The MSUSorting MaxSAT solver

Eivind Jahren, Roberto Asín Achá

1 Solver Description

The MSUSorting solver builds on the work by Martins et al. [1] of leveraging incremental SAT solvers for the MSU3 algorithm [2], and the Totalizer encoding. The MSUSorting solver extends this work to the the mixed encoding by Abio et al. [3] and the MSU4 algorithm [4], using the glucose-syrup SAT solver [5].

2 Incremental MSU3 and MSU4 algorithms

The MSU3 algorithm is an unsatisfiable core based algorithm, akin to the Fu-Malik algorithm [6], [7], which uses a cardinality encoding to bound the number of unsatisfied clauses. The main difference for the incremental version of the algorithm is enabling the cardinality encoding to be updated. Martins et al. [1] uses the totalizer encoding [8] for this purpose.

We extend this work with new updatable cardinality encodings based on cardinality networks [9] and parametric cardinality networks [3]. We also use these updatable cardinality encodings to make the MSU4 algorithm [4] incremental. See [10] for details.

3 Updatable Cardinality Encodings

We use a generic framework for making updatable cardinality encodings which we call *delayed variables* [10]. This framework enables us to make updatable versions of the totalizer, cardinality network, and mixed encoding. A delayed variable is one which is not yet introduced to the SAT solver, so any clause it occurs in is not given to the SAT solver until the variable is undelayed. This allows delayed variables to be substituted without changing the formula given to the SAT solver.

4 Selecting strategies

The solver has two tweakable parameters: whether to choose MSU3 or MSU4, and whether to choose the cardinality networks or the totalizer encoding in the mixed encoding [3]. We found that many benchmarks can quickly be solved with either MSU3 or MSU4 but not by both. Since MSU3 and MSU4 share internal state, we simply switch to MSU4 once a time limit (500s) has been reached. This ensures that some time is spent solving the problem with both algorithms, and progress made with MSU3 is reused for MSU4.

The mixed encoding combines the totalizer and cardinality network encoding. Using the totalizer encoding encoding means fewer variables, while the cardinality network encoding has fewer clauses. We found that when using the mixed encoding with the MSU3 & MSU4 algorithms, the encoding should favor the totalizer encoding heavily. The solver is given a limit on the of number of extra clauses beyond the minimal amount, and uses totalizer as long as the budget is not exceeded. If the limit is exceeded, cardinality network is used where it saves the most clauses per additional variable until the limit is satisfied. The limit is quite generous: eight times the number of clauses in the input formula.

References

[1] R. Martins, S. Joshi, V. Manquinho, and I. Lynce, "Incremental cardinality constraints for maxsat," in *Principles and Practice of Constraint Programming*, B. O'Sullivan, Ed. Springer, 2014, pp. 531–548.

[2] J. Marques-Silva and J. Planes, "On using unsatisfiability for solving maximum satisfiability," *arXiv preprint arXiv:0712.1097*, 2007.

[3] I. Abío, R. Nieuwenhuis, A. Oliveras, and E. Rodríguez-Carbonell, "A parametric approach for smaller and better encodings of cardinality constraints," in *Principles and Practice of Constraint Programming*, C. Schulte, Ed. Springer, 2013, pp. 80–96.

[4] J. Marques-Sila and J. Planes, "Algorithms for maximum satisfiability using unsatisfiable cores," in *Advanced Techniques in Logic Synthesis, Optimizations and Applications*, K. Gulati, Ed. Springer, 2011, pp. 171–182.

[5] G. Audemard and L. Simon, "Predicting learnt clauses quality in modern sat solvers." in *Proceedings of the Twenty-First International Joint Conference On Artificial Intelligence*, vol. 3. IJCAI, 2009, pp. 399–404.

[6] Z. Fu and S. Malik, "On solving the partial max-sat problem," in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2006, pp. 252–265.

[7] Z. Fu, *Extending the power of Boolean satisfiability solvers: Techniques and applications*. Princeton University, 2007.

[8] O. Bailleux and Y. Boufkhad, "Efficient cnf encoding of boolean cardinality constraints," in *Principles and Practice of Constraint Programming*, F. Rossi, Ed. Springer, 2003, pp. 108–122.

[9] R. Asín, R. Nieuwenhuis, A. Oliveras, and E. Rodríguez-Carbonell, "Cardinality networks: a theoretical and empirical study," *Constraints*, vol. 16, no. 2, pp. 195–221, 2011.

[10] E. Jahren and R. Asín Achá, "Resizing cardinality constraints for maxsat," *Manuscript submitted for publication*, 2017.