expressive Power of Graph Neural Networks

Sandra Kiefer (University of Oxford, GB)

Graph Neural Networks (GNNs) are a machine learning architecture to learn functions on graphs. For example, since problem instances for combinatorial optimisation tasks are often modelled as graphs, GNNs have recently received attention as a natural framework for finding good heuristics in neural optimisation approaches. The question which functions can actually be learnt by message-passing GNNs and which ones exceed their power has been studied extensively. In this talk, I will consider it from a graph-theoretical perspective. I will survey the Weisfeiler-Leman algorithm as a combinatorial procedure to analyse and compare graph structure, and I will discuss some results concerning the power of the algorithm on natural graph classes. The findings directly translate into insights about the power of GNNs and of their extensions to higher-dimensional neural networks.

4.8 Non-Clairvoyant Scheduling with Prediction

Alex Lindermayr (Universität Bremen, DE)

In this talk, we explore some recent and ongoing advancements in non-clairvoyant scheduling in the learning-augmented framework, which integrates error-prone predictions into online algorithm design. We examine various prediction models, showcasing algorithms for non-clairvoyant scheduling with strong error-dependent performance guarantees. In particular, we ask which type of information is required and how an algorithm should receive it to achieve reasonable performance guarantees. Moreover, we consider recently popular prediction models for other problems, and discuss how they may be integrated into scheduling problems.

4.9 Progress and Open Questions on DFL and Constrained NNs

Michele Lombardi (University of Bologna, IT)

This talk will provide perspectives, progress, and open questions on two distinct forms of integration between optimization and learning. First, we will present an analysis of Decision Focused Learning approaches, highlighting structural problem properties that can cause their advantage w.r.t. classical techniques to be significantly diminished. We will suggest a possible way out, with interesting practical applications, together with one solution technique. Second, we consider the problem of enforcing hard constraints in Neural Networks, discussing a training-time approach with satisfaction guarantees. The method combines Projected Gradient Descent with a neural architecture that embeds a trainable over-approximation module. For both the considered setting, we highlight open problems and promising research directions.

4.10 Open Problem: Machine Learning for Edge Contractions in Multilevel Graph Partitinioning

Nikolai Maas (KIT – Karlsruher Institut für Technologie, DE)

In this talk, I will discuss some open problems in applying machine learning for edge contractions in multilevel graph partitioning. This will be in the context of our graph partitioning software Karlsruhe Hypergraph Partioning (KaHyPar).

4.11 Graph Neural Networks as Ordering Heuristics for Parallel Graph Coloring

Fredrik Manne (University of Bergen, NO)

Joint work of Fredrik Manne, Kenneth Langedal

The graph coloring problem asks for an assignment of the minimum number of distinct colors to vertices in an undirected graph with the constraint that no pair of adjacent vertices share the same color. The problem is a thoroughly studied NP-hard combinatorial problem with several real-world applications. As such, a number of greedy heuristics have been suggested that strike a good balance between coloring quality, execution time, and also parallel scalability. In this work, we introduce a graph neural network (GNN) based ordering heuristic and demonstrate that it outperforms existing greedy ordering heuristics both on quality and performance. Previous results have demonstrated that GNNs can produce high-quality colorings but at the expense of excessive running time. The current paper is the first that brings the execution time down to compete with existing greedy heuristics. Our GNN model is trained using both supervised and unsupervised techniques. The experimental

results show that a 2-layer GNN model can achieve execution times between the largest degree first (LF) and smallest degree last (SL) ordering heuristics while outperforming both on coloring quality. Increasing the number of layers improves the coloring quality further, and it is only at four layers that SL becomes faster than the GNN. Finally, our GNN-based coloring heuristic achieves superior scaling in the parallel setting compared to both SL and LF.

4.12 Neural Combinatorial Optimization – One Model to Solve Them All?

Sofia Michel (NAVER Labs Europe – Meylan, FR)

Joint work of Sofia Michel, Darko Drakulic, Jean-Marc Andreoli

Recent years have seen considerable progress in machine learning–based heuristics for combinatorial optimization, particularly constructive neural heuristics that use neural networks to build solutions step by step. Despite their success across a variety of combinatorial tasks, these methods often require specialized models trained separately for each problem and instance distribution. In this talk, I will discuss recent works on improving the generalization of neural combinatorial optimization approaches. I will introduce a framework that leverages the symmetries inherent in many combinatorial problems to boost the out-of-distribution generalization. Building on this framework, I will present GOAL, a generalist model capable of efficiently solving multiple combinatorial problems and which can be fine-tuned to handle new ones. By tackling a broad range of tasks – including routing, scheduling, packing, location, and graph problems – GOAL represents a promising step toward developing a foundation model for combinatorial optimization. I will finish with some current challenges and open questions.

