

Theory and Practice of SAT and Combinatorial Solving

Olaf Beyersdorff^{*1}, Armin Biere^{*2}, Vijay Ganesh^{*3},
Jakob Nordström^{*4}, and Andy Oertel^{†5}

1 Friedrich-Schiller-Universität Jena, DE. olaf.beyersdorff@uni-jena.de

2 Universität Freiburg, DE. armin.biere@gmail.com

3 University of Waterloo, CA. vijay.ganesh@uwaterloo.ca

4 University of Copenhagen, DK & Lund University, SE. jn@di.ku.dk

5 Lund University, SE. andy.oertel@cs.lth.se

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 22411 “*Theory and Practice of SAT and Combinatorial Solving*”. The purpose of this workshop was to explore the Boolean satisfiability (SAT) problem, which plays a fascinating dual role in computer science. By the theory of *NP*-completeness, this problem captures thousands of important applications in different fields, and a rich mathematical theory has been developed showing that all these problems are likely to be infeasible to solve in the worst case. But real-world problems are typically not worst-case, and in recent decades exceedingly efficient algorithms based on so-called conflict-driven clause learning (CDCL) have turned SAT solvers into highly practical tools for solving large-scale real-world problems in a wide range of application areas. Analogous developments have taken place for problems beyond *NP* such as SAT-based optimization (MaxSAT), pseudo-Boolean optimization, satisfiability modulo theories (SMT) solving, quantified Boolean formula (QBF) solving, constraint programming, and mixed integer programming, where the conflict-driven paradigm has sometimes been added to other powerful techniques.

The current state of the art in combinatorial solving presents a host of exciting challenges at the borderline between theory and practice. Can we gain a deeper scientific understanding of the techniques and heuristics used in modern combinatorial solvers and why they are so successful? Can we develop tools for rigorous analysis of the potential and limitations of these algorithms? Can computational complexity theory be extended to shed light on real-world settings that go beyond worst case? Can more powerful methods of reasoning developed in theoretical research be harnessed to yield improvements in practical performance? And can state-of-the-art combinatorial solvers be enhanced to not only solve problems, but also provide verifiable proofs of correctness for the solutions they produce?

This workshop gathered leading applied and theoretical researchers working on SAT and combinatorial optimization more broadly in order to stimulate an exchange of ideas and techniques. We see great opportunities for fruitful interplay between theory and practice in these areas, as well as for technology transfer between different paradigms in combinatorial optimization, and our assessment is that this workshop demonstrated very convincingly that a more vigorous interaction has potential for major long-term impact in computer science, as well for applications in industry.

Seminar October 9–14, 2022 – <http://www.dagstuhl.de/22411>

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Complexity theory and logic; Theory of computation → Logic and verification; Theory of computation → Automated reasoning; Theory of computation → Constraint and logic programming; Theory of computation → Discrete optimization; Security and privacy → Logic and verification

Keywords and phrases Boolean satisfiability (SAT), SAT solving, computational complexity, proof complexity, combinatorial solving, combinatorial optimization, constraint programming, mixed integer linear programming

Digital Object Identifier 10.4230/DagRep.12.10.84

* Editor / Organizer

† Editorial Assistant / Collector



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Theory and Practice of SAT and Combinatorial Solving, *Dagstuhl Reports*, Vol. 12, Issue 10, pp. 84–105
Editors: Olaf Beyersdorff, Armin Biere, Vijay Ganesh, Jakob Nordström, and Andy Oertel



Dagstuhl Reports

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

1 Executive Summary

Olaf Beyersdorff

Armin Biere

Vijay Ganesh

Jakob Nordström

License © Creative Commons BY 4.0 International license
© Olaf Beyersdorff, Armin Biere, Vijay Ganesh, and Jakob Nordström

This event gathered leading researchers working on applied and theoretical aspects of the satisfiability (SAT) problem in areas like Boolean satisfiability (SAT) solving and proof complexity and computational complexity theory more broadly, as well as representatives from neighbouring areas such as, e.g., satisfiability modulo theories (SMT) solving, maximum satisfiability (MaxSAT) solving, pseudo-Boolean optimization, constraint programming, and mixed integer linear programming (MIP) on the applied side, and from other areas of computational complexity theory such as exact exponential-time algorithms and parameterized complexity on the theoretical side. This was meant to create an environment conducive to exchange of ideas and techniques between different fields of research. Among the goals of the workshop were to develop a better scientific understanding of real-world efficient computation in general and of the starkly different perspective between the theory and practice of *NP*-hard problems in particular, to explore new approaches for SAT and other challenging combinatorial problems that would have the potential to go beyond the current state of the art, and to stimulate a technology transfer between SAT and other related areas.

This workshop is part of a highly successful series starting at the Banff International Research Station (BIRS) in Canada in 2014, and with follow-up editions held at Schloss Dagstuhl in 2015 (Seminar 15171), at the Fields Institute in Toronto, Canada in 2016, and at the BIRS-affiliated Casa Matemática Oaxaca in Oaxaca, Mexico in 2018. After this fifth edition at Schloss Dagstuhl in October 2022, a sixth edition “*Satisfiability: Theory, Practice, and Beyond*” has already been organized at the Simons Institute for the Theory of Computing at UC Berkeley in April 2023 as part of an eponymous two-month scientific program.

Topic of the Workshop

What served as the point of departure of this workshop is one of the most significant problems in all of mathematics and computer science, namely that of proving logic formulas. This is a problem of immense importance both theoretically and practically. On the one hand, it is believed to be intractable in general, and deciding whether this is so is one of the famous million dollar Clay Millennium Problems (the *P* vs. *NP* problem). On the other hand, today so-called SAT solvers are routinely and successfully used to solve large-scale real-world instances in a wide range of application areas (such as hardware and software verification, electronic design automation, artificial intelligence research, cryptography, bioinformatics, and operations research, just to name a few examples).

During the last two decades there have been dramatic – and surprising – developments in SAT solving technology that have improved real-world performance by many orders of magnitude. However, while modern solvers can often handle formulas with millions of variables, there are also tiny formulas with just a few hundred variables that cause even the very best solvers to stumble. The fundamental question of when SAT solvers perform well or badly, and what underlying properties of the formulas influence performance, remains very

poorly understood. Other practical SAT solving issues, such as how to optimize memory management and how to exploit parallelization on modern multicore architectures, are even less well studied and understood from a theoretical point of view.

Perhaps even more surprisingly, the best SAT solvers today are still based on relatively simple methods from the early 1960s (though the introduction of so-called conflict-driven learning in the 1990s was a very important addition), searching for proofs in the so-called resolution proof system. Although other mathematical methods of reasoning are known that are much stronger than resolution in theory, in particular methods based on algebra (Gröbner bases) and geometry (cutting planes), attempts to harness the power of such methods have mostly failed to deliver significant improvements in practical performance for SAT solving. And while resolution is a fairly well-understood proof system, even very basic questions about these stronger algebraic and geometric methods remain wide open.

