Report from Dagstuhl Seminar 25371

# Interactions in Constraint Optimization

**Katalin Fazekas**[*1], **Matti Järvisalo**[*2], **Nina Narodytska**[*3], **Peter J. Stuckey**[*4], **and Christoph Jabs**[†5]

1 **TU Wien, AT.** katalin.fazekas@tuwien.ac.at
2 **University of Helsinki, FI.** matti.jarvisalo@helsinki.fi
3 **VMware Research – Palo Alto, US.** n.narodytska@gmail.com
4 **Monash University – Caulfield, AU.** peter.stuckey@monash.edu
5 **University of Helsinki, FI.** christoph.jabs@helsinki.fi

**Abstract**

This report documents the Dagstuhl Seminar 25371 "Interactions in Constraint Optimization". Our Dagstuhl Seminar gathered 41 researchers from 15 countries, working on different constraint optimization paradigms. The report consists of an executive summary, and abstracts on tutorials, research talks, and panel discussions.

**Seminar** September 7–12, 2025 – https://www.dagstuhl.de/25371
**2012 ACM Subject Classification** Mathematics of computing → Combinatorial algorithms; Theory of computation → Discrete optimization
**Keywords and phrases** constraint programming, maximum satisfiability, mixed integer linear programming, optimization modulo theories, pseudo-boolean optimization
**Digital Object Identifier** 10.4230/DagRep.15.9.1

## 1 Executive Summary

*Katalin Fazekas (TU Wien, AT)*
*Matti Järvisalo (University of Helsinki, FI)*
*Nina Narodytska (VMware Research – Palo Alto, US)*
*Peter J. Stuckey (Monash University – Caulfield, AU)*

Optimization problems are one of the most common problems that real-world applications face: scheduling, rostering, hardware verification, vehicle routing, to name a few. Constraint optimization is the main technology to solve these problems in practice as it offers a principled way to explore the search space and find an optimal (or good enough) solution. However, as the complexity of real-world applications increases and these techniques are applied in safety critical applications, the need for more advanced features and techniques to be developed grows. In fact, the term constraint optimization can be seen as an umbrella term that covers a number of well-developed and practically useful technologies. Our main goal was to see how these technologies can leverage each other to increase practicality and adoption of optimization techniques.

This report documents the program and the outcomes of Dagstuhl Seminar 25371 "Interactions in Constraint Optimization". We gathered leading researchers from following research areas working on constraint optimization:

- Constraint Programming (CP),
- Mixed Integer Linear Programming (MIP),
- Boolean Satisfiability (SAT) / Maximum Satisfiability (MaxSAT),

[*] Editor / Organizer
[†] Editorial Assistant / Collector





- Satisfiability Modulo Theories (SMT) / Optimization Modulo Theories (OMT/MaxSMT), and
- Pseudo-Boolean optimization (PBO).

The main focus of the seminar was to foster interactions between these communities and consider how to build synergy between them in the coming decade. The seminar was structured to provide necessary background to the participants through tutorials presented by leading researchers in the corresponding subfield of optimization. These tutorials aimed to bring the audience up to speed on the state-of-the-art advances in each subarea. We had a number of contributing talks that reported on recent results and in-progress research developments. These fostered interesting discussions and outlined potential collaborations.

We held two panels to discuss future directions and collaborative opportunities. We believe we had fruitful discussions that triggered new ideas. The report presents a summary of talks, discussions and panel outcomes.

## 2 Table of Contents

# 3 Overview of Talks

## 3.1 Uncovering and Classifying Bugs in MaxSAT Solvers through Fuzzing and Delta Debugging

*Armin Biere (Universität Freiburg, DE)*


**Joint work of** Tobias Paxian
**Main reference** Tobias Paxian, Armin Biere: "Uncovering and Classifying Bugs in MaxSAT Solvers through Fuzzing and Delta Debugging", in Proc. of the 14th International Workshop on Pragmatics of SAT co-located with the 26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023), Alghero, Italy, July 4, 2023, CEUR Workshop Proceedings, Vol. 3545, pp. 59–71, CEUR-WS.org, 2023.
**URL** https://ceur-ws.org/Vol-3545/paper5.pdf

We give a brief introduction into existing fuzzing and delta debugging techniques in automated reasoning with focus on SAT including model based testing of SAT solver APIs. Then we discuss extensions to MaxSAT and present results on cross checking MaxSAT solvers submitted to the last three MaxSAT evaluations through these techniques.

## 3.2 Parallelising CP-SAT

*Toby Davies (Google – Pyrmont, AU), Frédéric Didier (Google – Paris, FR), Laurent Perron, and Peter J. Stuckey (Monash University – Caulfield, AU)*


**Main reference** Toby O. Davies, Frédéric Didier, Laurent Perron, Peter J. Stuckey: "Parallelising Lazy Clause Generation with Trail Sharing", in Proc. of the Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pp. 205–221, Springer Nature Switzerland, 2025.
**URL** https://doi.org/10.1007/978-3-031-95973-8_13

CP-SAT implements many complementary ideas and runs them in a parallel portfolio, but this is just part of its parallelism approach. Workers share information, about variable and objective bounds, and a subset of high-quality clauses. In addition to these traditional approaches "shared tree workers" also partition the search space, and perform "trail sharing" to complement traditional clause sharing.