4.13 Machine Learning for Algorithm Selection

Nysret Musliu (TU Wien, AT)

Algorithm selection addresses the challenge of determining which of the available algorithms is most appropriate for solving a particular problem instance. Hyper-heuristics are a high-level, problem-independent approach that aims to automate the design of heuristic methods by combining and selecting low-level heuristics.

In this talk, we will present our work on algorithm selection for various problem domains, including scheduling and tree decomposition. We will also discuss our recent work on reinforcement learning for selection hyper-heuristics.

4.14 Group Counterfactual Explanations with a Mathematical Optimization Lens

María Dolores Romero Morales (Copenhagen Business School, DK)

Counterfactual Analysis has shown to be a powerful tool in the burgeoning field of Explainable Artificial Intelligence. In Supervised Classification, this means associating with each record a so-called counterfactual explanation: an instance that is close to the record and whose probability of being classified in the opposite class by a given classifier is high. In this talk we take a stakeholder perspective, and we address the setting in which a group of counterfactual explanations is sought for a group of instances. We introduce some mathematical optimization models as illustration of each possible allocation rule between counterfactuals and instances and tasks beyond Supervised Classification.

4.15 Optimal Action Imitation Learning

Louis-Martin Rousseau (Polytechnique Montréal, CA)

Predictive models are playing an increasingly pivotal role in optimizing decision-making. This talk introduces an approach wherein a series of (possibly) stochastic sequential decision problems are initially solved offline, and subsequently, a predictive model is constructed based on the offline solutions to facilitate real-time decision-making. The applicability of this methodology is demonstrated through two use cases: patient radiation therapy, where managers need to preserve capacity for emergency cases, and a novel dynamic employee call-timing problem for the scheduling of casual personnel for on-call work shifts.

4.16 Two-Oracle Models for Matroid Optimization Problems

Jens Schlöter (CWI – Amsterdam, NL)

Joint work of Jens Schlöter, Franziska Eberle, Felix Hommelsheim, Alexander Lindermayr, Nicole Megow, Zhenwei Liu

In many combinatorial optimization problems, access to input data is modeled in an abstract way via oracles, e.g., independence oracles for matroids, distance functions in a metric space, or comparators for sorting elements. Depending on the underlying application, the actual use of such oracles can be computationally expensive, as it may require access to slow database systems or large machine learning (ML) models. In the two-oracle model, we assume access to a second, weaker oracle, which is computationally cheap but may provide inaccurate answers. The goal is to design algorithms that exploit the information provided by the cheap oracle to minimize the number of calls to the expensive oracle.

Two-oracle models have been studied in several ML applications, such as data labeling with weak and strong data labelers, text clustering with two oracles for text similarity, or more generally in ML pipelines that combine the use of scalable and expensive models. More

recently, Bai and Coester [NeurIPS'23] considered two-oracle models from the point of view of learning-augmented algorithm design for sorting with access to two comparators. As usual, the goal is to design robust algorithms with performance guarantees that improve with the quality of the second, cheap oracle. In this talk, we discuss the two-oracle model and its applications, and study it from the perspective of learning-augmented algorithms by applying it to the fundamental problem of finding a maximum-weight basis in a matroid.

The talk is based on a joint work with Franziska Eberle, Felix Hommelsheim, Alexander Lindermayr, Nicole Megow, and Zhenwei Liu.

4.17 Graph Learning for Planning

Sylvie Thiébaux (University of Toulouse, FR & Australian National University, Canberra, AU)

Joint work of Sylvie Thiébaux, Dillon Chen, Mingyu Hao, Joerg Hoffmann, Felipe Trevizan

I will present recent work on graph representation learning to guide the search of AI planners. I will introduce GNN and other graph learning representations that exploit the relational structure of planning domains. They allow our planner GOOSE to learn search guidance (e.g. heuristic cost estimates, state rankings) from solutions to just a few small problems, and solve substantially larger problems than trained on. Perhaps surprisingly, our experimental results show that classical machine learning approaches vastly outperform deep learning ones in this context. Moreover, Greedy Best-First Search guided by our best learnt heuristics outperforms the state of the art model-based planner, Lama, on the problems of the latest International Planning Competition Learning track, leading to the possibility that learnt heuristics may replace existing model-based heuristics in the near future.

