Enhanced Lower Bound Computation in Branch-and-Bound for MaxSAT

Jialu Zhang
MIS UR 4290, Université de Picardie Jules Verne, Amiens, France

Chu-Min Li
MIS UR 4290, Université de Picardie Jules Verne, Amiens, France

Sami Cherif
MIS UR 4290, Université de Picardie Jules Verne, Amiens, France

Shuolin Li
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Abstract

Maximum Satisfiability (MaxSAT) is an optimization extension of the Satisfiability (SAT) problem. In Branch-and-Bound (BnB) MaxSAT solving, the quality of the lower bound estimation is critical for effective search space pruning. State-of-the-art BnB solvers typically estimate this bound by identifying disjoint inconsistent subformulas (cores) via Unit Propagation (UP). However, a limitation of this standard approach is that UP fails to detect cores that exhibit complex dependencies with already identified cores. In this paper, we propose a further lookahead algorithm that leverages pre-detected cores to uncover additional disjoint inconsistencies, thereby tightening the lower bound. Experimental results demonstrate that the proposed algorithm significantly tightens the lower bound, enabling the state of the art BnB solver MaxCDCL to solve more instances.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases Maximum Satisfiability, Branch and Bound, Lower Bound

1 Introduction

MaxSAT is an important optimization problem that can be used in many domains, including scheduling [2], hardware and software debugging [8], explainable artificial intelligence [3], maximum clique [5], and protein design [1], among many others. MaxSAT algorithms are commonly divided into two categories: exact (complete) algorithms and heuristic (incomplete) algorithms. In this paper, we focus on exact algorithms, which mainly fall into two classes: SAT-based approaches and Branch-and-Bound (BnB) approaches. In SAT-based approaches, MaxSAT is reduced to a sequence of SAT decision problems, each solved by a SAT oracle. In BnB approaches, the search algorithm explores the assignment tree in a depth-first manner. A key factor in the efficiency of BnB algorithms is the ability to compute a tight lower bound (LB) for each partial assignment. MaxCDCL [6] is a state-of-the-art BnB MaxSAT solver that won the unweighted track of the MaxSAT Evaluation (MSE) 2024.

2 Further Lookahead Algorithm

Recently, Li et al. [7] incorporated the unlocking mechanism into MaxCDCL, which improves LB estimation by exploiting the currently detected disjoint inconsistent subformulas. However, the existing unlocking mechanism in MaxCDCL [4] unlocks locked soft literals only passively, which limits the capacity of MaxCDCL for detecting a tight LB estimation. To remedy this drawback, we propose a *further lookahead* (FLA) algorithm, by actively unlocking previous cores detected by the standard lookahead algorithm. This enables the detection of additional cores when the standard lookahead fails to tighten the lower bound. Fundamentally, a core κ

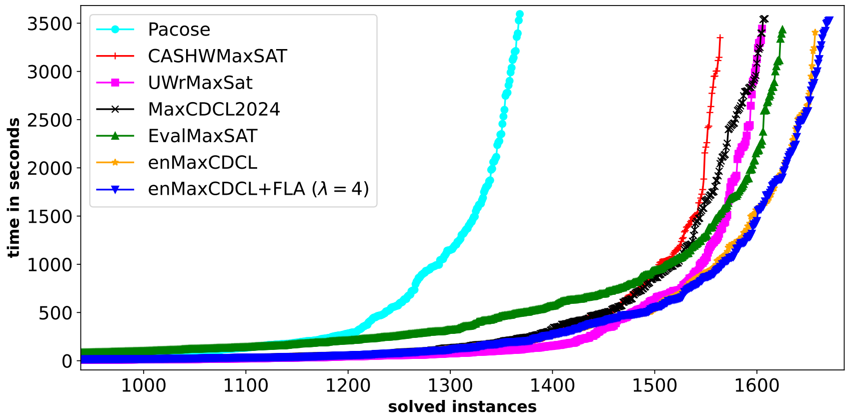

Figure 1 Cactus plot illustrating the cumulative distribution of solved instances for state-of-the-art MaxSAT solvers on the MSE 2019–2024 benchmarks.

with $w(\kappa) = 1$ is guaranteed to contain at least one falsified literal for any extension of α, and allows to increase LB by 1 thanks to this guarantee. If one of the literals in κ is falsified, the weight $w(\kappa) = 1$ should be removed from LB, because this falsified literal already increases LB by 1. So, κ and its literals are unlocked if one of its literals is falsified upon extending α. Since we do not know in advance which literal will be falsified by an extension of α, we must examine each $l \in \kappa$. If a core is detected for each l assumed to be falsified, the union of these cores and κ contains more than one falsified literal for any extension of α, allowing to increase LB. Moreover, we also introduced a Reinforcement Learning (RL) strategy to effectively reduce the computational overhead and improve the efficiency of FLA.