This is an interesting contrast to developments in neighbouring areas such as, e.g., constraint programming and mixed integer programming. There much more powerful methods of reasoning are successfully used to guide the search, but compared to SAT solving the attempts to employ conflict-driven learning have had much less of an impact. Also, while for SAT solvers it is at least possible to understand some aspects of their performance (by analysing proof systems such as resolution), a corresponding theoretical framework for constraint programming and mixed integer linear programming seems to be mostly missing.

In this workshop, we gathered leading researchers working on SAT and other challenging combinatorial optimization problems in order to stimulate an increased exchange of ideas between theoreticians and practitioners. As discussed above, previous editions of this workshop series at the Banff International Research Station, Schloss Dagstuhl, Fields Institute, and Casa Matemática Oaxaca have had a major impact on the involved communities and has helped to create bridges and stimulate the emergence of a joint research agenda. We are happy to report that the October 2022 workshop at Schloss Dagstuhl fully delivered on the expectation of serving as the valuable next step on this journey. During recent years we have already seen how computational complexity theory can shed light on the power and limitations on current and possible future techniques for SAT and other optimization problems, and that problems encountered on the applied side can spawn interesting new areas in theoretical research. We see great potential for continued interdisciplinary research at the border between theory and practice in this area, and believe that more vigorous interaction between practitioners and theoreticians could have major long-term impact in both academia and industry.

Goals of the Workshop

A strong case can be made for the importance of increased exchange between the two fields of SAT solving on the one hand and proof complexity (and more broadly computational complexity) on the other. Given how many questions that would seem to be of mutual interest, it is striking that the level of interaction had been so low until our workshop series started with the first two meetings in Banff in 2014 and Dagstuhl in 2015. Below, we outline some of the concrete questions that served as motivation for organizing our second Dagstuhl workshop in this series in 2022, and for broadening the scope from Boolean satisfiability to combinatorial solving and optimization in general. We want to stress that this inst is far from exhaustive, and we believe that one important outcome of the seminar was to uncover also other questions at the intersection of theoretical and applied research, and of different research areas within combinatorial solving and optimization.

What Makes Formulas Hard or Easy in Practice for Modern SAT Solvers?

The best SAT solvers known today are based on the DPLL procedure, augmented with optimizations such as conflict-driven clause learning (CDCL) and restart strategies. The propositional proof system underlying such algorithms, resolution, is arguably the most well-studied system in all of proof complexity.

Given the progress during the last decade on solving large-scale instances, it is natural to ask what lies behind the spectacular success of CDCL solvers at solving these instances. And given that there are still very small formulas that resist even the most powerful CDCL solvers, a complementary interesting question is if one can determine whether a particular formula is hard or tractable. Somewhat unexpectedly, very little turns out to be known about these questions.

In view of the fundamental nature of the SAT problem, and in view of the wide applicability of modern SAT solvers, this seems like a clear example of a question of great practical importance where the theoretical field of proof complexity could potentially provide useful insights. In particular, one can ask whether one could find theoretical complexity measures for formulas that would capture their practical hardness in some nice and clean way. Besides greatly advancing our theoretical understanding, answering such a question could also have applied impact in the longer term by clarifying the limitations, and potential for further improvements, of modern SAT solvers.

Can Proof Complexity Shed Light on Crucial SAT Solving Issues?

Understanding the hardness of proving formulas in practice is not the only problem for which more applied researchers would welcome contributions from theoretical computer scientists. Examples of some other possible practical questions that would benefit from a deeper theoretical understanding follow below.

- Firstly, we would like to study the question of memory management. One major concern for clause learning algorithms is to determine how many clauses to keep in memory. Also, once the algorithm runs out of the memory currently available, one needs to determine which clauses to throw away. These questions can have huge implications for performance, but are poorly understood.
- In addition to clause learning, the concept of restarts is known to have decisive impact on the performance on modern CDCL solvers. It would be nice to understand theoretically why this is so. The reason why clause learning increases efficiency greatly is clear – without it the solver will only generate so-called tree-like proofs, and tree-like resolution is known to be exponentially weaker than general resolution. However, there is still ample room for improvement of our understanding of the role of restarts and what are good restart strategies.
- Given that modern computers are multi-core architectures, a highly topical question is whether this (rather coarse-grained) parallelization can be used to speed up SAT solving. While there are some highly successful attempts in parallelizing SAT for solving theoretical problems in, e.g., extremal combinatorics, the speed-ups obtained for more applied problems are rather modest or sometimes non-existent. This is a barrier for further adoption of SAT technology already today, and will become a more substantial problem as thousands of cores and cloud computing are becoming the dominant computing platforms in the future. A theoretical understanding of if and how SAT can be parallelized will be essential to develop new parallelization strategies to adapt SAT to this new computing paradigm.

We believe that progress on any of these questions has the potential of influencing the further development of both theoretical and applied research, and to stimulate a further cross-pollination between these two areas.

Can we build SAT Solvers based on Stronger Proof Systems than Resolution?

Although the performance of modern CDCL SAT solvers is impressive, it is nevertheless astonishing, not to say disappointing, that the state-of-the-art solvers are still based on resolution. This method lies close to the bottom in the hierarchy of propositional proof systems, and there are many other proof systems based on different forms of mathematical reasoning that are known to be strictly stronger. Some of these appear to be natural candidates on which to build stronger SAT solvers than those using CDCL.

In particular, proof systems such as polynomial calculus (based on algebraic reasoning) and cutting planes (based on geometry) are known to be exponentially more powerful than resolution. While there has been some work on building SAT solvers on top of these proof systems, progress has been fairly limited. We believe it would be fruitful to discuss what the barriers are that stops us from building stronger algebraic or geometric SAT solvers, and what is the potential for future improvements. An important part of this work would seem to be to gain a deeper theoretical understanding of the power and limitations of these proof methods. Here there are a number of fairly long-standing open theoretical questions. At the same time, only in the last couple of years proof complexity has made substantial progress, giving hope that the time is ripe for decisive break-throughs in these areas.

Can Technology Be Transferred Between Different Combinatorial Optimization Paradigms?

Continuing the discussion of stronger methods of reasoning, it is natural to ask whether techniques from e.g., constraint programming (CP) and mixed integer linear programming (MIP) could be imported into SAT solving and vice versa. At a high level, the main loop of combinatorial solvers in all of these paradigms consists of two phases:

- During the *search phase*, the solver makes *decisions* (guesses) about variable assignments and propagates “obvious” consequences until it either finds a solution or discovers a violated constraint (a *conflict*).
- In case of a conflict, during the *backtracking phase* the solver analyses what went wrong and reverses some decision(s) to remove the violation, after which it switches to search again.