This is joint work with Dillon Chen, Mingyu Hao, Joerg Hoffmann, and Felipe Trevizan

4.18 Learning-Based Algorithms for Graph Searching Problems

Ali Vakilian (TTIC – Chicago, US)

Joint work of Adela Frances DePavia, Erasmo Tani, Ali Vakilian
Main reference Adela Frances DePavia, Erasmo Tani, Ali Vakilian: "Learning-Based Algorithms for Graph Searching Problems", CoRR, Vol. abs/2402.17736, 2024.
URL http://dx.doi.org/10.48550/ARXIV.2402.17736

We consider the problem of graph searching with prediction recently introduced by [1]. In this problem, an agent, starting at some vertex r has to traverse a (potentially unknown) graph G to find a hidden goal node g while minimizing the total distance travelled. We study a setting in which at any node v, the agent receives a noisy estimate of the distance from v to g. We design algorithms for this search task on unknown graphs. We establish the first formal guarantees on unknown weighted graphs and provide lower bounds showing that the algorithms we propose have optimal or nearly optimal dependence on the prediction error. Further, we perform numerical experiments demonstrating that in addition to being robust to adversarial error, our algorithms perform well in typical instances in which the error is stochastic. Finally, we provide alternative simpler performance bounds on the algorithms of [1] for the case of searching on a known graph, and establish new lower bounds for this setting.

The presentation is based on the results in [2].

References

1 S. Banerjee, V. Cohen-Addad, A. Gupta, and Z. Li. Graph Searching with Predictions. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023.

2 A. F. DePavia, E. Tani, and A. Vakilian. Learning-Based Algorithms for Graph Searching Problems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 928–936, 2024. PMLR.

4.19 Serving MPE Queries on Tensor Networks by Computing Derivatives

Maurice Wenig (Friedrich-Schiller-Universität Jena, DE)

Joint work of Maurice Wenig, Hanno Barschel, Joachim Giesen, Andreas Goral, Mark Blacher
Main reference Maurice Wenig, Hanno Barschel, Joachim Giesen, Andreas Goral, Mark Blacher: "Serving MPE Queries on Tensor Networks by Computing Derivatives", in Proc. of the Proceedings of The 12th International Conference on Probabilistic Graphical Models, Proceedings of Machine Learning Research, Vol. 246, pp. 515–527, PMLR, 2024.
URL https://proceedings.mlr.press/v246/wenig24a.html

Recently, tensor networks have been proposed as a data structure for weighted model counting. Computing a weighted model count is thus reduced to contracting a factorized tensor expression. Inference queries on graphical models, especially PoE (probability of evidence) queries, can be expressed directly as weighted model counting problems. Maximization problems can also be addressed on the same data structure, only the standard sum-product semiring has to be replaced by either the tropical (max-sum) or the Viterbi (max-product) semiring in the computations, that is, the tensor contractions. However, tensor contractions only provide maximal values, but MPE (most probable explanation) queries on graphical models do not ask for the maximal value, but for a state, or even the states, at which the maximal value is attained. In the special case of tropical tensor networks for ground states of spin glasses, it has been observed that the ground state can be obtained by computing a derivative of the tensor network over the tropical semiring. Here, we generalize this observation, provide a generic algorithm for computing the derivatives, and prove its correctness.

5 Research Directions

5.1 Establishing connections between the related paradigms of machine learning for combinatorial optimization, decision-focused learning, algorithms with predictions and neuro-symbolic computing

We discussed key connections, shared concepts and research directions across the related paradigms of machine learning for combinatorial optimization, decision-focused learning (DFL), algorithms with predictions, and neuro-symbolic computing.

Theme 1: Finding a "Backbone" for Combinatorial Problems. We observed a recurring theme across these paradigms: the identification of a "backbone," a crucial subset of variables that significantly influences the solution of a combinatorial problem. This concept manifests in various forms, such as "learning to prune," "predict then search," and "finding kernel," across different research communities. Herein, we outline key research questions related to formally defining the concept of "backbone" and its effective computation.

- Formalizing the "Backbone" Concept and its Computation: A key challenge lies in formally defining the "backbone" in a general context and finding a generic computational technique to detect a "backbone". While specific problem-tailored techniques for computing "backbone" exist, a more general technique, applicable to problems such as Mixed-Integer Programming (MIP), Constraint Satisfaction Problems (CSP), and general Combinatorial Optimization Problems (COP), is highly desirable.
- Leveraging Symmetries: Investigating how problem symmetries can be exploited to identify and refine the "backbone" represents a promising research direction.
- Constraint Programming Considerations: Exploring the concept of the "backbone" within the context of constraint programming solvers can potentially result in more effective constraint programming solvers.
- Guaranteeing Solution Quality: A critical aspect is to ensure that identifying and potentially reducing the problem based on the "backbone" does not compromise the feasibility or optimality of the solution. Rigorous analysis of the potential risks and error bounds associated with "backbone" predictions is essential.

