The Stochastic Local Search Solver: SSA

Robert Stelzmann
Knowledge Representation and Reasoning Group
Technische Universitt Dresden, 01062 Dresden, Germany

***Abstract*—This document briefly describes the SLS solver SSA in the configuration it has been submitted to the SAT Challenge 2012.**

I. Major solving techniques

The solver SSA is an implementation of the SPARROW [1] algorithm with some minor enhancements.

II. Parameter description

We basically took the whole parameter setup from the reference SPARROW implementation. Further we slightly adapted the parameter c_3 on 3-SAT instances to 200000, since it seems that the original setting of 100000 is not optimal on large 3-SAT instances.

III. Special Algorithms, Data Structures and Features

We replaced the list data structure used by UBCSAT SPARROW [2] to maintain the promising variables by a more efficient heap. Consequently we got a performance boost on large 3-SAT instances regarding to the number of flips per second.

IV. Implementation detail

The complete solver is completely written from scratch. The primary goal was the development of an easy applicable and robust framework to investigate new procedures and explore different configurations and parameter setups in the future. The solver is highly modularized and allows an easy exchange and extension of single components like the variable selection heuristic or the clause weighting strategy. Still, solving performance is a design criterion - the solver should be competitive with state-of-the-art SLS solver. As an implementation of SPARROW , the solver is capable of the follow state-of-the-art SLS techniques:

- G2WSAT -like [3] greedy search
- PAWS -like [4] clause weighting
- SPARROW -like [1] selection heuristic

V. SAT Challenge 2012 Specifics

Our solver is written in C++11 and was built as a static -O3 64-bit binary by gcc 4.6.2 on a linux machine, running kernel 2.6.32. We submitted our solver to the random track.

VI. Availability

The SSA framework together with some documentation can be found at: http://tools.computational-logic.org.

Acknowledgment

The authors would like to thank Norbert Manthey for his advice.

References

[1] A. Balint and A. Fröhlich, "Improving stochastic local search for sat with a new probability distribution," in *Theory and Applications of Satisfiability Testing - SAT 2010*, ser. Lecture Notes in Computer Science, O. Strichman and S. Szeider, Eds. Springer Berlin / Heidelberg, 2010, vol. 6175, pp. 10–15, 10.1007/978-3-642-14186-7_3. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14186-7_3

[2] D. A. D. Tompkins and H. H. Hoos, "UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT," in *Revised Selected Papers from the Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, ser. Lecture Notes in Computer Science, H. Hoos and D. Mitchell, Eds., vol. 3542. Springer Berlin / Heidelberg, 2005, pp. 306–320.

[3] C. M. Li and W. Q. Huang, "Diversification and determinism in local search for satisfiability," in *Proceedings of the 8th international conference on Theory and Applications of Satisfiability Testing*, ser. SAT'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 158–172. [Online]. Available: http://dx.doi.org/10.1007/11499107_12

[4] J. Thornton, D. N. Pham, S. Bain, and V. Ferreira, "Additive versus multiplicative clause weighting for sat," in *Proceedings of the 19th national conference on Artifical intelligence*, ser. AAAI'04. AAAI Press, 2004, pp. 191–196. [Online]. Available: http://dl.acm.org/citation.cfm?id=1597148.1597181