Boolean Satisfiability and Beyond: Algorithms, Analysis, and AI Applications

Matti Järvisalo
Helsinki Institute for Information Technology HIIT, Department of Computer Science,
University of Helsinki, Finland

Abstract

This overview accompanies the author's IJCAI-16 Early Career Spotlight Talk, highlighting aspects of the author's research agenda with a strong focus on some of the author's recent research contributions.

1 Introduction

The importance of constraint satisfaction and optimization is underlined by our need to solve and reason about increasingly large and complex systems and computational problems, arising from AI and industrial applications, as efficiently as possible. Motivated by this need, a fundamental quest is to understand the underlying factors that make real-world instances of computational problems hard to solve, and, going beyond this, to harness this knowledge for developing efficient and robust algorithms for solving hard computational problems with high practical relevance. In this quest, it is central to combine rigorous theoretical and empirical analysis of hard decision and optimization problems with the development of robust practical automated reasoning techniques, rather than focusing on either theoretical analysis, implementation-level work on constraint solvers, or domain-specific application studies alone.

From the classical theoretical perspective, determining the satisfiability of a given propositional formula, i.e., the Boolean satisfiability (SAT) problem, is one of the most fundamental decision problems in computer science, being the canonical NP-complete problem. From the more applied perspective, however, despite the presumed intrinsic worst-case intractability, state-of-the-art SAT solver technology today allows us to tackle immense structured search spaces in an efficient manner, to the point that SAT provides the best practical approach to many important problems, surpassing the efficiency of dedicated algorithms.

The SAT-based approach is a form of declarative programming: first, a reduction, often referred to as an encoding, from the original problem at hand to propositional logic is developed. A complete algorithm for determining satisfiability, i.e., a SAT solver, is then used to obtain either solutions or proofs of non-existence of solutions to the original problem instance. SAT solvers act as core search engines within other decision procedures developed for various expressive declarative languages, such as Satisfiability Modulo Theories (SMT), and are also becoming central components in Boolean optimization procedures, either on their own, or together with state-of-the-art integer programming systems. For many practical settings, modern SAT solvers can be viewed as generic "*practical NP oracles*" that can be used a building blocks for developing complex search procedures for tackling important problems *beyond NP*.

In the rest of this companion paper of a IJCAI-16 Early Career Spotlight talk, I aim to give an overview of some of my recent research achievements, focusing on development, analysis, and AI applications of SAT-based approaches within and beyond NP.

2 Inprocessing SAT Solving

Decision procedures for SAT, especially modern conflict-driven clause learning (CDCL) SAT solvers, first put forward by Marques-Silva and Sakallah, act routinely as core solving engines in many industrial and other real-world applications today. Formula simplification techniques applied before the actual satisfiability search, i.e., in preprocessing, have proven integral in enabling efficient conjunctive normal form (CNF) level SAT solving for real-world application domains, and have become an essential part of the SAT solving tool chain. While the most important single preprocessing technique for SAT is *bounded variable elimination*, in recent line of work [Heule *et al.*, 2011; Järvisalo *et al.*, 2012a; Heule *et al.*, 2013; 2015] we have taken SAT preprocessing further by developing novel *clause elimination* [Järvisalo *et al.*, 2012a; Heule *et al.*, 2015] and *binary clause reasoning techniques* [Heule *et al.*, 2011; 2013] that can provide further simplifications especially when applied in conjunction with other preprocessing techniques.

Going beyond developing new algorithmic techniques, understanding the effects of different preprocessing techniques on the underlying problem structure of CNF-encoded problem instances is important, in terms of both the interplay between different CNF-level techniques and the potential of achieving structure-based simplifications solely by reasoning on the clause-level. In this line of work, one of the major insights is that *blocked clause elimination* simulates various earlier structure-based simplifications for circuit-level representations, including, e.g., the Plaisted-Greenbaum encoding [Järvisalo *et al.*, 2012a]. Another important question is how efficiently specific simplification techniques can (even in theory) be implemented; in order to be applicable (until fixpoint) in the SAT solving workflow to real-world problem instances with up to tens of millions of variables and clauses, simplification techniques need to be scalable. As shown in [Järvisalo and Korhonen, 2014], one way of formally analyzing the runtime complexity of simplification techniques is via proving lower bounds conditional to the *strong exponential-time hypothesis*.