## 3.3 MIP aspects of Google OR-tools CP-SAT solver

*Frédéric Didier (Google – Paris, FR)*



While CP-SAT focuses on pure integer-problems, a lot of its design is similar to the one of a "classic" MIP solver and they can be used to solve optimization problems in mostly the same way. After explaining the class of MIP problems that CP-SAT can deal with, I will dive into some of its implementation details. I will focus on the handling of linear constraints and how we use the LP relaxation of the problem, both being critical components like they are in MIP.

## 3.4 Large Neighbourhood Search (LNS)

*Pierre Flener (Uppsala University, SE)*



LNS is a hybrid solving technology for optimisation problems. A local-search algorithm invokes at every iteration a systematic-search solver (usually a CP solver) in order to explore a large neighbourhood that it constructs. I give a brief tutorial on LNS, show how to extend it to satisfaction problems, and outline its research frontier.

## 3.5 Towards Quantifying Fairness and Designing Interpretable Machine Learning: A Formal Methods Approach

*Bishwamittra Ghosh (MPI-SWS – Kaiserslautern, DE)*


**Joint work of** Bishwamittra Ghosh, Kuldeep Meel, Debabrota Basu, Dmitry Malioutov
**Main reference** Bishwamittra Ghosh, Debabrota Basu, Kuldeep S. Meel: "Justicia: A Stochastic SAT Approach to Formally Verify Fairness", in Proc. of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pp. 7554–7563, AAAI Press, 2021.
**URL** https://doi.org//10.1609/AAAI.V35I9.16925

Despite its widespread success, machine learning faces critical societal challenges, including unfair predictions and a lack of interpretability. My research, at the intersection of formal methods and machine learning, develops tools based on formal reasoning, satisfiability solving, and constrained optimization to formally quantify classifier fairness, identify sources of unfairness, and design inherently interpretable rule-based classifiers. In particular, I will discuss the application of stochastic satisfiability (SSAT) and maximum satisfiability (MaxSAT) in addressing fairness and interpretability, advancing the goal of trustworthy machine learning.

## 3.6 Tutorial: LP relaxations in MIP solving

*Ambros Gleixner (HTW – Berlin, DE) and Gioni Mexi (Zuse-Institut Berlin, DE)*



In this tutorial we survey the basics of how modern MIP solving uses, solves, and strengthens LP relaxations with pointers to recent trends and computational aspects. In particular, we touch upon valid inequalities and the geometry behind infeasibility analysis and Dantzig-Wolfe reformulation.

## 3.7 Constraint Optimisation as an accessible AI toolbox

*Tias Guns (KU Leuven, BE)*


**URL** https://wms.cs.kuleuven.be/chat-opt

Our research combines constraint optimisation with machine learning and explainability. For this, we need easy and efficient access to multiple solvers; as most of the work involves novel integrations. This is the case for industry projects, like our CP25 best application paper award on workforce scheduling, on our work evaluating LLMs for constraint modeling, and earlier neuro-symbolic projects like the Sudoku Assistant. To do this research, we had to build the CPMpy library. A Python library that translates to CP, SMT, ILP, PB and SAT solvers. To support all these translations, we converged on an elegant "waterfall" model, showing how much transformations are shared between the different transformations. We'll also highlight gaps and new possiblities: the need for standard APIs, for open source implementations of translations and components, and how this can enable cross-technology evaluations and dataset generation.

## 3.8 Tutorial: Maximum Satisfiability Solving

*Alexey Ignatiev (Monash University – Clayton, AU)*



This talk provides a tutorial on Maximum Satisfiability (MaxSAT) solving, a powerful declarative paradigm for tackling complex optimisation problems across a wide range of domains. The tutorial begins by defining the MaxSAT problem as an optimisation counterpart of Propositional Satisfiability. It then covers the fundamentals of representing practical problem constraints as propositional formulas in conjunctive normal form (CNF), using hard and soft clauses, illustrating how non-clausal constraints can be "clausified" using techniques like Tseitin transformation. The core of the talk focuses on state-of-the-art, core-guided MaxSAT algorithms. It explains the foundational principle of iteratively identifying and relaxing unsatisfiable cores. The presentation details the evolution of these methods, leading to the OLL algorithm, which introduces the key concept of relaxable cardinality constraints. Finally, it discusses practical considerations and performance enhancements as implemented in the open-source RC2 solver, including incremental SAT solving, incremental cardinality constraints, core exhaustion, Boolean lexicographic optimisation, and stratification.

## 3.9 IHS for PBO: Key Techniques in a State-of-the-Art Solver and Recent Developments

*Hannes Ihalainen (University of Helsinki, FI)*


**Joint work of** Hannes Ihalainen, Dieter Vandesande, André Schidler, Jeremias Berg, Bart Bogaerts, Matti Järvisalo
**Main reference** Hannes Ihalainen, Dieter Vandesande, André Schidler, Jeremias Berg, Bart Bogaerts, Matti Järvisalo: "Efficient and Reliable Hitting-Set Computations for the Implicit Hitting Set Approach", CoRR, Vol. abs/2508.07015, 2025.
**URL** https://doi.org//10.48550/ARXIV.2508.07015

Recently, the so-called implicit hitting set (IHS) approach has proven to be a succesful method for pseudo-Boolean optimization (PBO). To achieve practical competitiveness, IHS solvers incorporate many additional techniques in addition to the basic IHS algorithm. This talk provided an overview of the techniques implemented in a state-of-the-art PBO-IHS solver. In addition, the talk highlighted some recent advances in PBO-IHS: a symmetric core learning technique to tackle highly symmetric instances, and efforts to develop alternative methods – such as PB reasoning and stochastic local search – for hitting set computations, aimed at making solving trustworthy via proof logging and improving performance in practice.

