Open-WBO-Inc in MaxSAT Evaluation 2018

Saurabh Joshi[‡], Prateek Kumar[‡], Vasco Manquinho[*], Ruben Martins[†], Alexander Nadel[⋆], Sukrut Rao[‡]
[‡]Indian Institute of Technology Hyderabad, India
[*]INESC-ID / Instituto Superior Técnico, Universidade de Lisboa, Portugal
[†]Carnegie Mellon University, USA
[⋆]Intel Corporation, Israel

I. Introduction

Open-WBO-Inc is developed on top of Open-WBO [1], [2], [3], which is one of the best solvers in the MaxSAT Evaluations of 2014–2017. For many applications that can be encoded into MaxSAT, it is important to quickly find solutions even though these may not be optimal. Open-WBO-Inc is designed to find a good solution[1] in a short amount of time. Open-WBO-Inc starts with an incomplete stage where it is not guaranteed to converge to an optimal solution. Once this stage is completed, we switch to a complete algorithm that can further improve the solution and eventually find the optimal solution. Since Open-WBO-Inc is based on Open-WBO, it can use any MiniSAT-like solver [4]. For this evaluation we use Glucose 4.1 [5] as our back-end SAT solver.

II. Unweighted Incomplete MaxSAT

For unweighted incomplete MaxSAT, we submitted two versions: Open-WBO-Inc-MCS and Open-WBO-Inc-OBV. The first version is based on Minimal Correction Subset (MCS) enumeration. A MCS of an unsatisfiable set of constraints is a minimal subset that, if removed, makes the constraint set satisfiable. We use a linear search algorithm [6] to enumerate MCSes. We impose a limit of 100,000 conflicts or a maximum of 30 MCSes when enumerating MCSes. Once this limit is reached or all MCSes are found, the solver will continue its search using a complete linear search algorithm SAT-UNSAT (LSU) [7] for MaxSAT starting from the best upper bound value found by MCS enumeration.

Open-WBO-Inc-OBV is based on bit-vector optimization and follows a similary strategy to the incomplete approach used in Mrs. Beaver [8]. This approach operates over a vector $\mathcal{T}$ that represents the relaxation variables introduced in each soft clause. We run 100 iterations of the following loop:

- Run the UMS-OBV-BS algorithm;
- Reverse $\mathcal{T}$. Run another UMS-OBV-BS iteration;
- Reverse $\mathcal{T}$. Run the OBV-BS algorithm;
- Reverse $\mathcal{T}$. Run another OBV-BS iteration.

At the end of each loop we randomly shuffle the vector $\mathcal{T}$. We also impose a limit of 10,000 conflicts when calling the UMS-OBV-BS and OBV-BS algorithms. For a detailed description of these algorithms we refer the reader to Mrs. Beaver paper [8]. If this algorithm terminates the incomplete stage, we continue the search by using LSU algorithm [7] for MaxSAT starting from the best upper bound value found by the bit-vector optimization stage.

To restrict the upper bound at each iteration for the LSU algorithm, we need to encode cardinality constraints into CNF. Both Open-WBO-Inc-MCS and Open-WBO-Inc-OBV versions use the Modulo Totalizer encoding [9] for cardinality constraints.

III. Weighted Incomplete MaxSAT

For weighted incomplete MaxSAT, we submitted two versions: Open-WBO-Inc-Cluster and Open-WBO-Inc-BMO. Open-WBO-Inc-Cluster uses a technique described in [10] where it partitions the clauses in clusters and all the clauses in a cluster are given a weight equal to the representative weight of the cluster, which is a function of original weights of the clauses in the cluster. For the purpose of MaxSAT Evaluation 2018, we use arithmetic mean of the weights of clauses as representative weight of the cluster. The number of clusters is set to 2 for the purpose of this evaluation, as it is reported to strike a good balance between formula size and precision [10]. Open-WBO-Inc-Cluster uses LSU algorithm [7] with the modified weights after clustering. It uses the Generalized Totalizer Encoding (GTE) [11] to encode Pseudo-Boolean constraints that are generated to restrict weighted sum of the unsatisfied soft clauses. If an optimal solution is found for the modified MaxSAT instance, this will be an upper bound of the original MaxSAT instance. When this occurs, we revert the weights to the original weights and resume the search using the LSU algorithm starting from the best known solution.