Taking things further than mere preprocessing, some of the strongest SAT solvers today add more reasoning to search by interleaving formula simplification and CDCL search. Such *inprocessing SAT solvers* witness the fact that implementing additional deduction rules within CDCL solvers can leverage the efficiency of state-of-the-art SAT solving. However, applying complex combinations of simplification rules during CDCL search comes at a price. It requires in-depth understanding on how different techniques can be combined together and interleaved with the CDCL algorithm in a satisfiability-preserving way. Moreover, the fact that many simplification techniques only preserve satisfiability but not logical equivalence poses additional challenges, since in many practical applications of SAT, a solution is required for satisfiable formulas, not only the knowledge of the satisfiability of the input formula. Hence, when designing inprocessing SAT solvers for practical purposes, one also has to address the intricate task of solution reconstruction. The central article on inprocessing [Järvisalo *et al.*, 2012b] establishes formal foundations for inprocessing SAT solvers via an abstract framework that captures generally the deduction mechanisms applied within inprocessing SAT solvers. The framework consists of four generic and clean deduction rules. Importantly, the rules specify general conditions for sound inprocessing SAT solving, against which specific inprocessing techniques can be checked for correctness, as well as capture solution reconstruction for essentially proposed simplification techniques.

Going beyond pure SAT and thus NP, a goal is to generalize known SAT preprocessing techniques and to developed novel simplification rules for more general SAT-related contexts, such as Quantified SAT (satisfiability of quantified Boolean formulas) [Heule *et al.*, 2015], for extraction of minimally unsatisfiable subsets of clauses (MUSes) [Belov *et al.*, 2013], and Maximum satisfiability (the Boolean optimization counterpart of SAT) [Berg *et al.*, 2015a; 2015b]. In this line of work, an important aspect is to analyze to what extent SAT preprocessing is directly applicable without further restriction—most often it is not—and to go beyond merely adapting already proposed techniques from SAT.

3 Hybrid Boolean Optimization

A great majority of important decision and optimization problems in artificial intelligence and knowledge representation and reasoning (KR) are notoriously hard. In fact, variants of various central KR problems, such as propositional circumscription, abduction and belief revision, are hard at least for the second level of the polynomial hierarchy, and thus presumably go beyond NP. While NP-hard decision and optimization problems are in the classical sense intractable, the rise of surprisingly effective constraint solving technology, including e.g. SAT and integer programming (IP) solvers, enables finding optimal solutions to complex NP-hard real-world problems in a variety of domains.

The number of real-world applications of Maximum satisfiability (MaxSAT), the optimization counterpart of the infamous Boolean satisfiability problem (SAT), is increasing as recent breakthroughs in MaxSAT solvers are making MaxSAT more and more competitive as a constraint optimization paradigm. A great majority of state-of-the-art MaxSAT solvers are *core-guided*, heavily relying on the power of SAT solvers as very effective means of proving unsatisfiability of subsets of soft constraints, or *unsat cores*, in an iterative fashion towards an optimal solution. Sidestepping from the more popular approach to developing approaches to MaxSAT based on iterative applications of SAT solvers alone, the SAT-IP hybrid MaxSAT solvers MaxHS and most recently our own LMHS [Saikko *et al.*, 2016a], implementing so-called *implicit hitting set algorithms* first proposed for different contexts by Karp *et al.*, have taken top positions in recent MaxSAT Evaluations (`http://www.maxsat.udl.cat`). LMHS also offers various additional features, including an incremental application interface, solution enumeration [Saikko *et al.*, 2016a], and integrated preprocessing [Berg *et al.*, 2015a; 2015b].