## 3.10 Multi-Objective Optimization: A Pseudo-Boolean Perspective

*Christoph Jabs (University of Helsinki, FI)*


**Joint work of** Christoph Jabs, Jeremias Berg, Matti Järvisalo
**Main reference** Christoph Jabs, Jeremias Berg, Matti Järvisalo: "Engineering and Evaluating Multi-objective Pseudo-Boolean Optimizers", in Proc. of the Logics in Artificial Intelligence, pp. 115–134, Springer Nature Switzerland, 2026.
**URL** https://doi.org/10.1007/978-3-032-04587-4_8

Various real-world settings give rise to combinatorial optimization problems with multiple conflicting objectives, motivating the development of practical approaches to the challenging task of finding Pareto-optimal solutions to declarative models of multi-objective problems. In this talk we view multi-objective optimization through the lens of pseudo-Boolean constraints (MO-PBO) as an extension of propositional clauses and, at the same time, an important class of 0-1 linear constraints. We provide a first-of-kind cross-community evaluation of a selection of recently-proposed approaches applicable to MO-PBO, including first implementations of native MO-PBO algorithms we provide, as well as approaches based on integer linear programming techniques and a translation-based approach to MO-MaxSAT, providing insights into the current state-of-the-art approaches to MO-PBO, and a glimpse into how these paradigms compare more generally.

## 3.11 Core-Guided Linear Programming-based Maximum Satisfiability

*George Katsirelos (INRAE – Palaiseau, FR)*


**Main reference** George Katsirelos: "Core-Guided Linear Programming-Based Maximum Satisfiability", in Proc. of the 28th International Conference on Theory and Applications of Satisfiability Testing, SAT 2025, Glasgow, Scotland, August 12-15, 2025, LIPIcs, Vol. 341, pp. 17:1–17:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.
**URL** https://doi.org//10.4230/LIPICS.SAT.2025.17

The core-guided algorithm OLL is the basis of some of the most successful algorithms for MaxSAT in recent evaluations. It works by iteratively finding cores of the formula and transforming it so that it exhibits a higher lower bound. It has recently been shown to implicitly discover cores of the original formula, as well as a compact representation of its reasoning within a linear program. In this paper, we use and extend these results to design a practical MaxSAT solver. We show an explicit linear program which matches and usually exceeds the bound computed by OLL. We show that OLL can be restated as an algorithm that explicitly computes a feasible dual solution of this linear program. This restated algorithm naturally works with an arbitrary dual solution. It can in fact be used to improve any LP representation of the MaxSAT instance. This presents a large increase of the potential design space for such algorithms. We describe some potential improvements from this insight and show that an implementation outperforms the state of the art algorithms on the set of instances from the latest MaxSAT evaluation.

## 3.12 Integrating Column Generation and Large Neighborhood Search

*Lucas Kletzander (TU Wien, AT)*


**Joint work of** Lucas Kletzander, Tommaso Mannelli Mazzoli, Nysret Musliu, Pascal Van Hentenryck
**Main reference** Lucas Kletzander, Tommaso Mannelli Mazzoli, Nysret Musliu, Pascal Van Hentenryck: "Integrating Column Generation and Large Neighborhood Search for Bus Driver Scheduling with Complex Break Constraints", CoRR, Vol. abs/2505.02485, 2025.
**URL** https://doi.org//10.48550/ARXIV.2505.02485

This talk presents a framework that integrates column generation with large neighborhood search (LNS) to solve large bus driver scheduling problems with complex break constraints. Branch and price (B&P) is an established technique for such problems, using set partitioning as the master problem, and a resource constrained shortest path problem (RCSPP) as the pricing problem. However, the complex constraints require several optimizations to solve the high-dimensional sub-problem, in particular to split the sub-problem into disjoint sub-problems depending on characteristics like the break scheme, using exponential arc throttling to keep the sub-problem size small for as long as possible, and the use of k-d trees to deal with the current pareto frontier. However, scaling to very large instances is still not possible with B&P. Therefore LNS is used to select subsets of the current solution which are then optimized using column generation (CG). Instead of restarting CG for every subset, a column storage is introduced to reuse columns for future subsets, and a background thread works on the restricted master problem over the whole set of columns. LNS provides state-of-the-art results on instances of all sizes, and the tighter integration shows significant benefits over the naive combination of the technologies.

