Pacose: An Iterative SAT-based MaxSAT Solver

Tobias Paxian, Bernd Becker
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 051
79110 Freiburg, Germany
{paxiant|becker}@informatik.uni-freiburg.de

I. Overview

Pacose is a SAT-based MaxSAT solver, using two incremental CNF encodings, a binary adder [1] and the Dynamic Polynomial Watchdog (DPW) [2], for Pseudo-Boolean (PB) constraints. It is an extension of QMaxSAT 2017 [3], based on Glucose 4.2.1 [4] SAT solver. It uses a Boolean Multilevel Optimization (BMO) pre- / inprocessing method to simplify the instances. Additionally a trimming method is applied to cut off unsatisfiable soft clauses and find a good initial satisfiable weight to reduce the size of the encoding.

II. Pre- / Inprocessing

The 2019 version of Pacose contains two new pre-/inprocessing methods, Generalized Boolean Multilevel Optimization (GBMO) and a trimming algorithm.

Multi-Objective Combinatorial Optimization (MOCO) [5] problems are addressing multiple optimization problems with possibly conflicting purposes. Boolean Multilevel Optimization (BMO) [6], [7] is the mapping of MOCO to MaxSAT solving. We generalized the plain variant of Boolean Multilevel Optimization thereby making it possible to split additional instances, even in cases where the weight differences of the sum of smaller weights is non-strictly smaller than the next biggest weight.

The trimming algorithm tries to satisfy each soft clause at least once with the additional goal to find a good approximation of the weight. It works in two phases, in the first phase it optimizes the overall weight and in the second phase it satisfies as many soft clauses as possible in the next solver call. After a timeout which is based on the number of soft clauses, it switches from the first phase to the second. An additional timeout for each incrementally solver call is included.

III. Encoding and Algorithm

Our DPW encoding is based on the Polynomial Watchdog (PW) encoding [8], which uses totalizer networks [9]. Essentially the DPW encoding employs multiple totalizer networks to perform a binary addition with carry on the sorted outputs. A special algorithm to solve these instances incremental is presented in [2].

Additionally the adder network [1] is used which has a linear complexity in encoding size in contrast to at least $\mathcal{O}(n^2)$ for the DPW sorting network. With the adder network many complementary instances to the DPW encoding can be solved and therefore it is well suited, to be chosen, together with DPW by a heuristic, as described in the following chapter. The algorithm and encoding are partly adapted and inspired from QMaxSAT.

This work is supported by DFG “Algebraic Fault Attacks” (BE 1176/20-2)

IV. Heuristics

Pacose uses straightforward heuristics based on available MaxSAT benchmarks. All heuristics are based on the number of soft clauses and the overall sum of soft weights.

- *Encoding:* The DPW encoding empirically works best if the average weight for soft clauses is small, or the overall sum of soft weights is huge (bigger than 80 billion). For the other benchmarks the binary adder is chosen.
- *Trimming:* As for instances with only a few soft clauses the trimming preprocessing algorithm is not effective, it is only used if the benchmark contains at least a certain amount of soft clauses.
- *Compression Rate:* For benchmarks with only a few soft clauses, the encoding is smaller and additional clauses can be added. Therefore the binary adder encoding can solve overall more benchmarks if the compression rate is chosen accordingly.

References

[1] J. P. Warners, “A linear-time transformation of linear inequalities into conjunctive normal form,” *Information Processing Letters*, vol. 68, no. 2, pp. 63–69, 1998.

[2] T. Paxian, S. Reimer, and B. Becker, “Dynamic polynomial watchdog encoding for solving weighted MaxSAT,” in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2018, pp. 37–53.

[3] M. Koshimura, T. Zhang, H. Fujita, and R. Hasegawa, “QMaxSAT: A partial Max-SAT solver system description,” *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 8, pp. 95–100, 2012.

[4] G. Audemard and L. Simon, “On the glucose SAT solver,” *International Journal on Artificial Intelligence Tools*, vol. 27, no. 01, p. 1840001, 2018.

[5] E. L. Ulungu and J. Teghem, “Multi-objective combinatorial optimization problems: A survey,” *Journal of Multi-Criteria Decision Analysis*, vol. 3, no. 2, pp. 83–104, 1994.

[6] J. Argelich, I. Lynce, and J. Marques-Silva, “On solving boolean multilevel optimization problems,” in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[7] J. Marques-Silva, J. Argelich, A. Graça, and I. Lynce, “Boolean lexicographic optimization: algorithms & applications,” *Annals of Mathematics and Artificial Intelligence*, vol. 62, no. 3-4, pp. 317–343, 2011.

[8] O. Bailleux, Y. Boufkhad, and O. Roussel, “New encodings of pseudo-boolean constraints into CNF,” in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2009, pp. 181–194.

[9] O. Bailleux and Y. Boufkhad, “Efficient CNF encoding of Boolean cardinality constraints,” in *Principles and Practice of Constraint Programming–CP 2003*. Springer, 2003, pp. 108–122.