Motivated by this success, in recent work [Saikko *et al.*, 2016b] we outline a general framework for implicit hitting set algorithms (see also Figure 1). Specifically (but by no means restricted to), the framework is developed with instantiations based on SAT and IP solvers in mind; the SAT solver acts (or, going beyond NP, multiple SAT solvers act) the role of a “core extractor” used for extracting non-solutions, and the IP solver acts as a hitting set optimizer, used for ruling out the thus far found non-solutions from further consideration. The framework thus provides novel algorithms for a variety of hard reasoning tasks via modularly instantiating the core extraction and hitting set modules in domain-specific ways via SAT and IP solvers specifically well-suited for the respective tasks of providing proofs of unsatisfiability and optimiza-

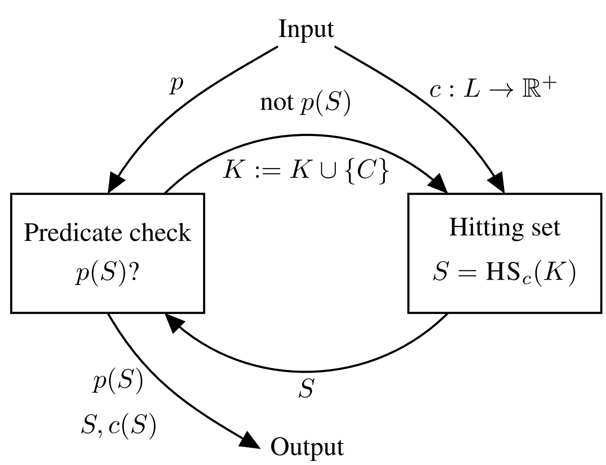

Figure 1: Generic implicit hitting set algorithm

tion. For some more details, as outlined in Figure 1, in the general abstract setting, the approach is based on iteratively checking a predicate p (essentially describing the problem at hand without attention to the objective function c), extracting a domain-specific *core* in case the check fails, and computing a minimum-cost hitting set S over the thus far accumulated set K of cores, until the predicate check succeeds. (For more details, we refer the reader to [Saikko *et al.*, 2016b].)

To illustrate the practical potential of the general framework, going beyond NP, we have shown [Saikko *et al.*, 2016b] that a practical instantiation for the problem of propositional abduction—hard for the second-level of the polynomial hierarchy—surpasses the efficiency of an approach based on disjunctive answer set programming.

4 AI Applications

Argumentation

Argumentation is today a core topic in modern AI research, with potential applications in e.g. decision support, legal reasoning, and multi-agent systems. Argumentation frameworks (AFs) originally proposed by Dung provide the fundamental formal model for knowledge representation and reasoning for many approaches to argumentation. Syntactically, AFs are directed graphs, where arguments are abstract entities represented by vertices. Conflicts among arguments are formalized as attacks, and represented with directed edges between arguments. Semantics of AFs—several of which have been proposed—specify criteria for arguments' acceptance resulting in sets of jointly acceptable arguments called *extensions*. Notably, for central decision problems over AFs, including, e.g., credulous and skeptical reasoning, a great majority of AF semantics give rise to NP-hardness—and, in many cases, beyond-NP complexity. This gives incentives of developing SAT-based beyond-NP procedures for addressing various types of decision and optimization problems arising from argumentation applications.

A successful approach to AF reasoning is provided by our CEGARTIX system [Dvořák *et al.*, 2014]. Implementing a SAT-based counterexample-guided abstraction refinement approach to second-level complete skeptical and credulous reasoning over AFs, the system ranked at the top in the First International Competition on Computational Models of Argumentation (ICCMA 2015; see `http://argumentationcompetition.org`) in beyond-NP problem categories.

While skeptical and credulous acceptance are central AF problems, argumentation is inherently a dynamic process. A more recent focus in computational aspects of argumentation is on understanding and reasoning about argumentation dynamics. While the complexity landscape of non-dynamic problems on AFs is already well-established, the complexity of reasoning about argumentation dynamics is less understood. In a recent line of work, we have focused on the so-called extension enforcement problem in abstract argumentation and its generalizations [Niskanen *et al.*, 2016; Wallner *et al.*, 2016]. In [Wallner *et al.*, 2016], we provide a nearly complete computational complexity map of fixed-argument extension enforcement under various major AF semantics, with results ranging from polynomial-time algorithms to completeness for the second-level of the polynomial hierarchy. Complementing the complexity results, we propose algorithms for NP-hard extension enforcement based on constrained optimization. Going beyond NP, we propose novel counterexample-guided abstraction refinement procedures for the second-level complete problems, as well as an open-source system implementation of the approach. Most recently, we have generalized the approach to the so-called *status enforcement problem* [Niskanen *et al.*, 2016], bringing together concepts from both static credulous/skeptical acceptance and AF dynamics, most closely, extension enforcement.