## 3.13 Solving the Identifying Code Set Problem with Grouped Independent Support

*Anna Latour (TU Delft, NL)*


**Joint work of** Anna L. D. Latour, Arunabha Sen, Kuldeep S. Meel
**Main reference** Anna L. D. Latour, Arunabha Sen, Kuldeep S. Meel: "Solving the Identifying Code Set Problem with Grouped Independent Support", in Proc. of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, pp. 1971–1978, ijcai.org, 2023.
**URL** https://doi.org//10.24963/IJCAI.2023/219

An important problem in network science is finding an optimal placement of sensors in nodes in order to uniquely detect failures in the network. This problem can be modelled as an identifying code set (ICS) problem, introduced by Karpovsky et al. in 1998. The ICS problem aims to find a cover of a set S, such that the elements in the cover define a unique signature for each of the elements of S, and to minimise the cover's cardinality. In this work, we study a generalised identifying code set (GICS) problem, where a unique signature must be found for each subset of S that has a cardinality of at most k (instead of just each element of S). The concept of an independent support of a Boolean formula was introduced by Chakraborty et al. in 2014 to speed up propositional model counting, by identifying a subset of variables whose truth assignments uniquely define those of the other variables. In this work, we introduce an extended version of independent support, grouped independent support (GIS), and show how to reduce the GICS problem to the GIS problem. We then propose a new solving method for finding a GICS, based on finding a GIS. We show that the prior state-of-the-art approaches yield integer-linear programming (ILP) models whose sizes grow exponentially with the problem size and k, while our GIS encoding only grows polynomially with the problem size and k. While the ILP approach can solve the GICS problem on networks of at most 494 nodes, the GIS-based method can handle networks of up to 21 363 nodes; a $\tilde{4}0\times$ improvement. The GIS-based method shows up to a $520\times$ improvement on the ILP-based method in terms of median solving time. For the majority of the instances that can be encoded and solved by both methods, the cardinality of the solution returned by the GIS-based method is less than 10% larger than the cardinality of the solution found by the ILP method.

## 3.14 Lower Bound in Branch-and-Bound (BnB) MaxSAT Solvers

*Chu Min Li (University of Amiens, FR)*


**Joint work of** Shuolin Li, Chu-Min Li, Jordi Coll, Djamal Habet, Felip Manya
**Main reference** Shuolin Li, Chu-Min Li, Jordi Coll, Djamal Habet, Felip Manyà: "Improving the Lower Bound in Branch-and-Bound Algorithms for MaxSAT", in Proc. of the AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 – March 4, 2025, Philadelphia, PA, USA, pp. 11272–11281, AAAI Press, 2025.
**URL** https://doi.org//10.1609/AAAI.V39I11.33226

The MaxSAT problem is an optimization version of the satisfiability problem (SAT). A tight lower bound (LB) on the number of falsified soft clauses in a MaxSAT solution is crucial for the efficiency of Branch-and-Bound (BnB) MaxSAT solvers. To compute an LB, modern BnB solvers detect disjoint inconsistent subsets of soft clauses, called cores, using

unit propagation. A notable feature of these solvers is that soft clauses belonging to already detected cores cannot be reused to detect additional cores, limiting the number of cores that can be detected. In this paper, we propose an unlocking mechanism that allows the reuse of soft clauses in already detected cores while ensuring the soundness of LB. Experimental results show that this unlocking mechanism consistently improves the performance of a state-of-the-art BnB solver. In addition, it allowed us to win the first two places in the exact unweighted category of the MaxSAT Evaluation 2024.

## 3.15 Accelerating Column Generation via Template Pricing

*Luke Marshall (Microsoft Research – Redmond, US)*


**Joint work of** Luke Marshall, Santanu Dey, Prachi Shah

Column Generation (CG) is an iterative algorithm effective in solving large-scale integer programming problems. It often relies on the fast re-optimization of the primal simplex algorithm for suitable performance; however, CG is notorious for convergence issues (with degenerate problems).

Although there is much research on stabilization techniques to avoid these issues, I'll introduce a new approach "Template Pricing" that can converge orders of magnitude faster. I'll illustrate with a simple example and give insights why it works so well (with computational results). Specifically, the choice of "what" columns to add in each iteration can make a significant impact on re-optimization performance.

## 3.16 My point of view on BDD/ZDD, SAT, and PB techniques for constraint optimization

*Shin-ichi Minato (Kyoto University, JP)*


**Joint work of** Shin-ichi Minato, Jun Kawahara, Mutsunori Banbara, Takashi Horiyama, Ichigaku Takigawa, Yutaro Yamaguchi
**Main reference** Shin-ichi Minato, Jun Kawahara, Mutsunori Banbara, Takashi Horiyama, Ichigaku Takigawa, Yutaro Yamaguchi: "Fast enumeration of all cost-bounded solutions for combinatorial problems using ZDDs", Discrete Applied Mathematics, Vol. 360, pp. 467–486, 2025.
**URL** https://doi.org//10.1016/j.dam.2024.10.003

Recently I'm interested in integrating the techniques for enumeration, optimization, and satisfiability. A task of constraint optimization is strongly related to those techniques, to generate all cost-bounded solutions satisfying a given combinatorial constraint. In this talk, I will review a classical SAT-based constraint optimization method using BDDs, and then present the method of enumerating all cost-bounded solutions using ZDDs. I would like to discuss a future direction how to collaborate SAT/MaxSAT techniques, DD-based techniques and ILP/PB techniques.