For CP and MIP solvers, significant effort is spent during the search phase on making intelligent decisions and deriving (sometimes not so obvious) consequences. In comparison, the backtracking phase is not so sophisticated. For SAT and pseudo-Boolean (PB) solvers it is exactly the other way around. The decisions during the search phase are done quite naively, and since the constraints are relatively simple, propagations cannot be too strong and are fairly easy to detect. Once a conflict is reached, however, an elaborate *conflict analysis* algorithm combines the constraints involved in this conflict to *learn* a new, globally valid constraint that is added to the input formula.

Could it be possible to make more efficient use of conflict analysis in MIP and CP solving? CDCL-style analysis has already been tried with some success, but since this approach boils down to reasoning with clauses it is provably exponentially weaker than what pseudo-Boolean techniques can offer. In the other direction, a quite tempting proposition is to integrate into SAT and PB solvers the vastly stronger propagation used in CP and MIP. Another change

of perspective would be to turn *core-guided MaxSAT solving* into a general pseudo-Boolean optimization technique, using PB conflict analysis to extract better cores, and letting these cores serve as a heuristic for introducing new variables in the spirit of *extended resolution*.

Organization of the Workshop

The scientific program of the seminar consisted of 29 presentations. Among these there were 14 50-minute surveys of different core topics of the workshop. These talks occupied most of the morning schedule Monday-Thursday, and were intended to make sure that the diverse audience would have a bit of a common background for the more technical talks reporting on recent research projects. The list of survey talks and speakers were as follows:

- SAT: interactions between theory and practice (Olaf Beyersdorff)
- SAT and computational complexity theory (Ryan Williams)
- Proof complexity and SAT solving (Jakob Nordström)
- Efficient proof search in proof complexity, a.k.a. automatability (Susanna de Rezende)
- Satisfiability modulo theories (SMT) solving (Nikolaj Bjørner)
- Quantified Boolean formula (QBF) solving and proof complexity (Meena Mahajan)
- Constraint programming (Ciaran McCreesh)
- Mixed integer linear programming (Ambros Gleixner)
- Algebraic methods for circuit verification (Daniela Kaufmann)
- First-order theorem proving (Laura Kovacs and Martin Suda)
- Automated planning (Malte Helmert)
- Formally verified combinatorial solvers (Mathias Fleury)
- Certifying solvers with proof logging (Armin Biere)
- Formally verified proof checking for certifying solvers (Yong Kiam Tan)

The rest of the talks were 25-minute presentations on recent research of the participants. The time after lunch each day was left for self-organized collaborations and discussions, and there was no schedule on Wednesday afternoon.

Based on polling of participants before the seminar week, it was decided to have an open problem session on Thursday afternoon. The poll also asked whether a panel discussion should be organized, but the support for this idea was weaker, and several participants emphasized that the workshop program should not be too dense and that the evenings should be left free of any program. Therefore, the organizers decided not to have a panel discussion. As a nice contribution, some of the participants of the workshop organized a music night on the last evening of the workshop.

2 Table of Contents

Executive Summary

<i>Olaf Beyersdorff, Armin Biere, Vijay Ganesh, and Jakob Nordström</i>	85
-----------------------------------------------------------------------------------	----

Overview of Talks

Clause Redundancy and Preprocessing in Maximum Satisfiability <i>Jeremias Berg</i>	92
Theory and practice of SAT solving <i>Olaf Beyersdorff</i>	92
Trusting SAT Solvers <i>Armin Biere</i>	92
In introduction to SMT with Proofs <i>Nikolaj S. Bjørner</i>	93
On Symmetries and Certification <i>Bart Bogaerts</i>	93
CDCL vs resolution: the picture in QBF <i>Benjamin Böhm</i>	94
Theoretical Barriers for Efficient Proof Search (a Survey) <i>Susanna de Rezende</i>	94
On Design Decisions of Extending CDCL with External Propagators <i>Katalin Fazekas</i>	94
Discussion: How to combine and compare options in solvers? <i>Mathias Fleury and Armin Biere</i>	95
Verifying Solvers: How Much Do You Want to Prove? <i>Mathias Fleury</i>	95
Algorithmic Mixed Integer Programming: Between Exactness and Performance in Theory and Practice <i>Ambros M. Gleixner</i>	95
The Packing Chromatic Number of the Infinite Square Grid is 15 <i>Marijn J. H. Heule</i>	96
Pseudo-Boolean Optimization by Implicit Hitting Sets <i>Matti Järvisalo</i>	96
Exploring Algebraic Methods for Circuit Verification <i>Daniela Kaufmann</i>	97
First-Order Theorem Proving – Theory and Practice <i>Laura Kovács</i>	97
Towards a Deeper Understanding of Modern CDCL SAT Solvers <i>Chunxiao (Ian) Li, Jonathan Chung, Vijay Ganesh, Antonina Kolokolova, Alice Mu, Soham Mukherjee, and Marc Vinyals</i>	98
Quantified Boolean Formulas: (Solving and) Proof Complexity <i>Meena Mahajan</i>	98

How Constraint Programming Isn't Like SAT <i>Ciaran McCreesh</i>	98
Certification of Samplers <i>Kuldeep S. Meel</i>	99
Proof complexity and SAT solving <i>Jakob Nordström</i>	99
Certified CNF Translations for Pseudo-Boolean Solving <i>Andy Oertel</i>	100
Scalable optimization with SAT-based local improvement (SLIM) <i>Andre Schödler, Friedrich Slivovsky, and Stefan Szeider</i>	100
On the Use of SAT Solvers in a Modern ATP <i>Martin Suda</i>	101
The Last Mile in Trustworthy Automated Reasoning <i>Yong Kiam Tan</i>	101
A SAT Solver + Computer Algebra Attack on the Minimum Kochen-Specker Problem <i>Vijay Ganesh</i>	102
Theoretical limits of 1UIP Learning <i>Marc Vinyals</i>	102
Exponential separations using guarded extension variables <i>Emre Yolcu</i>	102
Evaluation by Participants	103
Participants	105

3 Overview of Talks

3.1 Clause Redundancy and Preprocessing in Maximum Satisfiability

Jeremias Berg (University of Helsinki, FI)

License © Creative Commons BY 4.0 International license
© Jeremias Berg

Joint work of Ihalainen, Hannes; Järvisalo, Matti; Berg, Jeremias

Main reference Hannes Ihalainen, Jeremias Berg, Matti Järvisalo: “Clause Redundancy and Preprocessing in Maximum Satisfiability”, in Proc. of the Automated Reasoning – 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings, Lecture Notes in Computer Science, Vol. 13385, pp. 75–94, Springer, 2022.

