PACE Solver Description: Minimum Hitting Set Computation via Core-Guided MaxSAT Solving

André Schidler
University of Freiburg, Germany

Abstract

This paper describes our hybrid MaxSAT and mixed integer programming approach for finding minimum hitting sets as submitted to the 2025 PACE challenge. We also discuss hitting set specific challenges, lower bounds, preprocessing and design choices.

2012 ACM Subject Classification Theory of computation → Discrete optimization; Theory of computation → Constraint and logic programming; Theory of computation → Automated reasoning

Keywords and phrases hitting set, maxsat, core-guided

Digital Object Identifier 10.4230/LIPIcs.IPEC.2025.37

Supplementary Material *Software*: `https://github.com/ASchidler/PACE25-submission`
archived at `swh:1:dir:44546f4c17c6ee57221ccfa0ce05db41be0e310b`

Funding Funded by an Amazon Research Award (Fall/2023).

1 Introduction

The 2025 PACE Challenge evaluated (among others) new approaches for finding a *minimum hitting set (MHS)*, where given a set of sets $\mathcal{S}$ the goal is to find a minimum size set H – the hitting set – that intersects with all given sets, i.e., $\emptyset \neq H \cap S$, for all $S \in \mathcal{S}$. The decision version of the problem is one of the original 21 NP-complete problems identified by Karp [12]. We present the details to our approach that uses a combination of MaxSAT and mixed integer programming (MIP) to tackle the problem.

The use of a MaxSAT solver is motivated by the success of a MaxSAT approach for the directed feedback vertex set problem at the 2022 PACE challenge, which could be converted into an implicit hit set problem, where not all sets are known a priori [13, 14]. Further, MHS can be seen as a special case of MaxSAT, where variables occur in only one polarity. However, the MaxSAT solver solves only 94 of the 100 public competition instances within 30 minutes, which necessitates a complementary approach for the remaining instances.

MaxSAT solvers often use MIP solvers, either as a preprocessing step or to explicitly solve an (implicit) MHS posed by the MaxSAT solver [9, 16]. The use of a MIP solver before the actual MaxSAT algorithm is motivated by the observation that MIP solvers are often faster on small but hard instances [3], which matches the characteristics of the MHS instances not solved by our MaxSAT solver. While neither solver can solve all public competition instances, combined they are able to solve all 100 instances within 30 minutes.

2 Architecture

Our approach first preprocesses (Section 3) the instance and then passes it to either the MIP solver or our MaxSAT solver (Section 4) depending on the number of distinct elements of the instance: any instance with fewer than 250 distinct elements ($|\bigcup_{S \in \mathcal{S}} S|$) is passed on to the MIP solver.

The preprocessing and MaxSAT implementations are ours and will be described subsequently. As the MIP solver, we use SCIP[1] [4] (version 8.1.0). Another advantage of this approach – although not used in our implementation – is that SCIP as well as MaxSAT offer a mode where the optimality of the solution can be certified [2, 8].

Correctness

The preprocessing, as subsequently discussed, is straightforward and not novel. We trust the correctness of the SCIP solver, as it is a well-established academic solver, based on proven methods. We implement the OLL [1, 15] algorithm for MaxSAT solving. The correctness of the algorithm has been proven in prior work and recently it has been shown that the algorithm, including all state-of-the-art extensions used in our solver, can be certified [2]. We only adjust the heuristics of the algorithm, hence, the approach remains correct.

3 Preprocessing

We use two simple preprocessing rules whose correctness follow from propositional logic and are also mentioned in hitting set literature [6].

1. Whenever there exist two distinct elements u and v such that v occurs in every set that u occurs in, we can remove u from all sets, as there exists a minimum hitting set not using u. Further, replacing u by v in a minimum hitting set yields another minimum hiting set.
2. Whenever there exist two distinct sets $S, S' \in \mathcal{S}$ such that $S \subset S'$, we can remove S', as any hitting set that intersects S also intersects S'.

Results show that the first rule is crucial for good performance, as SCIP does not solve several instances without this preprocessing. The reduction is also necessary to accurately identify instances that should be tackled using a MIP solver. Our experiments suggest that the second rule is unnecessary, as MIP and MaxSAT solvers already efficiently deal with this type of redundancy.

4 MaxSAT Solver

Proving that no hitting set of a given size exists is hard for SAT solvers when limiting the hitting set size with a single cardinality constraint. The reason for this behavior is that this propositional encoding provides very little structure that is exploitable by the SAT solver to avoid iterating over all possible sets of this given maximum size. Indeed, we expect that pure SAT encodings of the problem are similar to pigeonhole encodings, which are known to require exponential runtime for most SAT solvers [10]. This is relevant as most exact MaxSAT algorithms use a SAT solver in their backend.

We implement the core-guided algorithm OLL [1, 15] using Cadical 2.1.3 [5] with hyperparameters manually adjusted to MHS. Core-guided approaches use *cores* to dissect the MaxSAT instance into smaller parts and then connect these cores to form *meta-cores*. (Meta-)cores are used to reformulate the original MaxSAT instance, which introduces structure to the propositional formula visible to the internal SAT solver. This structure makes it easier for the SAT solver to find a minimum hitting set and prove optimality. Which (meta-)cores are used for these reformulations determines the structure of the propositional formula and can have immense impact on the time required to solve the instance [17].

[1] `https://www.scipopt.org`

Upper bounds are one way that can help find good (meta-)cors for MHS. We initially run local search for five seconds to get an initial solution. We use our implementation of the NuWLS local search algorithm [7] for this purpose. This upper bound guides the search for (meta-)cores, can avoid a costly last call to the SAT solver, and is also used to warmstart the MIP solver whenever the MHS instances is small.