Machine Learning

Integration of the fields of constraint solving and machine learning has recently been identified within the AI community as an important research direction with high potential. From the perspective of constraint solving, there are great opportunities for developing novel constraint-based approaches to various data analysis tasks. As illustrated in Figure 2 for the problem of learning optimal Bayesian network structures BNSL, this is due to the fact that, after processing the input data at hand into a suitable form (to obtain, e.g., local scores in BNSL), in many cases the underlying computational task has an exact declarative form as a constrained optimization problem. In a recent line of research, we have developed new approaches to various data analysis problems, including structure learning of Bayesian networks [Berg *et al.*, 2014; Saikko *et al.*, 2015] and chain graphs [Sonntag *et al.*, 2015], causal discovery and inference [Hyttinen *et al.*, 2013; 2014; 2015], correlation clustering [Berg and Järvisalo, 2016], information visualization [Bunte *et al.*, 2014], and frequent itemset mining [Järvisalo, 2011], based on SAT and Boolean optimization solvers. As shown in these works, applying constraint solvers in such domains can have various kinds of benefits, e.g., in terms of the quality of solutions obtained [Berg and Järvisalo, 2016; Malone *et al.*, 2015; Hyttinen *et al.*, 2014]; the ability to integrate non-trivial additional constraints—such as bounding the treewidth of network structures—on the solutions of interest [Berg *et al.*, 2014; Berg and Järvisalo, 2016]; and the potential of developing more general exact approaches in comparison with earlier proposed algorithmic solutions [Hyttinen *et al.*, 2013; 2014].

Acknowledgments

The author is financially supported by Academy of Finland under grants 251170 COIN, 276412, and 284591; and Research Funds of the University of Helsinki. The author sincerely thanks all of his co-authors.

References

[Belov *et al.*, 2013] A. Belov, M. Järvisalo, and J. Marques-Silva. Formula preprocessing in MUS extraction. In *Proc. TACAS*, volume 7795 of *LNCS*, pages 110–125. Springer, 2013.

[Berg and Järvisalo, 2016] J. Berg and M. Järvisalo. Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artif. Intell.*, 2016.

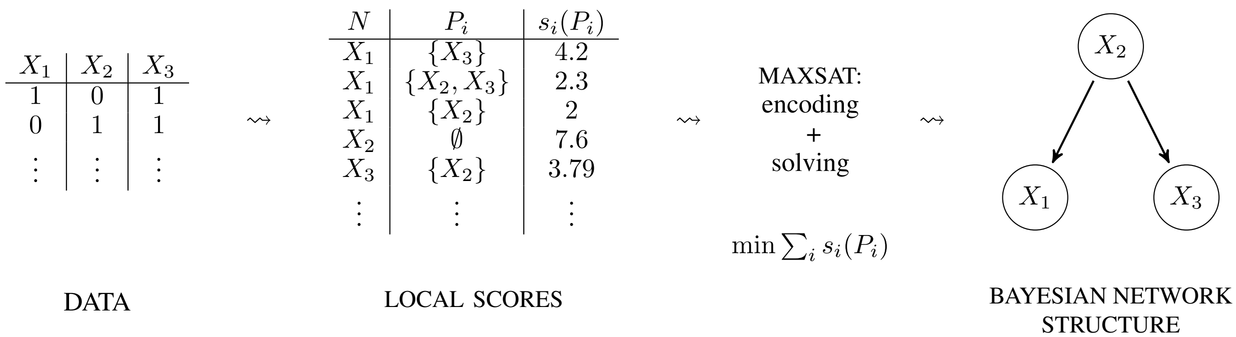

Figure 2: The Bayesian network structure learning problem by example

[Berg *et al.*, 2014] J. Berg, M. Järvisalo, and B. Malone. Learning optimal bounded treewidth bayesian networks via maximum satisfiability. In *Proc. AISTATS*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 86–95. JMLR, 2014.

[Berg *et al.*, 2015a] J. Berg, P. Saikko, and M. Järvisalo. Improving the effectiveness of SAT-based preprocessing for MaxSAT. In *Proc. IJCAI*, pages 239–245. AAAI Press, 2015.

[Berg *et al.*, 2015b] J. Berg, P. Saikko, and M. Järvisalo. Reusing auxiliary variables for MaxSAT preprocessing. In *Proc. ICTAI*, pages 813–820. IEEE, 2015.