URL http://dx.doi.org/10.1007/978-3-031-10769-6_6

The study of clause redundancy in Boolean satisfiability (SAT) has proven significant in various terms, from fundamental insights into preprocessing and inprocessing to the development of practical proof checkers and new types of strong proof systems. I will present our recent work on liftings of the recently-proposed notion of propagation redundancy — based on a semantic implication relationship between formulas — in the context of maximum satisfiability (MaxSAT), where of interest are reasoning techniques that preserve optimal cost (in contrast to preserving satisfiability in the realm of SAT). We establish the strongest MaxSAT-lifting of propagation redundancy allows for changing in a controlled way the set of minimal correction sets in MaxSAT. This ability is key in succinctly expressing MaxSAT reasoning techniques and allows for obtaining correctness proofs in a uniform way for MaxSAT reasoning techniques very generally. I will also highlight some interesting directions for future work.

3.2 Theory and practice of SAT solving

Olaf Beyersdorff (Friedrich-Schiller-Universität Jena, DE)

License © Creative Commons BY 4.0 International license
© Olaf Beyersdorff

This talk provides a brief survey on the relations between proof complexity and SAT solving. What can proof complexity tell us about the strength and limitations of SAT solving? Why should practitioners be interested in proof complexity results and why should theorists study SAT solving? What have we achieved in the past 25 years and which problems remain open?

3.3 Trusting SAT Solvers

Armin Biere (Universität Freiburg, DE)

License © Creative Commons BY 4.0 International license
© Armin Biere

Many critical applications crucially rely on the correctness of SAT solvers. Particularly in the context of formal verification, the claim by a SAT solver that a formula is unsatisfiable corresponds to a safety or security property to hold, and thus needs to be trusted. In order to increase the level of trust an exciting development in this century was to let SAT solvers produce certificates, i.e., by tracing proofs of unsatisfiability, which can independently be

checked. In the last ten years this direction of research gained substantial momentum, e.g., solvers in the main track of the SAT competition are required to produce such certificates and industrial applications of SAT solvers require that feature too. In this talk we review this quarter of century of research in certifying the result of SAT solvers, discuss briefly alternatives, including testing approaches and verifying the SAT solver directly, mention exciting research on new proof systems produced in this context as well as how these ideas extend beyond formulas in conjunctive normal form.

3.4 In introduction to SMT with Proofs

Nikolaj S. Bjørner (Microsoft – Redmond, US)

License © Creative Commons BY 4.0 International license
© Nikolaj S. Bjørner

The talk provides an overview of selected current trends in SMT solving theories and techniques <https://z3prover.github.io/slides/proofs.html>. An active area of discussion in the SMT community is around proof formats for SMT solvers. I give an introduction to current approaches pursued in solvers such as Z3, CVC5, VeriT and SMTInterpol.

3.5 On Symmetries and Certification

Bart Bogaerts (VU – Brussels, BE)

License © Creative Commons BY 4.0 International license
© Bart Bogaerts

Joint work of Bogaerts, Bart; Devriendt, Jo; Gocht, Stephan; McCreesh, Ciaran; Nordström, Jakob
Main reference Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, Jakob Nordström: “Certified Symmetry and Dominance Breaking for Combinatorial Optimisation”, in Proc. of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 – March 1, 2022, pp. 3698–3707, 2022.

URL <https://ojs.aaai.org/index.php/AAAI/article/view/20283>

In this talk, we take a deep-dive in the fascinating world of symmetry handling for Boolean Satisfiability, by reviewing three main classes of techniques: static symmetry breaking, dynamic symmetry breaking, and (dynamic) symmetric learning. We focus on proof logging techniques that have been developed for these symmetry handling methods, and in particular study our recent symmetry handling methods built on VeriPB. We end with some open problems and challenges.

3.6 CDCL vs resolution: the picture in QBF

Benjamin Böhm (Friedrich-Schiller-Universität Jena, DE)

License © Creative Commons BY 4.0 International license
© Benjamin Böhm

Joint work of Peitl, Tomás; Beyersdorff, Olaf

Main reference Benjamin Böhm, Tomás Peitl, Olaf Beyersdorff: “Should Decisions in QCDCL Follow Prefix Order?”, in Proc. of the 25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel, LIPIcs, Vol. 236, pp. 11:1–11:19, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

URL <http://dx.doi.org/10.4230/LIPIcs.SAT.2022.11>

This talk will cover the relations between QBF resolution and QCDCL solving algorithms. Modelling QCDCL as proof systems we show that QCDCL and Q-Resolution are incomparable. We also introduce new versions of QCDCL that turn out to be stronger than the classic models. This talk is based on a couple of recent papers (joint with Olaf Beyersdorff and Tomas Peitl, which appeared in ITCS’21, SAT’21, SAT’22 and IJCAI’22).

3.7 Theoretical Barriers for Efficient Proof Search (a Survey)

Susanna de Rezende (Lund University, SE)

License © Creative Commons BY 4.0 International license
© Susanna de Rezende

The proof search problem is a central question in automated theorem proving and SAT solving. Clearly, if a propositional tautology F does not have a short (polynomial size) proof in a proof system P , any algorithm that searches for P -proofs of F will necessarily take super-polynomial time. But can proofs of “easy” formulas, i.e., those that have polynomial size proofs, be found in polynomial time? This question motivates the study of automatability of proof systems. In this talk, we give an overview of known non-automatability results, focusing on the more recent ones, and present some of the main ideas used to obtain them.

3.8 On Design Decisions of Extending CDCL with External Propagators

Katalin Fazekas (TU Wien, AT)

License © Creative Commons BY 4.0 International license
© Katalin Fazekas

Joint work of Katalin Fazekas, Armin Biere

Solving combinatorial problems often combines SAT solving with different reasoning techniques. An external propagator can interpret the partial assignment built by the SAT solver during search and, based on such different reasoning methods, can construct clauses that are propagating or conflicting under that assignment. The use of such external propagators allows to directly guide the search of the SAT solver into a preferred direction, which can make problem solving in several problem domains more efficient (consider e.g. dynamic symmetry breaking). However, both the efficient combination of external propagators with the complex features of modern SAT solvers (e.g. proof logging and inprocessing), and the theoretical understanding of such combined reasoning methods are open problems. This talk, based on current work in progress, presents some of the design decisions that must be considered when external propagation is combined with modern CDCL solvers. We describe

some challenges and formulate both practical and theoretical open questions about how to implement external propagation in CDCL in the presence of current SAT solver features such as proof logging, clause database reduction and inprocessing.