Lower Bounds

We can initialize the MaxSAT solver with a promisingly structured reformulation using a good lower bound, a well-known method withing MaxSAT solvers [11]. The simplest lower bound is finding a large set of disjoint sets, where the cardinality of this set acts as a lower bound. This lower bound can be further improved whenever the instance contains sets of size two: let G be the graph created by viewing all sets of size two as edges. Then, a clique of size k implies that at least $k-1$ elements of the clique have to by in any minimum hitting set. This lower bound can be directly translated into OLL reformulations with a usually beneficial structure.

Finding good cliques is not the same as finding a large maximal clique: we can select several disjoint cliques but all cliques and other sets have to be disjoint. It is not established what the desired properties of such a lower bounding set are and different MaxSAT solvers follow different strategies. We repeatedly use a greedy heuristic using randomization and different criteria to find diverse sets of large cliques. For each such set of cliques, we greedily complete our lower bounding set adding other sets that are disjoint. We then choose the lower bounding set based on three criteria in descending priority: (i) high lower bound provided by the cliques, (ii) high lower bound provided by the whole set, and (iii) small cardinality of the lower bounding set.

5 Future Work

We use the MIP solver in our approach to solve small and complex instances. However, results from experiments running the MaxSAT solver repeatedly with different settings show that our solver can solve these instances. The problem is that the runtimes vary by orders of magnitude with several being within half an hour. Further analysis showed that this is mainly due to the structure introduced by the reformulations [17]. So far we were not able to establish a sufficiently good strategy for finding (meta-)cores that provides consistently good results, but see this as an interesting future challenge.

References

1 Benjamin Andres, Benjamin Kaufmann, Oliver Matheis, and Torsten Schaub. Unsatisfiability-based optimization in clasp. In *ICLP (Technical Communications)*, volume 17 of *LIPIcs*, pages 211–221. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. `doi:10.4230/LIPICS.ICLP.2012.211`.

2 Jeremias Berg, Bart Bogaerts, Jakob Nordström, Andy Oertel, and Dieter Vandesande. Certified core-guided maxsat solving. In *Automated Deduction - CADE 29 - 29th International Conference on Automated Deduction, Rome, Italy, July 1-4, 2023, Proceedings*, volume 14132 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2023. `doi:10.1007/978-3-031-38499-8_1`.

3 Jeremias Berg, Matti Järvisalo, Ruben Martins, Andreas Niskanen, and Tobias Paxian. *MaxSAT Evaluation 2024: Solver and Benchmark Descriptions*. Department of Computer Science Series of Publications B. Department of Computer Science, University of Helsinki, Finland, 2024.

4 Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 8.0. ZIB-Report 21-41, Zuse Institute Berlin, December 2021.

5 Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. CaDiCaL 2.0. In *CAV*, volume 14681 of *Lecture Notes in Computer Science*, pages 133–152. Springer, 2024. doi:10.1007/978-3-031-65627-9_7.

6 Thomas Bläsius, Tobias Friedrich, David Stangl, and Christopher Weyand. An efficient branch-and-bound solver for hitting set. In *Proceedings of the Symposium on Algorithm Engineering and Experiments, ALENEX 2022, Alexandria, VA, USA, January 9-10, 2022*, pages 209–220. SIAM, 2022. doi:10.1137/1.9781611977042.17.

7 Yi Chu, Shaowei Cai, and Chuan Luo. Nuwls: Improving local search for (weighted) partial maxsat by new weighting techniques. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 3915–3923. AAAI Press, 2023. doi:10.1609/AAAI.V37I4.25505.

8 William J. Cook, Thorsten Koch, Daniel E. Steffy, and Kati Wolter. A hybrid branch-and-bound approach for exact rational mixed-integer programming. *Math. Program. Comput.*, 5(3):305–344, 2013. doi:10.1007/S12532-013-0055-6.

9 Jessica Davies and Fahiem Bacchus. Exploiting the power of MIP solvers in MaxSat. In *SAT*, volume 7962 of *Lecture Notes in Computer Science*, pages 166–181. Springer, 2013. doi:10.1007/978-3-642-39071-5_13.

10 Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985. doi:10.1016/0304-3975(85)90144-6.

11 Alexey Ignatiev, António Morgado, and João Marques-Silva. RC2: an efficient maxsat solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019. doi:10.3233/SAT190116.

12 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

13 Rafael Kiesel and André Schidler. PACE solver description: Dager - cutting out cycles with maxsat. In *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPIcs*, pages 32:1–32:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.IPEC.2022.32.

14 Rafael Kiesel and André Schidler. A dynamic maxsat-based approach to directed feedback vertex sets. In *Proceedings of the Symposium on Algorithm Engineering and Experiments, ALENEX 2023, Florence, Italy, January 22-23, 2023*, pages 39–52. SIAM, 2023. doi:10.1137/1.9781611977561.CH4.

15 António Morgado, Carmine Dodaro, and João Marques-Silva. Core-guided MaxSAT with soft cardinality constraints. In *CP*, volume 8656 of *Lecture Notes in Computer Science*, pages 564–573. Springer, 2014. doi:10.1007/978-3-319-10428-7_41.

16 Paul Saikko, Jeremias Berg, and Matti Järvisalo. LMHS: A SAT-IP hybrid MaxSAT solver. In *SAT*, volume 9710 of *Lecture Notes in Computer Science*, pages 539–546. Springer, 2016. doi:10.1007/978-3-319-40970-2_34.

17 André Schidler and Stefan Szeider. Analyzing reformulation performance in core-guided maxsat solving. In *28th International Conference on Theory and Applications of Satisfiability Testing, SAT 2025, August 12-15, 2025, Glasgow, Scotland*, volume 341 of *LIPIcs*, pages 26:1–26:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.SAT.2025.26.