Theme 2: Learning Effective Proxy Solvers. This theme focused on defining and developing effective "proxy solvers". The research in this direction needs to be cognizant of whether the proxy is being used during training or also during the solving process.

Qualities of a good proxy.

- Accuracy: The proxy solver must accurately reflect the behavior of the original solver, aligning with performance metrics such as regret in decision-focused learning or closely approximating the initial solver's output.
- Computational Efficiency: The proxy solver must be computationally inexpensive to evaluate, ideally allowing for efficient gradient-based optimization and backpropagation.
- Theoretical Guarantees: Establishing theoretical guarantees for the proxy solver is crucial. For example, providing bounds on the error or ensuring that the proxy satisfies specific properties (e.g., ε-approximation) is highly desirable.

Use Cases for Proxy Solvers.

- DFL Objective Functions: Proxy solvers can serve as effective objective functions within the decision focused learning frameworks.
- Improving Combinatorial Solvers: Proxy solvers can be used to enhance the performance of existing combinatorial solvers, such as the use of linear relaxation for obtaining an upper bound.
- Model Pretraining: Proxy solvers can be used to pretrain machine learning models for combinatorial optimization tasks.
- Regret Estimation in DFL: Proxy solvers can be employed to estimate the regret in decision focused learning settings.

Building Effective Proxies. This line of research needs to take into account whether or not we want the proxy to preserve feasibility of solutions. Herein, we outline some potential approaches that can be explored to build effective proxies.

- Contrastive loss function
- Exploring the use of Lagrangian relaxation and decomposition techniques to construct proxies in a differentiable way
- Investigating the applicability of scheduling methods, such as the *envelope method*
- Differentiable Feasibility Pumps
- Investigating the relationship between decision focused learning and inverse optimization and exploring the potential for leveraging the decision focused learning techniques to solve inverse optimization problems

Theme 3: Defining and Exploring Neuro-Symbolic Solvers. This theme focused on defining and exploring the concept of "neuro-symbolic solvers." We can explore the space of "neuro-symbolic solvers" by considering the elements in the "Neuro", "Symbolic" and "Solver" parts.

- **Elements inside the *Neuro* part**: A neural network
- **Elements inside the *Symbolic* part**: A logic (e.g., first-order, fuzzy, temporal, etc.), a search procedure (a CP, MIP, SAT solver, etc.), Markov Random field or a graphical model
- **Elements inside the Solver part**: A way to solve optimization/satisfaction problem

Examples of neuro-symbolic solvers include **DeepProbLog**, **DeepStochLog**, **Toulbar** and methods for weighted model counting. We noted that significant similarities exist between neuro-symbolic approaches and decision focused learning. Historically, neuro-symbolic solvers have often focused on logic, while decision focused learning has primarily focused on constrained optimization. A thorough comparative analysis is needed to identify potential synergies and avoid redundant research efforts in these areas.

5.2 Theoretical frameworks for machine learning augmented combinatorial optimization algorithms

This discussion focused on identifying commonalities in theoretical frameworks and results between contextual optimization, decision-focused learning (DFL), ML augmented CO solvers and algorithms with predictions. For the purpose of this discussion, we established that DFL can be considered a sub-class of contextual optimization problems. We observed that many theoretical results in CO/DFL rely on learning theory, focusing on out-of-sample generalization based on in-sample training data. While ML-augmented CO solvers also exhibit generalization properties, extending these results to prove stronger generalization bounds for techniques like ML-augmented branch-and-bound would be highly valuable. On the other hand, algorithms with predictions often focus on consistency-robustness trade-offs, considering the level of prediction performance as an input. Understanding how to merge these results with the statistical generalization bounds prevalent in CO/DFL would significantly advance our understanding of these interconnected fields.

Specific research questions that emerged in this theme include:

- Robustness in ML-Augmented CO: Can we better understand how ML augmented CO solvers have robustness guarantees that are stronger/more fine-grained than just preserving optimality in the worst-case, and instead depend on the learning guarantees and may use distribution shift concepts (for example) to understand how robust ML augmented solvers are when they are applied to out of distribution data?

