UWrMaxSat Entering the MaxSAT Evaluation 2022

Marek Piotrów
Institute of Computer Science, University of Wrocław
Wrocław, Poland
marek.piotrow@uwr.edu.pl

***Abstract*—UWrMaxSat is a complete solver for partial weighted MaxSAT instances and pseudo-Boolean ones. It can be also characterized as an anytime solver, since it outputs the best known solution, whenever its run is interrupted. It needs a SAT solver as an oracle and can be used with a few modern solvers, from which COMiniSatPS by Chanseok Oh (2016) has been selected as a default one. Several solving strategies have been implemented in it (selected by parameters) but the default one is a core-guided OLL procedure, where its own sorter-based pseudo-Boolean constraint encoding is applied to translate cardinality constraints into CNF. This paper describes new elements in UWrMaxSat version 1.4, which is submitted to the MaxSAT Evaluation 2022. They include (1) the IPAMIR interface that standardizes an access to a MaxSAT solver as a library, and (2) an integration with SCIP solver for mixed integer programming.**

***Index Terms*—MaxSAT-solver, UWrMaxSat, COMiniSatPS, sorter-based encoding, core-guided, complete solver**

I. Introduction

An example of optimization problems can be represented as a partial weighted MaxSAT instance or, equivalently, as a set of linear inequation over Boolean variables, called pseudo-Boolean (PB), with a linear goal function, which value should be minimized. A MaxSAT instance consists of two sets containing hard clauses and weighted soft ones, respectively. Hard ones must be satisfied by any solution and the goal of MaxSAT optimization is to find a model that minimizes also the sum of weights of unsatisfied soft ones. It is clear that clauses can be easily converted into linear inequations over Boolean variables. Inverse translations are used in many PB solvers, but one needs an efficient encoder to convert PB constraints into clauses.

UWrMaxSat was created as an extension of the MiniSat+ 1.1 solver by Eén and Sörensson (2012) [4] in two basic steps: firstly, this PB solver was extended with a new sorter-based encoder by Michał Karpiński and Marek Piotrów to, so called, KP-MiniSat+ [6], [7], and then, a MaxSAT parser and the corresponding solving techniques were added to it by Piotrów and the solver was renamed to UWrMaxSat. Therefore, it is able to solve both PB and MaxSAT instances. Moreover, it appeared that the implemented translation of an PB instance into an equivalent MaxSAT one and the selected MaxSAT algorithm become an efficient way to solve many PB examples of optimization problems [8].

But, first of all, UWrMaxSat is a competitive complete MaxSAT solver, as it was shown by results od MaxSAT Evaluations (MSE) in the last three years. Furthermore, outcomes of the complete tracks of MSE 2021 indicated that translating clauses into linear inequation over 0-1 integer variables and using the mixed integer programming solver SCIP [3] to solve them can be a successful way to find the optimal solution of several small MaxSAT instances. Therefore, this method has been added as an option to UWrMaxSat.

Furthermore, this year's version is submitted with two new features: (1) the IPAMIR interface that has been defined to standardize the way of using a solver as a library, and (2) an improved greedy algorithm of the encoder module to better check if some of previously encoded sorting networks can be reused in a new encoding.

II. Description

A new version of UWrMaxSat is denoted as 1.4 and it is a fourth time when the solver takes part in MaxSAT Evaluations. For the main features of the previous versions see [13], [14], [16]. A more detailed description of UWrMaxSat ver. 1.1 can be found in [15]. In this year's version, we continue to use incrementally the SAT solver COMiniSatPS by Chanseok Oh (2016) [12] as an oracle. The default search strategy for the optimal solution is also the same as in previous years, that is, a core-guided linear one, where unsatisfiability cores are processed by the OLL procedure [1], [5], [10] and cardinality constraints generated by it are encoded with the help of 4-Way Merge Selection Networks [7] and Direct Networks [2]. The general description of search strategies used by MaxSAT solvers can be found, for example, in [11].

UWrMaxSat can be compiled with or without two additional libraries: the extended MaxSAT preprocessor MaxPre created at the University of Helsinki and implemented by Tuukka Korhonen [9], and the mixed integer programming solver SCIP created in cooperation of several organizations [3]. The submitted UWrMaxSat 1.4 uses only the second library in version 8.0.0. The default configuration is to run the SCIP solver on small MaxSAT instances in a separate thread without given timeout. Using the corresponding options one can force it to be run in the same thread and for a defined number of seconds before the start of the MaxSAT solver (these options are set in the competition). An input instance is preprocessed before sending it to SCIP by the UWrMaxSat's implementation of an algorithm to detect unit clauses and at-most-one cardinality constraints and, then, by simplifications done by the SAT solver.