3.9 Discussion: How to combine and compare options in solvers?

Mathias Fleury (Universität Freiburg, DE) and Armin Biere (Universität Freiburg, DE)

License © Creative Commons BY 4.0 International license
© Mathias Fleury and Armin Biere

Joint work of Sakallah, Kareem; Biere, Armin; Fleury, Mathias

Comparing options between solvers is a complicated task. There are three main ways: runtime options (with the risk of not understanding requirements between features), compile-time option (with the issue of testing and making the code very complicated), or different versions of the source code. A second question is how to measure the performance without implementing the most advanced version. This (short) talk should serve as a basis for discussion on how to organize the development of a solver with various options.

3.10 Verifying Solvers: How Much Do You Want to Prove?

Mathias Fleury (Universität Freiburg, DE)

License © Creative Commons BY 4.0 International license
© Mathias Fleury

Joint work of Blanchette, Jasmin; Lammich, Peter; Weidenbach, Christoph; Fleury, Mathias

In this talk, I present the two main approaches to verify solvers: partial verification (usually bottom-up from code to the specification) and complete verification (usually top-down from the specification towards the code). The former approach present many similarities to verify checkers, whereas the latter starts with a full formalization of underlying algorithm. I compare the approaches and show where the main challenges are.

3.11 Algorithmic Mixed Integer Programming: Between Exactness and Performance in Theory and Practice

Ambros M. Gleixner (HTW - Berlin, DE)

License © Creative Commons BY 4.0 International license
© Ambros M. Gleixner

Joint work of Eifler, Leon; Gleixner, Ambros M.

Main reference Leon Eifler, Ambros M. Gleixner: “A computational status update for exact rational mixed integer programming”, *Math. Program.*, Vol. 197(2), pp. 793–812, 2023.

URL <http://dx.doi.org/10.1007/s10107-021-01749-5>


Today’s state-of-the-art solvers for the general class of mixed integer programs exhibit both exact and heuristic properties. Both aspects are crucial for their lasting relevance in academic and industrial practice. In this talk, we give an overview of methods implemented and successfully used in mixed-integer programming solvers and try to point out connections to satisfiability solving and pseudo-Boolean optimization. We conclude by outlining our efforts to address the ubiquitous use of floating-point arithmetic in virtually all fast mixed integer programming solvers and report advances in performant roundoff-error-free MIP solving with proof logging [1].

References

- 1 Leon Eifler and Ambros Gleixner. A computational status update for exact rational mixed integer programming. *Mathematical Programming*, 2022.

3.12 The Packing Chromatic Number of the Infinite Square Grid is 15

Marijn J. H. Heule (Carnegie Mellon University – Pittsburgh, US)

License  Creative Commons BY 4.0 International license
© Marijn J. H. Heule

Joint work of Marijn J. H. Heule, Bernardo Subercaseaux

A packing k -coloring of a graph $G = (V, E)$ is a mapping from V to $1, \dots, k$ such that any pair of vertices u, v that receive the same color c must be at distance greater than c in G . Arguably the most fundamental problem regarding packing colorings is to determine the packing chromatic number of the infinite square grid. Various works in the last 20 years improved the bounds of this problem. We finally solve it and show that the answer is 15. A crucial part of our solution is a novel encoding that reduces the runtime by a factor of 30. Moreover, we construct and validate a DRAT proof of unsatisfiability for the direct encoding of the problem. This proof includes the symmetry-breaking and reencoding techniques that we applied.

3.13 Pseudo-Boolean Optimization by Implicit Hitting Sets

Matti Järvisalo (University of Helsinki, FI)

License  Creative Commons BY 4.0 International license
© Matti Järvisalo

Joint work of Smirnov, Pavel; Berg, Jeremias; Järvisalo, Matti

Main reference Pavel Smirnov, Jeremias Berg, Matti Järvisalo: “Pseudo-Boolean Optimization by Implicit Hitting Sets”, in Proc. of the 27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021, LIPIcs, Vol. 210, pp. 51:1–51:20, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

URL <http://dx.doi.org/10.4230/LIPIcs.CP.2021.51>

Recent developments in applying and extending Boolean satisfiability (SAT) based techniques have resulted in new types of approaches to pseudo-Boolean optimization (PBO), complementary to the more classical integer programming techniques. In this talk, we outline a first approach to pseudo-Boolean optimization based on instantiating the so-called implicit hitting set (IHS) approach [2, 1], motivated by the success of IHS implementations for maximum satisfiability (MaxSAT). In particular, we harness recent advances in native reasoning techniques for pseudo-Boolean constraints, which enable efficiently identifying inconsistent assignments over subsets of objective function variables (i.e. unsatisfiable cores in the context of PBO), as a basis for developing a native IHS approach to PBO, and study the impact of various search techniques applicable in the context of IHS for PBO.

References

- 1 P. Smirnov, J. Berg, and M. Järvisalo. Improvements to the implicit hitting set approach to pseudo-boolean optimization. In *SAT 2022*.
- 2 P. Smirnov, J. Berg, and M. Järvisalo. Pseudo-boolean optimization by implicit hitting sets. In *CP 2021*.

3.14 Exploring Algebraic Methods for Circuit Verification

Daniela Kaufmann (TU Wien, AT)

License © Creative Commons BY 4.0 International license
© Daniela Kaufmann

Joint work of Kaufmann, Daniela; Biere, Armin; Kauers, Manuel; Fleury, Mathias; Beame, Paul; Nordström, Jakob
Main reference Daniela Kaufmann, Paul Beame, Armin Biere, Jakob Nordström: “Adding Dual Variables to Algebraic Reasoning for Gate-Level Multiplier Verification”, in Proc. of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1431–1436, 2022.

URL <http://dx.doi.org/10.23919/DATE54114.2022.9774587>

Digital circuits are widely utilized in computers, because they provide models for various digital components and arithmetic operations. To avoid problems like the infamous Pentium FDIV bug, it is critical to ensure that these circuits are correct. Formal verification can be used to evaluate whether a circuit meets a given specification. Arithmetic circuits, in particular integer multipliers, pose a challenge to current verification approaches. Techniques that rely solely on SAT solving or decision diagrams appear incapable of tackling this problem in an acceptable period of time. In practice, circuit verification still requires a substantial amount of manual labor.

In this talk, we will demonstrate an automated verification technique that is based on algebraic reasoning and is currently considered to be one of the most successful verification methods for circuit verification. In this approach the circuit is modeled as a set of polynomial equations that is implied by the circuit. For a correct circuit, we must demonstrate that the specification is implied by the polynomial representation of the given circuit. However, some sections of the multiplier, such as final stage adders, are difficult to check using simply computer algebra. To address this issue, we will provide a hybrid solution that blends SAT and computer algebra.