## 3.17 TT-Open-WBO-Inc: The SAT Engine of Modern Anytime MaxSAT

*Alexander Nadel (Technion – Haifa, IL & NVIDIA – Yokneam, IL)*


**Main reference** Alexander Nadel: "TT-Open-WBO-Inc: An Efficient Anytime MaxSAT Solver", J. Satisf. Boolean Model. Comput., Vol. 15(1), pp. 1–7, 2024.
**URL** https://doi.org//10.3233/SAT-231504

MaxSAT extends the cornerstone NP-complete SAT problem from decision to the optimization of a linear objective. It has a wide range of applications in AI, CAD, planning, scheduling, and beyond. Many of these applications – especially in industry – require anytime solvers, which continuously produce improving solutions. This talk presents TT-Open-WBO-Inc, the de-facto SAT engine of modern anytime MaxSAT solvers. TT-Open-WBO-Inc was the SAT component in the winners of the last three MaxSAT Evaluations in all four anytime categories, differing only in their local search preprocessors. TT-Open-WBO-Inc's efficiency stems from effective problem approximations (via the BMO or Mrs. Beaver algorithms) combined with both classical local search and SAT-based local search (the Polosat algorithm). We will present the genealogy of TT-Open-WBO-Inc and devote most of the talk to its underlying algorithms.

## 3.18 SpotIT: Evaluating Text-to-SQL Evaluation with Formal Verification

*Nina Narodytska (VMware Research – Palo Alto, US)*


**Joint work of** Nina Narodytska, Rocky Klopfenstein, Yang He, Andrew Tremante, Yuepeng Wang, Haoze Wu

Text-to-SQL plays a central role in building natural language interfaces that enable non-expert users to query structured data sources. Given a natural language query N to a database, the goal is to generate a SQL query that retrieves the data requested by N. Due to its practical importance, a large number of Text-to-SQL frameworks have been developed over the last two years in both industry and academia. However, the evaluation of these frameworks' performance has received much less attention. Most evaluation relies on a static testing approach that compares a generated SQL query against a gold SQL query produced by a human annotator for the same natural language query N. The key issue is that such static testing performs an overoptimistic evaluation, as queries can be equivalent by chance, just on these test instances.

In this work, we propose an alternative approach in which we search for a database that reveals discrepancies in the results returned by these SQL queries. We implemented our evaluation framework, which is powered by efficient formal verification techniques, and conducted a performance analysis of ten SOTA Text-to-SQL frameworks on BIRD datasets. Our results reveal that their accuracy is significantly lower – by up to 15% – compared to the results reported by the static testing approach. We also perform a detailed analysis of the failure cases and provide iusefull insights about shortcomings of the benchmark datasets.

## 3.19 Certified Implicit Hitting Set Solving for Pseudo-Boolean Optimization

*Jakob Nordström (University of Copenhagen, DK & Lund University, SE)*


**Joint work of** Jakob Nordström, Benjamin Bogø, Xiamin Chen, Wietze Koops, Pinyan Lu, Marc Vinyals, Qingzhao Wu

Implicit hitting set (IHS) methods work well for SAT-based and pseudo-Boolean optimization, but have lacked proof logging due to the use of mixed integer programming (MIP) solvers for the hitting set subproblems. We replace MIP with a combination of local search and pseudo-Boolean solving, yielding the first fully certified IHS approach for pseudo-Boolean optimization with feasible proof generation and verification overhead. Our ongoing work focuses on improving the hitting-set phase and integrating it more tightly in the overall solving process, and on identifying MIP features crucial for efficient IHS solving.

## 3.20 Symbolic Conflict Analysis in Pseudo-Boolean Optimization

*Albert Oliveras (UPC Barcelona Tech, ES)*


**Joint work of** Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, Rui Zhao
**Main reference** Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, Rui Zhao: "Symbolic Conflict Analysis in Pseudo-Boolean Optimization", in Proc. of the 28th International Conference on Theory and Applications of Satisfiability Testing, SAT 2025, Glasgow, Scotland, August 12-15, 2025, LIPIcs, Vol. 341, pp. 23:1–23:18, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.
**URL** https://doi.org//10.4230/LIPICS.SAT.2025.23

In the the last two decades, a lot of effort has been devoted to the development of satisfiability-checking tools for a variety of SAT-related problems. However, most of these tools lack optimization capabilities. That is, instead of finding any solution, one is sometimes interested in a solution that is best according to some criterion.

Pseudo-Boolean solvers can be used to deal with optimization by successively solving a series of problems that contain an additional pseudo-Boolean constraint expressing that a better solution is required. A key point for the success of this simple approach is that lemmas that are learned for one problem can be reused for subsequent ones.

In this talk we go one step further and show how, by using a simple symbolic conflict analysis procedure, not only can lemmas be reused between problems but also strengthened, thus further pruning the search space traversal. In addition, we show how this technique automatically allows one to infer upper bounds in maximization problems, thus giving an estimation of how far the solver is from finding an optimal solution. Experimental results with our PB solver reveal that (i) this technique is indeed effective in practice, providing important speedups in problems where several solutions are found and (ii) on problems with very few solutions, where the impact of our technique is limited, its overhead is negligible.

## 3.21 Tutorial on Dantzig-Wolfe decomposition, column generation, and branch-price-and-cut

*Elina Rönnberg (Linköping University, SE)*


**Joint work of** Stephen J. Maher, Elina Rönnberg
**Main reference** Stephen J. Maher, Elina Rönnberg: "Integer programming column generation: accelerating branch-and-price using a novel pricing scheme for finding high-quality solutions in set covering, packing, and partitioning problems", Math. Program. Comput., Vol. 15(3), pp. 509–548, 2023.
**URL** https://doi.org//10.1007/S12532-023-00240-W

In optimisation, decomposition means making a reformulation of a problem into a set of simpler problems that are typically solved by an iterative scheme to produce a solution to the original problem. The purpose is to distribute the computational burden of the original problem onto the simpler ones in a way that pays off with respect to total solution time. Critical for obtaining this is to exploit problem structure in the decomposition and to design an efficient solution scheme for the simpler problems.