- Defining Robustness for CO/DFL: What constitutes the "correct" notion of robustness for CO/DFL? If we consider learning with a perfect future information (PFL) model as a baseline, how can we theoretically compare the performance of DFL and PFL models in a meaningful way?
- Point Predictions in Nonlinear Settings: Under what conditions are point predictions sufficient for 2-stage problems and more generally, for nonlinear optimization problems? How significant can the bias of DFL-trained models be?
- Can we prove some consistency-robustness properties when algorithms with predictions framework is used on branch-and-bound problems? Can we build a bridge between generalization guarantees and algorithms-with-predictions guarantees?
- Fairness and Out-of-Distribution Considerations: How can we incorporate fairness properties and out-of-distribution robustness into DFL formulations? Can counterfactual explanations be leveraged to build fairer DFL models? Can we establish generalization guarantees for fairness metrics in DFL?
- Computational Complexity in DFL, algorithms with prediction and ML augmented combinatorial optimization: How can we define and analyze the computational complexity in these areas, considering factors like the number of gradient or oracle calls? How can we optimally trade off the computational costs of training, inference, and optimization within the DFL framework? What are the optimal sample sizes for training ML-augmented CO algorithms, and are there scaling laws that govern this relationship? Can we get useful lower bounds on the sub-optimality gap while training?
- Dynamic Optimization and Regret: How can we effectively apply ML techniques in dynamic optimization settings, especially when traditional regret bounds may not be well-defined? How can we dynamically update ML models to adapt to changing environments?

5.3 Learning to fill the underspecified parts of algorithms

Many combinatorial optimization algorithms incorporate design choices that do not impact their worst-case theoretical bounds. However, these choices can significantly influence their practical performance, e.g., in terms of running time. By leveraging machine learning techniques to learn optimal or near-optimal choices for typical input instances, we can significantly enhance the efficiency of these algorithms.

We considered two concrete examples:

- The classic Bellman-Ford algorithm for finding shortest paths with negative edge weights relaxes all edges multiple times. While the worst-case running time remains unaffected by the order of edge relaxations, the practical performance can vary dramatically. By learning a near-optimal relaxation order for a given input distribution, we can potentially reduce the number of iterations required for convergence, significantly improving the algorithm's efficiency.
- In graph partitioning, particularly ϵ-balanced partitioning, multilevel approaches are state-of-the-art. These methods involve coarsening the graph by repeatedly contracting edges, generating an initial partitioning, and then uncoarsening to obtain a partition for the original graph. Current coarsening strategies often rely on greedy heuristics. However, by learning which edges should not be contracted using machine learning techniques, we can potentially improve the quality of the initial partitioning and ultimately the final solution.

Scalability Challenges. A crucial challenge in learning to fill these underspecified parts lies in scalability. Many algorithms, especially those operating on large graphs, require processing millions of edges per second. This stringent requirement necessitates the use of local features, as computing global features can be computationally prohibitive. For instance, in the graph partitioning context, determining which edges should not be contracted often requires global information. However, due to scalability constraints, we are typically limited to using local edge features. To overcome this limitation, we can explore techniques for overestimating global information using local features.

Input Instance Dependence and Training Data. Learning models must be tailored to specific input distributions. Different input distributions may require distinct learning models to achieve optimal performance.

- Data Scarcity and Bias: Obtaining sufficient and representative training data can be challenging. Existing datasets for some graph algorithms may exhibit biases, potentially hindering the generalization ability of learned models.
- Artificial Instance Generation: Generating artificial instances that challenge the algorithm can be crucial for improving the robustness and generalizability of learned models.
- Imitation Learning: Imitation learning techniques can be valuable for collecting high-quality training data by observing the behavior of expert algorithms or human experts.

Dynamic Ordering and Feature Propagation.

- Dynamic Ordering: For algorithms like Bellman-Ford, it might be more effective to learn dynamic orderings, where the order of relaxations changes in each iteration based on information gathered in previous iterations.
- Feature Propagation: In graph-based algorithms, propagating features within the search space (e.g., within a breadth-first search (BFS) tree) can provide valuable information to guide the search process more effectively. Techniques such as graph propagation can be explored to enable such feature propagation.
- Addressing Error Propagation: In iterative algorithms, errors made in early iterations can propagate and significantly impact subsequent iterations. This can pose a challenge for learning-based approaches. Careful consideration must be given to mitigating the impact of early errors on the overall learning process.