The recently-defined IPAMIR interface has been added to UWrMaxSat to standardize the API of the corresponding library. All elements of IPAMIR have been implemented. The only restriction is that the library allows an application th use a single instance of the solver at a time. In version 1.4 of the library, MaxPre is not used and should be compiled without it. On the other hand, it can be compiled to be used with the SCIP library and the corresponding options can be set in an environment variable UWRFLAGS ()as well as some other useful ones of UWrMaxSat).

A sorter-based encoding of PB constraints was implemented in the original MiniSat+ solver. Next, it was replaced by a much more complicated one in KP-MiniSat+, but still each constraint was encoded separately into clauses. Recently, Michał Karpiński and me have proposed an improvement of such technique and implemented it in UWrMaxSat. By using a variation of a greedy set cover algorithm, when adding constraints to our encoding, the encoder reuses parts of the already encoded PB-instance in order to decrease the size (the number of variables and clauses) of the resulting SAT instance [8].

Finally, the parser of UWrMaxSat was modified to accept the new proposed input format of partial weighted MaxSAT instances. Recall that there is no p-line in it and hard clauses are preceded by the letter 'h' instead of the "very big" weight given in a p-line.

References

[1] Benjamin Andres, Benjamin Kaufmann, Oliver Matheis, and Torsten Schaub. Unsatisfiability-based optimization in clasp. In *Technical Communications of the Twenty-eighth International Conference on Logic Programming (ICLP'12)*, volume 17, pages 212–221, 2012.

[2] Roberto Asín, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. Cardinality networks: a theoretical and empirical study. *Constraints*, 16(2):195–221, 2011.

[3] 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.

[4] Niklas Eén and Niklas Sörensson. Translating pseudo-boolean constraints into sat. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–26, 2006.

[5] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. Rc2: an efficient maxsat solver. *Journal on Satisfiability, Boolean Modeling and Computation*, pages 53–64, 2019.

[6] Michał Karpiński and Marek Piotrów. Competitive sorter-based encoding of pb-constraints into sat. In Daniel Le Berre and Matti Järvisalo, editors, *Proc. Pragmatics of SAT 2015 and 2018*, volume 59 of *EPiC Series in Computing*, pages 65–78. EasyChair, 2019.

[7] Michał Karpiński and Marek Piotrów. Encoding cardinality constraints using multiway merge selection networks. *Constraints*, Apr 2019.

[8] Michał Karpiński and Marek Piotrów. Reusing comparator networks in pseudo-boolean encodings. *arXiv preprint arXiv:2205.04129*, 2022.

[9] Tuukka Korhonen, Jeremias Berg, Paul Saikko, and Matti Järvisalo. MaxPre: An extended MaxSAT preprocessor. In Serge Gaspers and Toby Walsh, editors, *Proc. SAT*, volume 10491 of *LNCS*, pages 449–456. Springer, 2017.

[10] Antonio Morgado, Carmine Dodaro, and Joao Marques-Silva. Core-guided maxsat with soft cardinality constraints. In Barry O'Sullivan, editor, *Proc. CP*, LNCS, pages 564–573. Springer, 2014.

[11] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and João Marques-Silva. Iterative and core-guided maxsat solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013.

[12] Chanseok Oh. *Improving SAT Solvers by Exploiting Empirical Characteristics of CDCL.* PhD thesis, New York University, 2016.

[13] Marek Piotrów. Uwrmaxsat - a new minisat+-based solver in maxsat evaluation 2019. In Fahiem Bacchus, Matti Järvisalo, and Ruben Martins, editors, *MaxSAT Evaluation 2019 : Solver and Benchmark Descriptions*, Department of Computer Science Report Series B-2019-2, pages 11–12, 2019.

[14] Marek Piotrów. Uwrmaxsat - an efficient solver in maxsat evaluation 2020. In Fahiem Bacchus, Jeremias Berg, Matti Järvisalo, and Ruben Martins, editors, *MaxSAT Evaluation 2020 : Solver and Benchmark Descriptions*, Department of Computer Science Report Series B-2020-2, pages 34–35, 2020.

[15] Marek Piotrów. Uwrmaxsat: Efficient solver for maxsat and pseudo-boolean problems. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 132–136, 2020.

[16] Marek Piotrów. Uwrmaxsat in maxsat evaluation 2021. In Fahiem Bacchus, Jeremias Berg, Matti Järvisalo, and Ruben Martins, editors, *MaxSAT Evaluation 2021 : Solver and Benchmark Descriptions*, Department of Computer Science Report Series B-2021-2, pages 17–18, 2021.