In MIP, having a strong formulation matters. Dantzig-Wolfe decomposition yields an extended formulation in a higher-dimensional space that is at least as strong as the original one, and sometimes it is very strong. This strength, however, comes at the cost of a very high-dimensional representation, so that the extended formulation cannot be expressed explicitly. In the context of column generation, which is used for solving linear programs, the extended formulation is called the master problem. Instead of representing the full problem, one uses a restricted master problem that includes only a small subset of the variables. By using a pricing problem, the master problem is iteratively extended to containing an LP optimal solution. To find an optimal integer solution, the standard approach is to use branch-price-and-cut, where column generation is integrated into branch-and-bound.

In this tutorial, the basic principles of Dantzig-Wolfe decomposition, column generation, and branch-price-and-cut are introduced. This is continued with a discussion on our ongoing work on optimality conditions and pricing for integrality, highlighting why I believe these to be interesting areas of future research.

## 3.22 Quantum computing for discrete optimization: A glimpse into three technologies

*Philine Schiewe (Aalto University, FI)*


**Joint work of** Alexey Bochkarev, Raoul Heese, Sven Jäger, Philine Schiewe, Anita Schöbel
**Main reference** Alexey Bochkarev, Raoul Heese, Sven Jäger, Philine Schiewe, Anita Schöbel: "Quantum computing for discrete optimization: A highlight of three technologies", European Journal of Operational Research, Vol. 329(3), pp. 747–766, 2026.
**URL** https://doi.org//10.1016/j.ejor.2025.07.063

Quantum optimization has emerged as a promising frontier of quantum computing, providing novel numerical approaches to mathematical optimization problems. In this presentation, we aim to provide an initial intuition for quantum-powered methods in the context of discrete optimization. To this end, we consider three quantum-powered optimization approaches that make use of different types of quantum hardware available on the market. To illustrate these approaches, we solve three classical optimization problems: the Traveling Salesperson

Problem, Weighted Maximum Cut, and Maximum Independent Set. We attempt to provide an intuition behind each approach, describe the corresponding high-level workflow, and highlight crucial practical considerations. In particular, we emphasize the importance of problem formulations and device-specific configurations, and their impact on the amount of resources required for computation (where we focus on the number of qubits). These points are illustrated with a series of experiments on three types of quantum computers: a neutral atom machine from QuEra, a quantum annealer from D-Wave, and gate-based devices from IBM.

## 3.23 Weighted CP and related frameworks: soft arc consistency and bounds

*Thomas Schiex (INRA – Castanet-Tolosan, FR)*


**Joint work of** Thomas Schiex, George Katsirelos, Simon de Givry, Pierre Montalbano, Martin Cooper, Tomas Werner
**Main reference** Martin C. Cooper, Simon de Givry, Martí Sánchez-Fibla, Thomas Schiex, Matthias Zytnicki, Tomás Werner: "Soft arc consistency revisited", Artif. Intell., Vol. 174(7-8), pp. 449–478, 2010.
**URL** https://doi.org//10.1016/J.ARTINT.2010.02.001

In Weighted Constraint Programming, constraints are replaced by infinite-valued cost functions that contribute to the definition of both feasibility and a criterion. In this talk, the relations of this "network of cost functions" model with MaxSAT, QP and (01)LP are explored and the so-called "local polytope" shown to provide lower bounds The extension of arc consistency to this setting strictly generalizes usual (G)AC and is shown to provide dual feasible solutions of this local polytope using efficient combinatorial algorithms. This added understanding paves the way to the definition of algorithms enforcing soft local consistencies on global cost functions, when they have suitable LP formulations. Experiments on the QAPLib, using the AllDifferent constraint, handled as a "Linear assignement Problem", show the interest of this approach.

## 3.24 Introduction to Lazy Clause Generation

*Peter J. Stuckey (Monash University – Caulfield, AU)*


**Joint work of** Peter J. Stuckey, Olga Ohrimenko, Michael Codish
**Main reference** Olga Ohrimenko, Peter J. Stuckey, Michael Codish: "Propagation via lazy clause generation", Constraints An Int. J., Vol. 14(3), pp. 357–391, 2009.
**URL** https://doi.org//10.1007/S10601-008-9064-X

Finite domain propagation solvers effectively represent the possible values of variables by a set of choices which can be naturally modelled as Boolean variables. In this introduction we describe how to mimic a finite domain propagation engine, by mapping propagators into clauses in a SAT solver. This immediately results in strong nogoods for finite domain propagation. But a naive static translation is impractical except in limited cases. We show how to convert propagators to lazy clause generators for a SAT solver. The resulting system introduces flexibility in modelling since variables are modelled dually in the propagation engine and the SAT solver. The approach has proven to be the state of the art approach

to CP solving. We show how we can improve the straightforward implementation to lazy create the Boolean representation of integer variables; how we can extend conflict analysis to generate stronger, more reusable explanations, and how the language of learning greatly effects the method.

## 3.25 Tutorial: Optimization in SMT

*Nestan Tsiskaridze (Stanford University, US)*



Satisfiability Modulo Theories (SMT) has become a central technology in formal methods, enabling expressive reasoning across domains such as verification, synthesis, security, planning, and beyond. In recent years, Optimization Modulo Theories (OMT) has emerged as a powerful extension of SMT that allows not only deciding satisfiability but also optimizing cost functions over models. This tutorial offers a comprehensive introduction to optimization in the SMT setting.

