

# MaxHS in the 2018 MaxSat Evaluation

Fahiem Bacchus  
*Department of Computer Science*  
*University of Toronto*  
*Ontario, Canada*  
*Email: fbacchus@cs.toronto.edu*

## 1. MaxHS

MaxHS is a MaxSat solver that originated in the PhD work of Davies [4]. It was the first MaxSat solver to utilize the Implicit Hitting Set (IHS) approach, and its core components are described in [4], [2], [3], [5]. Additional useful insights into IHS are provided in [6], [7]. IHS solvers utilize both an integer programming (IP) solver and a SAT solver in a hybrid approach to MaxSat solving. MaxHS utilizes minisat v2.2 as its SAT solver and IBM's CPLEX v12.8 as its IP solver. Interestingly experiments with more sophisticated SAT solvers like Glucose <http://www.labri.fr/perso/lsimon/glucose/> and Lingeling <http://fmv.jku.at/lingeling/> yielded inferior performance. This indicates that the SAT problems being solved are quite simple, too simple for the more sophisticated techniques used in these SAT solvers to pay off. In fact, simpler SAT problems are one of the original motivations behind MaxHS [2].

The 2018 version of MaxHS is unchanged from the 2017 submission, with the exception that the newest CPLEX-12.8 is being used. To make this document more self contained we repeat here the main features of v3.0, as compared to the prior published descriptions of MaxHS are as follows (familiarity with the basics of the IHS approach is assumed).

**1.0.1. Termination based on Bounding.** MaxHS v3.0 maintains an upper bound (and best model found so far) and a lower bound on the cost of an optimal solution (the IP solver computes valid lower bounds). MaxHS terminates when the gap between the lower bound and upper bound is low enough (with integer weights when this gap is less than 1, the upper bound model is optimal). This means that MaxHS no longer needs to wait until the IP solver returns an hitting set whose removal from the set of soft clauses yields SAT; it can return when the IP solver's best lower bound is close enough to show that the best model is optimal.

**1.0.2. Early Termination of Cplex.** In previous versions of MaxHS, the IP solver was run to completion forcing it to find an optimal solution every time it is called. However, with bounding, optimal solutions are not always needed. In particular, if the IP solver finds a feasible solution whose cost is better than the current best model it can return that: either the IP solution is feasible for the MaxSat problem, in

which case we can lower the upper bound, or it is infeasible in which case we can obtain additional cores to augment the IP model (and thus increase the lower bound). Terminating the IP solver before optimization is complete can yield significant time savings.

**1.0.3. Reduced Cost fixing via the LP-Relaxation.** Using an LP relaxation and the reduced costs associated with the optimal LP solution, some soft clauses can be hardened or immediately falsified. See [1] for more details.

**1.0.4. Mutually Exclusive Soft Clauses.** Sets of soft clauses of which at most one can be falsified or at most one can be satisfied are detected. When all of these soft clauses have the same weight they can all be more compactly encoded with a single soft clause. This encoding does not always yield better performance due to some subtle effects. However, techniques were developed to better exploit such information, and a fuller description of these techniques is in preparation. With these techniques performance gains were achieved.

**1.0.5. Other clauses to the IP Solver.** Problems with a small number of variables are given entirely to the IP solver, so that it directly solves the MaxSat problem. In this case the SAT solver is used to first compute some additional clauses and cores, and to find a better initial model for the IP solver. This additional information from the SAT solver often makes the IP solver much faster than just running the IP solver and represents an alternate way of hybridizing SAT and IP solvers.

**1.0.6. Other techniques for finding Cores.** MaxHS iteratively calls the IP solver to obtain a hitting set of the cores computed so far. If that hitting set does not yield an optimal MaxSat solution then more cores must be added to the IP solver. In some of these iterations very few cores can be found causing only a slight improvement to the IP solver's model. This results in a large number of time consuming calls to the IP solver. Two methods were developed to aid this situation (a) we ask the IP solver for more solutions and generate cores from these as hitting sets as well and (b) if we have a new upper bound model we try to improve this model by converting it to a minimal correction set (MCS). In

converting the upper bound model to an MCS we either find a better model (lowering the upper bound) or we compute additional conflicts that can be added to the IP solver.

**1.0.7. Incomplete MaxSat Solving.** The solver maintains upper bounding models as described above, and in its normal operation it terminates only when it is able to prove that its best model is in fact optimal. However, often it is able to find very good upper bounding models or even optimal models long before termination (proving a model to be optimal is generally as hard or even harder than finding it). For the incomplete track we simply output the best model found so far at timeout.

## References

- [1] Bacchus, F., Hyttinen, A., Järvisalo, M., Saikko, P.: Reduced cost fixing in maxsat. In: Proc. CP. p. in press (2017)
- [2] Davies, J., Bacchus, F.: Solving MAXSAT by solving a sequence of simpler SAT instances. In: Proc. CP. Lecture Notes in Computer Science, vol. 6876, pp. 225–239. Springer (2011)
- [3] Davies, J., Bacchus, F.: Exploiting the power of MIP solvers in MaxSAT. In: Proc. SAT. Lecture Notes in Computer Science, vol. 7962, pp. 166–181. Springer (2013)
- [4] Davies, J.: Solving MAXSAT by Decoupling Optimization and Satisfaction. Ph.D. thesis, University of Toronto (2013), [http://www.cs.toronto.edu/~jdavies/Davies\\_Jessica\\_E\\_201311\\_PhD\\_thesis.pdf](http://www.cs.toronto.edu/~jdavies/Davies_Jessica_E_201311_PhD_thesis.pdf)
- [5] Davies, J., Bacchus, F.: Postponing optimization to speed up MAXSAT solving. In: Proc. CP. Lecture Notes in Computer Science, vol. 8124, pp. 247–262. Springer (2013)
- [6] Saikko, P., Berg, J., Järvisalo, M.: LMHS: A SAT-IP hybrid MaxSAT solver. In: Proc. SAT. Lecture Notes in Computer Science, vol. 9710, pp. 539–546. Springer (2016)
- [7] Saikko, P.: Re-implementing and Extending a Hybrid SAT-IP Approach to Maximum Satisfiability. Master's thesis, University of Helsinki (2015), <http://hdl.handle.net/10138/159186>