Open-WBO-Inc-BMO version is based on bounded multi-level optimization [12] using a variant of linear search algorithm SAT-UNSAT [7] along with the partitioning of clauses as described earlier [10]. The algorithm used in Open-WBO-Inc-BMO performs optimization on each cluster in the descending order of its representative weight. This is done by performing a sequence of calls to a SAT solver and refining an upper bound μ on the number of unsatisfied soft clauses. To restrict μ at each iteration, we need to encode cardinality constraints into CNF, for which, incremental Totalizer encoding [2] has been used. Once for a given cluster the upper bound μ cannot be improved, it is frozen, and the next cluster in the order is optimized. For the purpose of MaxSAT Evaluation 2018, we set the number of clusters to the total number of different weights of the clauses of the input formula. Therefore, the

[1]By "good solution" we mean that it can be potentially suboptimal but is not far from the optimal solution.

representative weight and the original weight remains the same in this case. As in Open-WBO-Inc-Cluster, if an optimal solution is found using this algorithm, then it is not necessarily an optimal solution of the input formula. When this occurs, we keep the best known solution and resume the search using the LSU algorithm which can potentially find better solutions and prove optimality.

IV. Availability

We submit the source of Open-WBO-Inc as part of our submissions to the MaxSAT Evaluations 2018. The code will be later integrated into the main release of Open-WBO available under a MIT license in GitHub at *https://github.com/sat-group/open-wbo*.

Acknowledgments

We would like to thank Laurent Simon and Gilles Audemard for allowing us to use Glucose in the MaxSAT Evaluation. We would also like to thank Mikoláš Janota, Inês Lynce and Miguel Terra-Neves for their authorship and contributions to Open-WBO on which Open-WBO-Inc is based.

References

[1] R. Martins, V. Manquinho, and I. Lynce, "Open-WBO: a Modular MaxSAT Solver," in *SAT*, ser. LNCS, vol. 8561. Springer, 2014, pp. 438–445.

[2] R. Martins, S. Joshi, V. Manquinho, and I. Lynce, "Incremental Cardinality Constraints for MaxSAT," in *CP*. Springer, 2014, pp. 531–548.

[3] M. Neves, R. Martins, M. Janota, I. Lynce, and V. Manquinho, "Exploiting Resolution-Based Representations for MaxSAT Solving," in *SAT*. Springer, 2015, pp. 272–286.

[4] N. Eén and N. Sörensson, "An Extensible SAT-solver," in *SAT*. Springer, 2003, pp. 502–518.

[5] G. Audemard and L. Simon, "Predicting Learnt Clauses Quality in Modern SAT Solvers," in *IJCAI*, 2009, pp. 399–404.

[6] J. Bailey and P. J. Stuckey, "Discovery of Minimal Unsatisfiable Subsets of Constraints Using Hitting Set Dualization," in *PADL*. Springer, 2005, pp. 174–186.

[7] D. Le Berre and A. Parrain, "The Sat4j library, release 2.2," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 7, no. 2-3, pp. 59–6, 2010.

[8] A. Nadel, "Solving MaxSAT with Bit-Vector Optimization," in *SAT*. Springer, 2018.

[9] T. Ogawa, Y. Liu, R. Hasegawa, M. Koshimura, and H. Fujita, "Modulo Based CNF Encoding of Cardinality Constraints and Its Application to MaxSAT Solvers," in *ICTAI*. IEEE, 2013, pp. 9 – 17.

[10] S. Joshi, P. Kumar, R. Martins, and S. Rao, "Approximation Strategies for Incomplete MaxSAT," in *CP*. Springer, 2018.

[11] S. Joshi, R. Martins, and V. M. Manquinho, "Generalized Totalizer Encoding for Pseudo-Boolean Constraints," in *CP*. Springer, 2015, pp. 200–209.

[12] 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.