We begin with the foundations of OMT, introducing key techniques. Special emphasis is placed on the evolution of OMT – from its origins in early extensions of SMT solving, through theory- and objective-specific and symbolic approaches, to the recently proposed Generalized OMT (GOMT) framework that unifies all single and multi-objective optimization and arbitrary theory combinations.

The tutorial aims to equip participants with a conceptual and practical understanding of OMT: how it works, why it matters, and how it continues to expand the reach of SMT/OMT technology into new application areas.

## 3.26 Decision Diagrams for Discrete Optimization

*Willem-Jan Van Hoeve (Carnegie Mellon University – Pittsburgh, US)*


**Main reference** Willem-Jan van Hoeve: "An Introduction to Decision Diagrams for Optimization", in Tutorials in Operations Research: Smarter Decisions for a Better World pp. 117–145, 2024.
**URL** https://doi.org//10.1287/educ.2024.0276

Over the last decade, decision diagram-based optimization has emerged as a novel approach to solving discrete optimization problems. We provide an overview of this new methodology, focusing on three computational paradigms: (1) stand-alone decision diagram-based solvers, (2) integration into constraint programming, and (3) integration into integer linear programming. We discuss applications including graph theoretic problems, scheduling, and vehicle routing. In particular, combining decision diagrams with network flow theory – via a process called "column elimination" – has resolved previously unsolved benchmark instances for problems such as vehicle routing with time windows, pickup-and-delivery with time windows, and graph multi-coloring. These advancements highlight the potential of decision diagram-based optimization as a powerful tool for addressing complex optimization challenges across domains.

### 3.27 Tutorial: Optimization in CP

*Hélène Verhaeghe (UC Louvain, BE)*



Constraint Programming (CP) targets combinatorial (optimization) problems. CP solvers require the problem to be modeled using variables (and their domains) and constraints (and an objective). In CP, variables have usually boolean domains or finite integer domains, but can have other special domains (set, graphs, sequences,...). The constraints in CP are arithmetic, logical or global constraints. Global constraints represent a relation between a set of variables (which can vary). The CP solver utilizes dedicated algorithms, known as propagators, for each constraint. They are responsible for filtering invalid values from the domains, based on reasoning corresponding to the constraints. These propagators are coordinated by the fixpoint, which decides which needs to be called. The fixpoint is called during the construction of the search space to prune invalid sub-trees. The search tree is built following the search, a heuristic selected by the user. CP is modular and can adapt to various formulations of the problems.

## 4 Working groups

### 4.1 How the optimization communities can work better together

*Peter J. Stuckey (Monash University – Caulfield, AU) and Luke Marshall (Microsoft Research – Redmond, US)*



The group discussion focused on how communities working in CP, SAT, MaxSAT, MIP, SMT, decision diagrams, etc. can collaborate more effectively.

One concern was that the neighboring solver communities still operate in silos due to a lack of a shared incremental modelling interface. SAT and MIP have stable low-level cores (clauses, linear constraints) and widely used APIs, while CP has a diverse collection of global constraints with varying propagation strengths and no common, structured, in-memory layer to expose them. FlatZinc was acknowledged as useful for interchange, but it flattens away intent, is not incremental, and cannot serve as a modern programmatic API with callbacks, assumptions, or partial additions. There was not enough CP solver implementors in attendance to acertain if this has sufficient support, but it is something that should be discussed in the CP community.

Decomposition methods (Benders, column generation, implicit hitting set approaches) were highlighted as useful approaches to build hybrid systems. Decision diagrams were viewed as a potential way to communicate the structural relationships between variables (i.e., bounds, conflict graphs etc.) with the various solvers. However, there is currently no agreed format (text or binary) for sharing them. Similarly, while SAT and pseudo-Boolean solvers have mature proof logging, CP lacks lightweight, checkable certificates for global propagations, limiting reproducibility, post-mortem analysis, and learning opportunities. Also, many MIP solvers will likely never include proof logging due to lack of commercial interest. The idea of having a library of encoding tools to map decision diagrams to CP, SAT, MIP seems like a valuable tool to allow us to share complex information between paradigms.

The group advocated for a curated set of cross-community benchmark instances that can compare solver methods, i.e., illustrate exponential separations, pathological behaviors, or technology advantages. They also emphasized structured synthetic generators with tunable parameters to enable controlled empirical studies rather than relying on ad hoc or purely random sets. In addition, they identified that encoding quality should not be overlooked. CSPlib could be used as a starting point for this, since it already includes curated CP models, and each technology can probably be applied. Of course finding volunteers to take this on is a challenge

Parallel and portfolio solving emerged as another cross-cutting topic. Recent SAT parallelism has shown surprising near-linear gains, suggesting untapped potential. Open questions center on what to share (bounds, cuts, symmetry info, nogoods, structural summaries), while being mindful of communication costs. Decision diagram partitioning in branch and bound was mentioned as a promising natural workload splitter.

Finally, there was enthusiasm for solver introspection and learning: mining proof traces or search logs to attribute progress to strategies, guiding restart policies, feature extraction for algorithm selection, and predicting which paradigm or encoding will excel based on structural instance features.