5.4 Reinforcement Learning for Combinatorial Optimization

In the context of leveraging reinforcement learning for combinatorial optimization, there has been some progress on end-to-end reinforcement learning (RL) algorithms for combinatorial optimization problems with no global constraints, and for which all constraints can be checked incrementally at each step. The key challenge in this area lies for problems with at least one global constraint. In this case, determining whether a candidate solution satisfies the global constraints often requires evaluating the solution as a whole. This presents a significant challenge, as it is not possible to incrementally check constraint satisfaction during the solution construction process. Potential approaches to address this challenge include:

- Penalty-based Methods: Penalizing “nogoods” (violations of global constraints) during RL training.
- Repair Strategies: Checking constraints at the end of the solving process and then repairing the infeasible solution.

- Predicting Feasibility: Predicting the likelihood that a partial solution can be extended to a complete, consistent solution.
- Perturbation Methods: Working with perturbations of existing feasible solutions rather than directly constructing solutions from scratch.

In contrast to integrating RL within or alongside a combinatorial optimization solver, end-to-end RL approaches may require backtracking mechanisms to handle constraint infeasibility.

A crucial challenge lies in the generalizability of end-to-end and learning-augmented solving. This includes generalization across problem instance sizes, generalization between related problem classes, and the inherent generality of the underlying combinatorial optimization solvers.

Key research questions to address in this area also include (i) developing a comprehensive theoretical framework for reinforcement learning in the context of combinatorial optimization problems and (ii) exploring the potential benefits of active learning techniques for improving the efficiency and effectiveness of learning-based optimization solvers.

5.5 Contextual Stochastic Optimisation and DFL

A key question in the area of machine learning for combinatorial optimization is "Can we develop foundation models that capture the underlying structural properties of different combinatorial optimization tasks?" These foundation models could then be fine-tuned for specific pruning or branching tasks.

Herein, we list some of the other research questions identified in this theme:

- Model Depth: A fundamental question is: How much expressive power, specifically in terms of model depth (e.g., the number of layers in a graph neural network), is necessary for an ML model to effectively address various combinatorial optimization tasks? For instance, what level of expressivity is required for a GNN to act as an effective proxy solver or for learning-to-prune?
- What are the advantages of decision focused learning (DFL) over the standard mean squared error loss functions?
- How can neural networks be effectively trained within the DFL framework?
- What biases get introduced when using point prediction for stochastic settings? There are potential connections between these biases and the field of inverse optimization, which aims to infer the underlying parameters of an optimization problem given observed decisions.
- Can we enable more compact and efficient representations of complex optimization problems in higher-dimensional spaces?
- Drawing inspiration from NLP, where foundation models like large language models (LLMs) are pre-trained on large text corpora, can we explore analogous pre-training tasks for combinatorial optimization? What simple, yet informative, tasks can be used for pre-training in the context of combinatorial optimization? In NLP, next-token prediction serves as a powerful pre-training objective. Can we identify an analogous task for combinatorial optimization problems, perhaps involving predicting partial solutions or exploring the local neighborhood of a given solution?
- What constitutes suitable training data for pre-training such foundation models? Should it include solutions, explored search trees, or other relevant information?

5.6 Fast and/or Interpretable ML Techniques for Combinatorial Problems

When integrating machine learning (ML) models into combinatorial optimization algorithms, the trade-off between efficiency, interpretability and accuracy is crucial. We first outline the motivation for efficiency and interpretability of the learning techniques in the context of optimization problems.

- Efficiency: If an ML model is used as a subroutine within a combinatorial optimization algorithm, it might be invoked very frequently. A fast inference of the learnt model can help ensure scalability of the optimization algorithm to large data.
- Interpretable models can facilitate the development of hand-crafted heuristics, enabling manual optimization and providing a deeper understanding of the algorithm's behavior.
- Interpretability enhances our understanding of how ML-augmented algorithms function, improving our ability to analyze and debug them.

Next, we consider the two popular models in terms of this trade-off between efficiency, interpretability and accuracy.

- Decision Trees: Offer high efficiency and interpretability but may have limited expressive power for complex tasks.
- Neural Networks (NNs): More expressive but generally less efficient and more difficult to interpret.

Specifically, the discussion focused on Graph Neural Networks (GNNs) and we noted that the expressive power of GNNs (without additional features) is equivalent to the Weisfeiler-Leman graph isomorphism test. This connection can be leveraged to reinterpret GNN decisions in terms of the colors assigned by the WL algorithm. However, developing automated techniques for interpreting GNN decisions beyond manual analysis of WL colorings remains an open challenge.