But who verifies the verifier? The ability to independently generate and check proof certificates boosts confidence in the outcomes of automated reasoning tools. We present an algebraic proof calculus that allows us to obtain certificates as a by-product of circuit verification and that can be efficiently verified with our independent proof checking tools.

3.15 First-Order Theorem Proving – Theory and Practice

Laura Kovács (TU Wien, AT)

License © Creative Commons BY 4.0 International license
© Laura Kovács

First-order theorem proving is undergoing a rapid development thanks to its successful use in software analysis, formal verification, IT security, symbolic computation, theorem proving in mathematics, and other related areas. Breakthrough results in all areas of theorem proving have been obtained, including improvements in theory, implementation, and the development of powerful theorem proving tools.

This talk serves as a mini-tutorial on the theory and practice of first-order theorem proving. We introduce the core concepts of automating first-order theorem proving in first-order logic with equality. We will discuss the resolution and superposition calculus, introduce the saturation principle, present various algorithms implementing redundancy elimination, and demonstrate how these concepts are implemented in our Vampire theorem prover. We also survey practical considerations for making saturation efficient.

The talk will next be complemented with the presentation of Dr. Martin Suda, discussing applications of SAT solvers in the efficient automation of first-order reasoning.

3.16 Towards a Deeper Understanding of Modern CDCL SAT Solvers

Chunxiao (Ian) Li (University of Waterloo, CA), Jonathan Chung, Vijay Ganesh (University of Waterloo, CA), Antonina Kolokolova (University of Newfoundland, CA), Alice Mu, Soham Mukherjee, and Marc Vinyals (University of Newfoundland, CA)

License © Creative Commons BY 4.0 International license

© Chunxiao (Ian) Li, Jonathan Chung, Vijay Ganesh, Antonina Kolokolova, Alice Mu, Soham Mukherjee, and Marc Vinyals

Main reference Chunxiao Li, Jonathan Chung, Soham Mukherjee, Marc Vinyals, Noah Fleming, Antonina Kolokolova, Alice Mu, Vijay Ganesh: “On the Hierarchical Community Structure of Practical Boolean Formulas”, in Proc. of the Theory and Applications of Satisfiability Testing – SAT 2021 – 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings, Lecture Notes in Computer Science, Vol. 12831, pp. 359–376, Springer, 2021.

URL http://dx.doi.org/10.1007/978-3-030-80223-3_25

Understanding why state-of-the-art SAT solvers are empirically successful has been a long standing question since the beginning of solver research. An ideal answer to the question should be both empirically verifiable, and in the same time can be theoretically analyzed. To shed light on this difficult problem, I will present a novel concept for SAT formulas and proofs, namely the hierarchical community structure (HCS). Empirically we show that hierarchical community structure can be used to distinguish industrial formulas from random, crafted and crypto formulas. And theoretically, we prove size upper bounds parameterized in the HCS structure. I will also discuss some recent developments in the locality of proofs through the lens of HCS.

3.17 Quantified Boolean Formulas: (Solving and) Proof Complexity

Meena Mahajan (The Institute of Mathematical Sciences – Chennai, IN)

License © Creative Commons BY 4.0 International license

© Meena Mahajan

QBF solving brings many new challenges and has thrown up many innovative approaches and heuristics. QBF proof complexity explores the theoretical underpinnings of these approaches rigorously, explains relative strengths of different approaches, exposes limitations, and suggests new approaches. This talk will survey some of the developments in the area.

3.18 How Constraint Programming Isn’t Like SAT

Ciaran McCreesh (University of Glasgow, GB)

License © Creative Commons BY 4.0 International license

© Ciaran McCreesh

This talk provides an overview of modern constraint programming and how it differs from SAT solving, both in technology and terminology. I’ll give an introduction to how the CP community thinks and speaks, starting with modelling and reformulation; then constraints, propagation, and lazy clause generation; and finally, search. Next we’ll take a closer look at the all-different constraint: I’ll explain how it’s propagated and why CNF can’t do the same thing, and then we’ll look at whether stronger propagation is actually a good idea in practice. I’ll conclude with a look at three exciting research topics: proof logging, belief propagation, and parallel search.

3.19 Certification of Samplers

Kuldeep S. Meel (National University of Singapore, SG)

License © Creative Commons BY 4.0 International license
© Kuldeep S. Meel

Joint work of Chakraborty, Sourav; Golia, Priyanka; Soos, Mate; Meel, Kuldeep S.

Main reference Mate Soos, Priyanka Golia, Sourav Chakraborty, Kuldeep S. Meel: “On Quantitative Testing of Samplers”, in Proc. of the 28th International Conference on Principles and Practice of Constraint Programming, CP 2022, July 31 to August 8, 2022, Haifa, Israel, LIPICs, Vol. 235, pp. 36:1–36:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

URL <http://dx.doi.org/10.4230/LIPICs.CP.2022.36>

Given a Boolean formula F , the problem of constrained sampling is to generate solutions uniformly at random. Constrained sampling is a fundamental problem in computer science with various applications. While constrained sampling is a computationally challenging problem, the SAT revolution has paved the way for various approaches focused on designing scalable samplers. On one end of the spectrum, we have techniques that provide provably rigorous guarantees but often fail to scale to large instances. On the other hand, we have samplers that can scale to large instances, but theoretical tools are often insufficient to argue about the quality of these samplers.

I will discuss our project, Barbarik, which seeks to certify samplers [1]. Barbarik builds on the recent advances in distribution testing and can identify samplers whose distributions are not of high quality. The availability of Barbarik led us to investigate the possibility of test-driven development of constrained samplers, and our efforts yielded CMSGen [2], a sampler that augments the standard CDCL routine with *just enough* randomization so as not to be *caught* by Barbarik. Surprisingly, CMSGen is highly effective in practice and thus paving the way forward to a new design methodology for samplers [3].

References

- 1 Sourav Chakraborty and Kuldeep S. Meel. On testing of uniform samplers. In *Proc. of AAAI*, pages 7777–7784. AAAI Press, 2019.
- 2 Priyanka Golia, Mate Soos, Sourav Chakraborty, and Kuldeep S. Meel. Designing samplers is easy: The boon of testers. In *Proc. of FMCAD*, pages 222–230, 2021.
- 3 Mate Soos, Priyanka Golia, Sourav Chakraborty, and Kuldeep S. Meel. On quantitative testing of samplers. In Christine Solnon, editor, *Proc. of CP*, volume 235 of *LIPICs*, pages 36:1–36:16, 2022.