# 5 Panel discussions

## 5.1 Challenges and opportunities for the next 10 years

*Ambros Gleixner (HTW – Berlin, DE), Alexey Ignatiev (Monash University – Clayton, AU), Ciaran McCreesh (University of Glasgow, GB), Thomas Schiex (INRA – Castanet-Tolosan, FR), and Christine Solnon (INSA Lyon / Inria, FR)*



The main goal of the panel is to discuss key technical directions and challenges for the coming decade in CP, SAT/MaxSAT, SMT and MIP, and identify synergy between these research areas and other areas that can promote fast development and adoption of the technology. The following schemes were discussed.

**Hybrid methods and efficiency scheme.** Optimization techniques span over a large number of paradigms, each of which has complementary strengths. For example, local search provides fast but possibly far from optimal solutions while complete search guarantees finding optimal solutions but is computationally demanding. However, with modern computational resources, hybrid search methods are increasingly viable paradigms that can take advantage of these resources, e.g. memory intensive methods like breadth-first search or decision diagrams. A recurring theme was the need to prioritize fast, high-quality solutions over provable optimality, particularly in real-world applications where a tradeoff between speed and suboptimal solutions is acceptable. Another important point is to take advantage of GPUs that are increasingly available. It will require change of the algorithms as existing algorithms are not well suited for GPUs. So, it is an interesting direction to pursue. Finally, the integration of CP's propagation mechanisms into decision diagrams and dynamic programming frameworks was proposed as a promising direction for solver interoperability and performance enhancement.

**Modeling and adoption scheme.** Modeling remains a central concern. The users might still find it challenging to express their problems even in the user friendly modeling languages like MiniZinc. Direct modeling to SAT or MIP paradigms is feasible for advanced users only. As problem complexity grows, e.g. multi objective optimization, interactive optimization, incremental solving, it is even more important for the community to provide the user a simple way to deal with challenging problems. One of the solutions offered was to use LLMs as an intermediate layer between the user and the model. The idea is that the user specifies the problem in natural language and using LLMs we translate it into a model. The process is iterative and requires user guidance and verification of correctness. This was identified as a promising approach. Finally, the panel discussed strategic directions for community investment, including shared benchmark libraries, accessible educational resources, and scalable solver infrastructure. We also need to invest in teaching optimization technology as it does help with adoption.

**Verification and explainability scheme.** The panel emphasized that verification and explanability are very important topics for the next decade. Proof logging is the central technique that allows verifying correctness of the solver output that has received significant attention in recent years. Indeed, it is an area under active development. Explanability is another key point as giving the user a solution (or no solution) answer might not be very satisfying. Explanability techniques need more development, starting from defining what exactly explanability is in a given context, the language of explanability and computing explanations efficiently. The panel advocated for modular solver architectures that facilitate cross-paradigm innovation and re-emphasized the importance of proof logging to ensure correctness and reproducibility.

**Priority areas.** If given substantial funding, the panel identified that priorities are would include parallelization, deployment in everyday decision-making contexts, and broader accessibility.

## Participants


- Florent Avellaneda
UQAM – Montreal, CA
- Armin Biere
Universität Freiburg, DE
- Bart Bogaerts
KU Leuven, BE
- Toby Davies
Google – Pyrmont, AU
- Frédéric Didier
Google – Paris, FR
- Katalin Fazekas
TU Wien, AT
- Pierre Flener
Uppsala University, SE
- Bishwamittra Ghosh
MPI-SWS – Kaiserslautern, DE
- Ambros Gleixner
HTW – Berlin, DE
- Tias Guns
KU Leuven, BE
- Alexey Ignatiev
Monash University – Clayton, AU
- Hannes Ihalainen
University of Helsinki, FI
- Christoph Jabs
University of Helsinki, FI
- Matti Järvisalo
University of Helsinki, FI
- George Katsirelos
INRAE – Palaiseau, FR
- Zeynep Kiziltan
University of Bologna, IT
- Lucas Kletzander
TU Wien, AT
- Anna Latour
TU Delft, NL
- Chu Min Li
University of Amiens, FR
- Luke Marshall
Microsoft Research – Redmond, US
- Ciaran McCreesh
University of Glasgow, GB
- Gioni Mexi
Zuse-Institut Berlin, DE
- Shin-ichi Minato
Kyoto University, JP
- Alexander Nadel
Technion – Haifa, IL & NVIDIA – Yokneam, IL
- Nina Narodytska
VMware Research – Palo Alto, US
- Robert Nieuwenhuis
Barcelona, ES
- Jakob Nordström
University of Copenhagen, DK & Lund University, SE
- Andy Oertel
Lund University, SE
- Albert Oliveras
UPC Barcelona Tech, ES
- Anastasia Paparrizou
CNRS – Montpellier, FR
- Elina Rönnberg
Linköping University, SE
- Andre Schidler
Universität Freiburg, DE
- Philine Schiewe
Aalto University, FI
- Thomas Schiex
INRA – Castanet-Tolosan, FR
- Mohamed Siala
LAAS – Toulouse, FR
- Christine Solnon
INSA Lyon / Inria, FR
- Peter J. Stuckey
Monash University – Caulfield, AU
- Nestan Tsiskaridze
Stanford University, US
- Willem-Jan Van Hoeve
Carnegie Mellon University – Pittsburgh, US
- Hélène Verhaeghe
UC Louvain, BE
- Allen Z. Zhong
Monash University – Clayton, AU