In terms of interpretability of standard neural networks, increasing the number of features or hidden layers in a NN significantly increases its complexity and hinders interpretability. However, the following techniques are used in this context:

- Distillation: Train a simpler model (e.g., using symbolic regression) to imitate the behavior of the original NN.
- LIME: Ranks the feature importance for specific decisions and given input instances.
- Visualization Techniques: Visualize the activation patterns within the NN to understand how it processes information.
- Encode the learning model as a combinatorial optimization problem, e.g., a MIP.
- Model Simplification: Pruning of low-weight connections and quantization can help to simplify a NN.

Unfortunately, the applicability of these techniques is limited and they are often not sufficient to understand complex NNs.

In terms of improving efficiency, the following techniques were discussed:

- Modular Subproblems: Decompose complex problems into smaller, more manageable subproblems, which can be solved more efficiently. An example application for this is planning.
- Select the appropriate ML model (classification, regression, ranking) based on the specific task to optimize efficiency and interpretability.
- GPU Acceleration: Utilize GPUs for efficient inference, although data transfer overhead must be considered. GPU acceleration can be particularly beneficial when the algorithm itself already runs on the GPU.

5.7 Scalability and Datasets for Decision Focused Learning

Decision-focused learning (DFL) has witnessed significant progress in recent years. However, to achieve real-world success and broader applicability, DFL methods must become more scalable. This session focused on open challenges and potential contributions to DFL in terms of:

- Open-Source Code and Datasets: Expanding the availability of open-source code and datasets specifically designed for DFL applications.
- Scalability for Real-World Problems: Enhancing the ability of DFL methods to handle the complexities of real-world data and problem sizes.

Currently, the most widely used open source software for DFL is PyEPO by Khalil et al.[1]. Although it has some limitations, such as being able to be used only for the objective coefficients of a single linear programming instance, it is still a good starting point for implementing DFL methods. The repository of the DFL survey paper [1] has implementations for several problems, namely, optimization, knapsack, matching, portfolio optimization, (Warcraft) shortest path independent of PyEPO.

A key hurdle in research on DFL methodologies is finding real-world data that aligns well with DFL's requirements. For example, in many routing applications, the travel time information is only observed for the selected path. A potential solution to this hurdle is to generate optimization problems synthetically whenever appropriate data is available. An example is the bike-sharing applications where there is a lot of data made available, but a synthetic problem such as a shortest path or traveling salesman problem can be defined on this real data. In addition, there is a need to develop a standardized format for storing data and optimization problems specifically for DFL applications. This would facilitate sharing and collaboration within the DFL research community.

Scalability in DFL can encompass various aspects, including the size of the optimization problem and the available data volume. Notably, problem size doesn't always directly correlate with optimization instance difficulty. To address the scalability issues, a potential approach is to train the learning model with small instances of the optimization problem, and then perform inference on larger instances. This kind of scaling can be done, for example, when the mapping from the features to the problem parameters is deterministic, but the feature is random. This mapping can be learned for small instances in a DFL fashion, and the learned mapping can be used in larger instances.

By addressing these challenges and fostering collaboration through open-source code and datasets, DFL can unlock its full potential for solving real-world decision-making problems at scale.

References

1 Mandi, Jayanta, et al. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. Journal of Artificial Intelligence Research 80 (2024): 1623-1701.

[1] https://github.com/khalil-research/PyEPO

5.8 Finding worst-case instances for algorithms using machine learning techniques

An important research direction at the intersection between algorithm design and machine learning is to design a learning framework that effectively generates instances that expose the worst-case performance of a given algorithm. Such a framework can greatly accelerate the pace of research in algorithm design and engineering. This is particularly relevant for approximation algorithms where the gap between the upper and lower bounds is very high and the worst-case instances are those that elicit the worst approximation ratio. Such a framework would enable researchers to:

- Identify Algorithmic Weaknesses: Pinpoint specific instances that highlight limitations and potential areas for improvement in existing algorithms.
- Refine Approximation Ratios: Tighten lower bounds on the approximation ratio, providing a more accurate understanding of the algorithm's performance guarantees.
- Guide Algorithm Design: Inform the development of new algorithms that are more robust against the identified worst-case scenarios.

In recent years, some progress was made in this direction by Wagner [1] who used a reinforcement learning framework based on cross-entropy to find counterexamples to some open conjectures. Another recent work [2] improved the state-of-the-art lower bounds for a central extremal graph theory problem which aims to find graphs with a given size (number of nodes) that maximize the number of edges without having 3- or 4-cycles. Recent work [3] in algorithm configuration community has also addressed this question, but from a different perspective.