3.20 Proof complexity and SAT solving


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

License © Creative Commons BY 4.0 International license
© Jakob Nordström

This talk is intended to give an overview of proof complexity and connections to Boolean satisfiability (SAT) solving. The focus will be on proof systems (and corresponding algorithms) such as resolution (DPLL and conflict-driven clause learning), Nullstellensatz and polynomial calculus (linear algebra and Gröbner basis computations), and cutting planes (pseudo-Boolean solving and 0-1 integer linear programming). Time permitting, we will also discuss briefly proof systems such as Sherali-Adams and sums of squares (linear programming and semidefinite programming hierarchies), stabbing planes (0-1 ILP), and extended resolution (SAT pre- and inprocessing).

3.21 Certified CNF Translations for Pseudo-Boolean Solving

Andy Oertel (Lund University, SE)

License  Creative Commons BY 4.0 International license
 © Andy Oertel

Joint work of Gocht, Stephan; Martins, Ruben; Nordström, Jakob; Oertel, Andy
Main reference Stephan Gocht, Ruben Martins, Jakob Nordström, Andy Oertel: “Certified CNF Translations for Pseudo-Boolean Solving”, in Proc. of the 25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel, LIPIcs, Vol. 236, pp. 16:1–16:25, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
URL <http://dx.doi.org/10.4230/LIPIcs.SAT.2022.16>

The dramatic improvements in Boolean satisfiability (SAT) solving since the turn of the millennium have made it possible to leverage state-of-the-art conflict-driven clause learning (CDCL) solvers for many combinatorial problems in academia and industry, and the use of proof logging has played a crucial role in increasing the confidence that the results these solvers produce are correct. However, the fact that SAT proof logging is performed in conjunctive normal form (CNF) clausal format means that it has not been possible to extend guarantees of correctness to the use of SAT solvers for more expressive combinatorial paradigms, where the first step is an unverified translation of the input to CNF.

In this work, we show how cutting-planes-based reasoning can provide proof logging for solvers that translate pseudo-Boolean (a.k.a. 0-1 integer linear) decision problems to CNF and then run CDCL. To support a wide range of encodings, we provide a uniform and easily extensible framework for proof logging of CNF translations. We are hopeful that this is just a first step towards providing a unified proof logging approach that will also extend to maximum satisfiability (MaxSAT) solving and pseudo-Boolean optimization in general.

3.22 Scalable optimization with SAT-based local improvement (SLIM)

Andre Schidler (TU Wien, AT), Friedrich Slivovsky (TU Wien, AT), and Stefan Szeider (TU Wien, AT)

License  Creative Commons BY 4.0 International license
 © Andre Schidler, Friedrich Slivovsky, and Stefan Szeider

SAT-based local improvement (SLIM) is an optimization metaheuristic. It repeatedly employs SAT-based solvers to local versions of the problem instance at hand, gradually improving a heuristically computed initial global solution. SLIM has been successfully instantiated for several problems, including graph decomposition and coloring, decision tree induction, Bayesian network structure learning, and circuit synthesis.

3.23 On the Use of SAT Solvers in a Modern ATP

Martin Suda (Czech Technical University – Prague, CZ)

License © Creative Commons BY 4.0 International license
© Martin Suda

Joint work of Reger, Giles; Suda, Martin

Main reference Giles Reger, Martin Suda: “The Uses of SAT Solvers in Vampire”, in Proc. of the 1st and 2nd Vampire Workshops, Vampire@VSL 2014, Vienna, Austria, July 23, 2014 / Vampire@CADE 2015, Berlin, Germany, August 2, 2015, EPiC Series in Computing, Vol. 38, pp. 63–69, EasyChair, 2015.

URL <http://dx.doi.org/10.29007/4w68>

A modern saturation-based automatic theorem prover for first-order logic (ATP) relies on the SAT technology at various places. We have the Inst-Gen calculus and global subsumption reduction rule, MACE-style finite model finding, and, most notably, the AVATAR architecture. I will give an overview of these pieces of technology and outline how they contribute to the performance of our ATP Vampire.

3.24 The Last Mile in Trustworthy Automated Reasoning

Yong Kiam Tan (Infocomm Research – Singapore, SG)

License © Creative Commons BY 4.0 International license
© Yong Kiam Tan

Joint work of Tan, Yong Kiam; Heule, Marijn J. H.; Myreen, Magnus O.

Main reference Yong Kiam Tan, Marijn J. H. Heule, Magnus O. Myreen: “cake_lpr: Verified Propagation Redundancy Checking in CakeML”, in Proc. of the Tools and Algorithms for the Construction and Analysis of Systems – 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 – April 1, 2021, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 12652, pp. 223–241, Springer, 2021.


URL http://dx.doi.org/10.1007/978-3-030-72013-1_12

State-of-the-art automated reasoning tools are complex and highly-optimized pieces of software. This complexity can lead to an increased risk of soundness-critical bugs, which may affect the trustworthiness of automatically generated results. To remedy this state of affairs, many tools now generate proof logs (or proof certificates), which can be independently checked for correctness.

This talk is about the “last mile” in highly trustworthy automated reasoning – the development of efficient, formally verified proof checkers that are capable of soundly scrutinizing proof logs for various theories. I will survey theories, proof systems, and proof checkers that have been formalized in proof assistants, including my work with various collaborators on verifying proof checkers using HOL4 and CakeML. Looking ahead, I speculate that today’s foundational software verification tools are well-suited to handle tougher challenges in end-to-end verification of proof checkers. For example, 1) building common infrastructure to ease verification of new proof systems and/or efficient proof checkers; 2) developing a unified proof checker that seamlessly handles proofs from different theories; or 3) verifying proof checkers for proof systems that feature probabilistic or interactive elements.

3.25 A SAT Solver + Computer Algebra Attack on the Minimum Kochen-Specker Problem

Vijay Ganesh (University of Waterloo, CA)


License  Creative Commons BY 4.0 International license
© Vijay Ganesh

Joint work of Vijay Ganesh, Curtis Bright, Brian Li

One of the most fundamental results in the foundations of quantum mechanics is the Kochen–Specker (KS) theorem, a “no-go” theorem which states that contextuality is an essential feature of any hidden-variable theory. The theorem hinges on the existence of a mathematical object called a KS vector system. Although the existence of a KS vector system was first established by Kochen and Specker, the problem of the minimum size of such a system has stubbornly remained open for over 50 years. In this paper, we present a new method based on a combination of a SAT solver and a computer algebra system (CAS) to address this problem. We improve the lower bound on the minimum number of vectors in a KS system from 22 to 23 and improve the efficiency of the search by a factor of over 1000 when compared to the most recent computational methods. Finding a minimum KS system would simplify experimental tests of the KS theorem and have direct applications in quantum information processing, specifically in the security of quantum cryptographic protocols based on complementarity, zero-error classical communication, and dimension witnessing.

