

EvalMaxSAT 2023

Florent Avellaneda

Université du Québec à Montréal
Montréal, Canada

Email: avellaneda.florent@uqam.ca

Centre de Recherche de l'IUGM
Montréal, Canada

Email: florent.avellaneda@gmail.com

INTRODUCTION

EvalMaxSAT is a MaxSAT solver written in modern C++ language mainly using the Standard Template Library (STL). The solver is built on top of the SAT solver CaDiCaL [1], but any other SAT solver can easily be used instead. EvalMaxSAT is based on the OLL algorithm [2] originally implemented in the MSCG MaxSAT solver [3], [4] and then reused in the RC2 solver [5].

Here, the significant modifications made in this new version of EvalMaxSAT are presented. For a general description of how EvalMaxSAT functions, please refer to [6].

A. Code Simplification

The first major modification is the simplification of the code and the replacement of the inheritance hierarchy with the use of templates. It is important to note that to further simplify the code, this new version is single-threaded.

B. SCIP Support

During the MSE 2022 competition, the solvers UW_rMaxSat [7] and Cashwmaxsat [8] showed good performance on a number of small problems by using a mixed-integer programming solver to solve them. Therefore, in this new version of EvalMaxSAT, the option to use the SCIP solver [9] to solve a formula has been added. In the version submitted to the competition, EvalMaxSAT with the SCIP solver allocates 500 seconds for instances with fewer than 100,000 variables.

C. Precalculation of UB with Incomplete Solver

A crucial technique employed by many MaxSAT solvers to tackle weighted MaxSAT formulas is the utilization of the hardening technique [10]. This technique involves deriving an upper bound on the cost of the optimal solution to deduce that certain high-weight soft clauses can be transformed into hard clauses. While the hardening technique is typically used in conjunction with a stratification technique [11], which allows for the determination of upper bounds from admissible solutions at different stratification levels, the process of obtaining high-quality upper bounds can be slow.

To address this drawback and improve the efficiency of EvalMaxSAT, the latest version introduces a precalculation phase using an incomplete solver. Specifically, EvalMaxSAT incorporates the Loandra [12] and Nuwls [13] solvers, running each for 2 minutes and 30 seconds before proceeding with the main solving process. These incomplete solvers aim to provide a higher-quality upper bound by exploring the search space and obtaining a partial solution within the given time limit.

D. Adaptive Research Core Times

The resolution time of MaxSAT solvers based on the OLL algorithm is heavily influenced by the quality of the considered unsat cores. In this regard, EvalMaxSAT distinguishes itself by dedicating significant time to optimizing the quality of unsat cores. It achieves this by making multiple calls to a SAT solver on the same formula, generating multiple unsat cores, and selecting the best one to retain [6]. Although the additional time spent searching for improved unsat cores is compensated in the long run by generating smaller cardinality cores, there needs to be a balance regarding the time allocated to optimizing unsat cores.

In this latest version of EvalMaxSAT, several criteria have been introduced to determine the time devoted to unsat core optimization, including:

- **Early stop:** After the 20th iteration, if the last improvement occurred before the $n/3$ iteration (where n represents the current iteration), the search process is halted. This mechanism prevents excessive time from being spent on optimizing unsat cores when the improvement rate becomes significantly slow. By stopping the search at this point, EvalMaxSAT avoids diminishing returns and focuses on other aspects of the solving process.
- **Diminishing optimization factor:** Since the expected benefit from discovering higher-quality unsat cores is proportional to the remaining computation time, a diminishing factor is applied to the time dedicated to unsat core optimization as time progresses.

CONCLUSION

In conclusion, this paper presented significant modifications made in the new version of EvalMaxSAT, including code simplification, SCIP solver support, precalculation of upper bounds, and adaptive research core times. These enhancements improve the efficiency and effectiveness of EvalMaxSAT in solving MaxSAT problems.

REFERENCES

- [1] A. Biere, K. Fazekas, M. Fleury, and M. Heisinger, "CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020," in *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2020-1. University of Helsinki, 2020, pp. 51–53.

- [2] A. Morgado, C. Dodaro, and J. Marques-Silva, “Core-guided MaxSAT with soft cardinality constraints,” in *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, ser. Lecture Notes in Computer Science, B. O’Sullivan, Ed., vol. 8656. Springer, 2014, pp. 564–573. [Online]. Available: https://doi.org/10.1007/978-3-319-10428-7_41
- [3] A. Morgado, A. Ignatiev, and J. Marques-Silva, “MSCG: robust core-guided maxsat solving,” *J. Satisf. Boolean Model. Comput.*, vol. 9, no. 1, pp. 129–134, 2014. [Online]. Available: <https://satassociation.org/jsat/index.php/jsat/article/view/127>
- [4] A. Ignatiev, A. Morgado, V. M. Manquinho, I. Lynce, and J. Marques-Silva, “Progression in maximum satisfiability,” in *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, ser. Frontiers in Artificial Intelligence and Applications, T. Schaub, G. Friedrich, and B. O’Sullivan, Eds., vol. 263. IOS Press, 2014, pp. 453–458. [Online]. Available: <https://doi.org/10.3233/978-1-61499-419-0-453>
- [5] A. Ignatiev, A. Morgado, and J. Marques-Silva, “RC2: an efficient maxsat solver,” *J. Satisf. Boolean Model. Comput.*, vol. 11, no. 1, pp. 53–64, 2019. [Online]. Available: <https://doi.org/10.3233/SAT190116>
- [6] F. Avellaneda, “A short description of the solver evalmaxsat,” *MaxSAT Evaluation*, vol. 8, 2020.
- [7] M. Piotrów, “Uwrmxat entering the maxsat evaluation 2022,” *MaxSAT Evaluation 2022*, p. 21.
- [8] Z. Lei, S. Cai, D. Wang, Y. Peng, F. Geng, D. Wan, Y. Deng, and P. Lu, “Cashwmaxsat: Solver description,” *MaxSAT Evaluation*, vol. 2021, p. 8, 2021.
- [9] K. Bestuzheva, M. Besançon, W.-K. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S. J. Maher, F. Matter, E. Mühmer, B. Müller, M. E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger, and J. Witzig, “The SCIP Optimization Suite 8.0,” Optimization Online, Technical Report, December 2021. [Online]. Available: http://www.optimization-online.org/DB_HTML/2021/12/8728.html
- [10] A. Morgado, F. Heras, and J. Marques-Silva, “Improvements to core-guided binary search for maxsat,” in *Theory and Applications of Satisfiability Testing—SAT 2012: 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings 15*. Springer, 2012, pp. 284–297.
- [11] C. Ansótegui, M. L. Bonet, J. Gabas, and J. Levy, “Improving wpm2 for (weighted) partial maxsat,” in *International Conference on Principles and Practice of Constraint Programming*. Springer, 2013, pp. 117–132.
- [12] J. Berg, “Loandra in the 2020 maxsat evaluation,” *MaxSAT Evaluation 2022*, vol. 2022, 2022.
- [13] Y. Chu, S. Cai, Z. Lei, and X. He, “Nuwlsc: Solver description,” *MaxSAT Evaluation 2022*, p. 28.