[Bunte *et al.*, 2014] K. Bunte, M. Järvisalo, J. Berg, P. Myllymäki, J. Peltonen, and S. Kaski. Optimal neighborhood preserving visualization by maximum satisfiability. In *Proc. AAAI*, pages 1694–1700. AAAI Press, 2014.

[Dvořák *et al.*, 2014] W. Dvořák, M. Järvisalo, J.P. Wallner, and S. Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, 206:53–78, 2014.

[Heule *et al.*, 2011] M. Heule, M. Järvisalo, and A. Biere. Efficient CNF simplification based on binary implication graphs. In *Proc. SAT*, volume 6695 of *LNCS*, pages 201–215, 2011.

[Heule *et al.*, 2013] M. Heule, M. Järvisalo, and A. Biere. Revisiting hyper binary resolution. In *Proc. CPAIOR*, volume 7874 of *LNCS*, pages 77–93, 2013.

[Heule *et al.*, 2015] M. Heule, M. Järvisalo, F. Lonsing, M. Seidl, and A. Biere. Clause elimination for SAT and QSAT. *J. Artif. Intell. Res.*, 53:127–168, 2015.

[Hyttinen *et al.*, 2013] A. Hyttinen, P. Hoyer, F. Eberhardt, and M. Järvisalo. Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *Proc. UAI*, pages 301–310. AUAI Press, 2013.

[Hyttinen *et al.*, 2014] A. Hyttinen, F. Eberhardt, and M. Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In *Proc. UAI*, pages 340–349. AUAI Press, 2014.

[Hyttinen *et al.*, 2015] A. Hyttinen, F. Eberhardt, and M. Järvisalo. Do-calculus when the true graph is unknown. In *Proc. UAI*, pages 395–404. AUAI Press, 2015.

[Järvisalo and Korhonen, 2014] M. Järvisalo and J.H. Korhonen. Conditional lower bounds for failed literals and related techniques. In *Proc. SAT*, volume 8561 of *LNCS*, pages 75–84. Springer, 2014.

[Järvisalo *et al.*, 2012a] M. Järvisalo, A. Biere, and M. Heule. Simulating circuit-level simplifications on CNF. *J. Autom. Reasoning*, 49(4):583–619, 2012.

[Järvisalo *et al.*, 2012b] M. Järvisalo, M. Heule, and A. Biere. Inprocessing rules. In *Proc. IJCAR*, volume 7364 of *LNCS*, pages 355–370. Springer, 2012.

[Järvisalo, 2011] M. Järvisalo. Itemset mining as a challenge application for answer set enumeration. In *Proc. LPNMR*, volume 6645 of *LNAI*, pages 304–311, 2011.

[Malone *et al.*, 2015] B. Malone, M. Järvisalo, and P. Myllymäki. Impact of learning strategies on the quality of Bayesian networks: An empirical evaluation. In *Proc. UAI*, pages 362–371. AUAI Press, 2015.

[Niskanen *et al.*, 2016] A. Niskanen, J.P. Wallner, and M. Järvisalo. Optimal status enforcement in abstract argumentation. In *Proc. IJCAI*. AAAI Press, 2016.

[Saikko *et al.*, 2015] P. Saikko, B. Malone, and M. Järvisalo. MaxSAT-based cutting planes for learning graphical models. In *Proc. CPAIOR*, volume 9075 of *LNCS*, pages 345–354. Springer, 2015.

[Saikko *et al.*, 2016a] P. Saikko, J. Berg, and M. Järvisalo. LMHS: A SAT-IP hybrid MaxSAT solver. In *Proc. SAT*, LNCS. Springer, 2016.

[Saikko *et al.*, 2016b] P. Saikko, J.P. Wallner, and M. Järvisalo. Implicit hitting set algorithms for reasoning beyond NP. In *Proc. KR*. AAAI Press, 2016.

[Sonntag *et al.*, 2015] D. Sonntag, M. Järvisalo, J.M. Peña, and A. Hyttinen. Learning optimal chain graphs with answer set programming. In *Proc. UAI*, pages 822–831. AUAI Press, 2015.

[Wallner *et al.*, 2016] J.P. Wallner, A. Niskanen, and M. Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. In *Proc. AAAI*. AAAI Press, 2016.