3.26 Theoretical limits of 1UIP Learning

Marc Vinyals (University of Newfoundland, CA)

License  Creative Commons BY 4.0 International license
© Marc Vinyals

Even though CDCL can reproduce resolution proofs with at most a polynomial overhead, it is not clear how large that overhead needs to be, or if one is needed at all. We investigate the role that learning schemes play in this simulation by focusing on syntactical properties of proofs generated by CDCL solvers that employ the standard 1UIP learning scheme. In particular we show that proofs of this kind can simulate resolution proofs with at most a linear overhead, but there also exist formulas where such overhead is necessary or, more precisely, that there exist formulas with resolution proofs of linear length that require quadratic CDCL proofs.

3.27 Exponential separations using guarded extension variables

Emre Yolcu (Carnegie Mellon University – Pittsburgh, US)

License  Creative Commons BY 4.0 International license
© Emre Yolcu

Joint work of Emre Yolcu, Heule, Marijn

I talked about the complexity of proof systems augmenting resolution with inference rules that allow, given a formula Γ in conjunctive normal form, deriving clauses that are not necessarily logically implied by Γ but whose addition to Γ preserves satisfiability. When the derived clauses are allowed to introduce variables not occurring in Γ , those systems

become equivalent to extended resolution. We are concerned with their versions “without new variables.” They are called BC^- , RAT^- , SBC^- , and GER^- , denoting respectively blocked clauses, resolution asymmetric tautologies, set-blocked clauses, and generalized extended resolution. Each of them formalizes some restricted version of the ability to make assumptions that hold “without loss of generality,” which is commonly used informally to simplify or shorten proofs. I described several new separations between those systems, which give an almost complete picture of their relative strengths.

4 Evaluation by Participants

In addition to the traditional Dagstuhl evaluation after the workshop, the organizing committee also arranged for a separate evaluation which specific questions about different aspects of the workshop. Below follows a summary of the answers collected in both evaluations.

In the Dagstuhl survey, the scientific quality of the workshop was ranked very highly, and the workshop also scored highly on the questions whether it inspired new ideas, led to insights from neighbouring fields or communities, and inspired new research. Some participants pointed out technical problems during the workshop with the beamer and pointers in the seminar room (which seem to have since been resolved) and with e-mail messages from organizers not getting through to participants via the seminar mailing list.

There were comments about how information about informal talks and gatherings can be spread more efficiently among all participants, and, in particular, about the fact that there was an unfortunate collision between an informal presentation and the music night on Thursday evening.

In the organizers survey (which was filled in by 23 participants – a similar number to the Dagstuhl survey), the decision to have an open problem session was uniformly assessed as good or very good. A majority of respondents to the organizer poll agreed that *not* having a panel discussion was good. A majority of respondents also said that the amount of scheduled activities in the workshop program overall was about right, but a large minority thought it was a bit too much. About an equally large majority found the balance between longer survey talks and shorter contributed talks in the program to be good, but again a large minority found the amount of surveys to be a bit too much, or even clearly too much. Overall, the large number of survey presentations were highlighted by different participants both as a positive and a negative aspect of the workshop in both surveys.

Since the workshop tried to cover a fairly large number of different research areas related to combinatorial solving and optimization, the organizer poll asked the participant whether there was a good balance between depth and breadth. Several participants commented that the coverage of many different areas was a good aspect of the workshop, but there was also a suggestion to maybe focus more clearly on one or two external research areas per workshop in order to be able to go deeper.

Among good aspects to keep for future editions in this workshop series the responses listed:

- keeping the mix of survey and research talks;
- keeping the mix of different research communities;
- setting aside lots of time for informal discussions;
- making sure to invite junior participants, including MSc students.

Some aspects that could be improved were considered to be:

- A bit more slack in the schedule, to allow for longer discussions when talks generated lots of interaction (maybe at the price of starting earlier in the afternoon).
- A better mode of communication than e-mail (or wiki) for spreading information during the workshop (e.g., about informal talks).
- More interaction between theoretical and applied researchers.

All in all, it seems fair to say that the the feedback from the participants was overwhelmingly positive, just as for the Dagstuhl workshop in 2015. When asked if they would come to a similar workshop again in Europe, 22 out of 23 respondents in the organizer poll replied that this is very likely. If such a workshop were instead to be held in North America, a clear majority would still want to come, but the enthusiasm in the responses went down slightly (perhaps reflecting that there was a clear bias of European participants in the workshop this time). For a workshop in East Asia or India, the responses were even more mixed, though there were still more positive than negative replies.

Participants

- Jeremias Berg
University of Helsinki, FI
- Olaf Beyersdorff
Friedrich-Schiller-Universität
Jena, DE
- Armin Biere
Universität Freiburg, DE
- Nikolaj S. Bjørner
Microsoft – Redmond, US
- Benjamin Böhm
Friedrich-Schiller-Universität
Jena, DE
- Bart Bogaerts
VU – Brussels, BE
- Jonas Conneryd
Lund University, SE
- Susanna de Rezende
Lund University, SE
- Katalin Fazekas
TU Wien, AT
- Mathias Fleury
Universität Freiburg, DE
- Vijay Ganesh
University of Waterloo, CA
- Mexi Gioni
Zuse Institut Berlin, DE
- Ambros M. Gleixner
HTW – Berlin, DE
- Malte Helmert
Universität Basel, CH
- Marijn J. H. Heule
Carnegie Mellon University –
Pittsburgh, US
- Matti Järvisalo
University of Helsinki, FI
- Mikoláš Janota
Czech Technical University –
Prague, CZ
- Daniela Kaufmann
TU Wien, AT
- Antonina Kolokolova
University of Newfoundland, CA
- Laura Kovács
TU Wien, AT
- Chunxiao (Ian) Li
University of Waterloo, CA
- Meena Mahajan
The Institute of Mathematical
Sciences – Chennai, IN
- Ciaran McCreesh
University of Glasgow, GB
- Kuldeep S. Meel
National University of
Singapore, SG
- Jakob Nordström
University of Copenhagen, DK &
Lund University, SE
- Andy Oertel
Lund University, SE
- Albert Oliveras
UPC Barcelona Tech, ES
- Pavel Pudlák
The Czech Academy of Sciences –
Prague, CZ
- Torsten Schaub
Universität Potsdam, DE
- Andre Schidler
TU Wien, AT
- Laurent Simon
University of Bordeaux, FR
- Friedrich Slivovsky
TU Wien, AT
- Martin Suda
Czech Technical University –
Prague, CZ
- Stefan Szeider
TU Wien, AT
- Yong Kiam Tan
Infocomm Research –
Singapore, SG
- Dieter Vandesande
VU – Brussels, BE
- Marc Vinyals
University of Newfoundland –
St. John's, CA
- Ryan Williams
MIT – Cambridge, US
- Emre Yolcu
Carnegie Mellon University –
Pittsburgh, US