3 Experiments

The benchmark MaxSAT instances are collected from the unweighted tracks of the MaxSAT Evaluations (MSE) 2019–2024. We implemented the proposed FLA algorithm on an enhanced version of MaxCDCL [4] (`enMaxCDCL`) from MSE 2024. We then compared `enMaxCDCL`, `enMaxCDCL+FLA`, and other state-of-the-art MaxSAT solvers from MSE 2024. The evaluation results show that FLA enables enMaxCDCL to solve 14 additional instances. Figure 1 represents the relative advantage of `enMaxCDCL+FLA` over `enMaxCDCL` becomes increasingly pronounced on harder instances requiring more than 3000 seconds of solving time. Specifically, `enMaxCDCL` and `enMaxCDCL+FLA` solve 1560 and 1565 instances within 1000 seconds, respectively. In contrast, for $3000s < t < 3600s$, they solve 4 and 13 instances, respectively. While FLA helps solve only 5 additional instances in the easier region ($t < 1000s$), it enables the solver to resolve 9 additional instances in the harder region ($t > 3000s$).

4 Conclusion

In this paper, we proposed a *further lookahead* (FLA) algorithm to improve lower bound estimation in Branch-and-Bound (BnB) MaxSAT solvers. The FLA algorithm identifies additional disjoint inconsistent subformulas (cores) by exploiting the structural information from pre-detected cores. Consequently, the accumulation of these newly identified disjoint cores yields a tighter lower bound, significantly enhancing the pruning efficiency of BnB solvers. For future work, we plan to extend the FLA algorithm to Weighted Partial MaxSAT.

References

1. David Allouche, Isabelle André, Sophie Barbe, Jessica Davies, Simon de Givry, George Katsirelos, Barry O'Sullivan, Steve Prestwich, Thomas Schiex, and Seydou Traoré. Computational protein design as an optimization problem. *Artificial Intelligence*, 212:59–79, 2014.
2. Sami Cherif, Heythem Sattoutah, Chu-Min Li, Corinne Lucet, and Laure Brisoux Devendeville. Minimizing working-group conflicts in conference session scheduling through maximum satisfiability. In Paul Shaw, editor, *30th International Conference on Principles and Practice of Constraint Programming, CP 2024, September 2-6, 2024, Girona, Spain*, volume 307 of *LIPIcs*, pages 34:1–34:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.CP.2024.34.
3. Hao Hu, Marie-José Huguet, and Mohamed Siala. Optimizing Binary Decision Diagrams with MaxSAT for Classification. In *36th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, February 2022.
4. Chu Min Li, Shuolin Li, Jordi Coll, Djamal Habet, Felip Manyà, and Kun He. Maxcdcl in maxsat evaluation 2024. *MaxSAT Evaluation 2024 Solver and Benchmark Descriptions*, pages 15–16, 2024.
5. Chu Min Li and Zhe Quan. An efficient branch-and-bound algorithm based on maxsat for the maximum clique problem. In *Proceedings of AAAI 2010*, pages 128–133, 2010.
6. Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamal Habet, and Kun He. Combining Clause Learning and Branch and Bound for MaxSAT. In Laurent D. Michel, editor, *CP 2021*, volume 210 of *LIPIcs*, pages 38:1–38:18, Dagstuhl, Germany, 2021. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CP.2021.38, doi:10.4230/LIPIcs.CP.2021.38.
7. Shuolin Li, Chu-Min Li, Jordi Coll, Djamal Habet, and Felip Manyà. Improving the lower bound in branch-and-bound algorithms for maxsat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. doi:10.1609/aaai.v39i11.33226.
8. Sean Safarpour, Hratch Mangassarian, Andreas Veneris, Mark H. Liffiton, and Karem A. Sakallah. Improved design debugging using maximum satisfiability. In *Formal Methods in Computer Aided Design (FMCAD'07)*, pages 13–19, 2007. doi:10.1109/FAMCAD.2007.26.