Since geometric problems generally offer more intuitive visualizations, potentially facilitating the learning and analysis process, we can focus on geometric problems as a more tractable starting point for this research. Here are a couple of examples of such worst-case instances:

- Can we find a configuration of rectangles such that the intersection graph of those rectangles corresponds to a given graph G?
- Can we find a configuration of rectangles such that the ILP formulation (that ensures that the sum of rectangle indicator variables covering any given point sums to atmost 1) for its maximum independent set will have the worst-possible integrality gap?

Potential approaches to address this research direction include:

- Black-Box Optimization: Treat the generation process as a black-box optimization problem, exploring techniques like gradient descent and evolutionary algorithms.
- Penalizing Configurations: Penalize configurations that are known to be not useful for eliciting the worst-case behaviour, including instances for which reduction rules apply easily.
- Solver-Guided Learning: Utilize existing solvers and generators to generate a large number of instances, analyze the performance of the target algorithm on these instances, and use this data to guide the learning process.
- Exploiting Known Hard Instances: Leverage existing knowledge about hard instances to guide the generation process, potentially by creating variations or generalizations of these known difficult configurations.
- Recursive Subgraph Construction: Explore the use of recursive techniques to construct complex structures by combining simpler, known hard subgraphs.
- Identify variables that are always 0 or always 1 in all optimal solutions and removing these variables to simplify and reduce the problem.

References

1 Adam Zsolt Wagner. Constructions in combinatorics via neural networks. CoRR abs/2104.14516 (2021)

2 Abbas Mehrabian, Ankit Anand, Hyunjik Kim, Nicolas Sonnerat, Matej Balog, Gheorghe Comanici, Tudor Berariu, Andrew Lee, Anian Ruoss, Anna Bulanova, Daniel Toyama, Sam Blackwell, Bernardino Romera-Paredes, Petar Velickovic, Laurent Orseau, Joonkyung Lee, Anurag Murty Naredla, Doina Precup and Adam Zsolt Wagner. Finding Increasingly Large Extremal Graphs with AlphaZero and Tabu Search Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence IJCAI 2024

3 Kate Smith-Miles and Mario Andrés Muñoz. Instance Space Analysis for Algorithm Testing: Methodology and Software Tools. ACM Computing Survey 2023

Participants

- Deepak Ajwani
University College Dublin, IE
- Brandon Amos
Meta – New York, US
- Senne Berden
KU Leuven, BE
- Quentin Cappart
Polytechnique Montréal, CA
- Eanna Curran
University College Dublin, IE
- Marianne Defresne
KU Leuven, BE
- Michelangelo Diligenti
University of Siena, IT
- Bistra Dilkina
USC – Los Angeles, US
- Adam Elmachtoub
Columbia University – New York, US
- Aaron Ferber
Cornell University – Ithaca, US
- Alexandre Forel
Polytechnique Montréal, CA
- Emma Frejinger
University of Montreal, CA
- Paul Grigas
University of California – Berkeley, US
- Ernestine Großmann
Universität Heidelberg, DE
- Tias Guns
KU Leuven, BE
- Mikhail Khodak
Princeton University, US
- Sandra Kiefer
University of Oxford, GB
- Alex Lindermayr
Universität Bremen, DE
- Michele Lombardi
University of Bologna, IT
- Nikolai Maas
KIT – Karlsruher Institut für Technologie, DE
- Irfan Mahmutogullari
KU Leuven, BE
- Fredrik Manne
University of Bergen, NO
- Johannes Meintrup
THM – Gießen, DE
- Ulrich Carsten Meyer
Goethe University – Frankfurt am Main, DE
- Sofia Michel
NAVER Labs Europe – Meylan, FR
- Nysret Musliu
TU Wien, AT
- Mathias Niepert
Universität Stuttgart, DE
- Siegfried Nijssen
UC Louvain, BE & KU Leuven, BE
- Ryan O Connor
University College Dublin, IE
- Manuel Penschuck
Goethe University – Frankfurt am Main, DE
- Adam Polak
Bocconi University – Milan, IT
- María Dolores Romero Morales
Copenhagen Business School, DK
- Louis-Martin Rousseau
Polytechnique Montréal, CA
- Jens Schlöter
CWI – Amsterdam, NL
- Christian Schulz
Universität Heidelberg, DE
- Darren Strash
Hamilton College – Clinton, US
- Edward Tansley
University of Oxford, GB
- Sylvie Thiébaux
University of Toulouse, FR & Australian National University, Canberra, AU
- Ali Vakilian
TTIC – Chicago, US
- Jacobus G. M. van der Linden
TU Delft, NL
- Maurice Wenig
Friedrich-Schiller-Universität Jena, DE
- Neil Yorke-Smith
TU Delft, NL