CHBR_glucose

Seongsoo Moon
Graduate School of Information Science and Technology,
The University of Tokyo, Japan

Inaba Mary
Graduate School of Information Science and Technology,
The University of Tokyo, Japan

***Abstract*—We briefly introduce our solver CHBR_glucose, CHBR_glucose_tuned, tb_glucose and tc_glucose submitted to SAT-Competition 2016. All solvers are based on glucose3.0, and CHB, introduced at AAAI 2016, is implemented in CHBR_glucose, CHBR_glucose_tuned, and tc_glucose. CHBR_glucose_tuned is for entering the Glocose Hack track in the SAT Competition 2016.**

I. Introduction

Decision heuristic is one of the most important elements in modern SAT solvers. The most prominent method is VSIDS[1]. There were lots of attempts to surpass VSIDS [2] [3] [4], but VSIDS is still most popular decision heuristic because of its robustness.

Recently new branching heuristic CHB[5] was provised and it showed significant improvements for some benchmarks.

In our program, we implemented CHB and select decision heuristic between VSIDS and CHB dynamically.

When a variable is selected by the score of VSIDS a lot of ties happened. We added some scores to VSIDS to reduce ties, and select more valuable variable from ties.

II. CHB tuned

In CHB, each variable has Q score, and is updated using Equation as follows based on reinforcement learning.

$$Q[v] = (1 - \alpha)Q[v] + \alpha r_v$$

We've selected several parameters those would change running time a lot to tune CHB and tested. The initial value of α is set to 0.4 in original CHB, and we changed this to 0.7 based on our tests.

III. CHBR_glucose

We've noticed CHB works very well with small problems, but VSIDS performs better than CHB for big problems. So, we divided problems for 2 groups by the number of variables. As default decision heuristic, our program choose VSIDS. If the number of variables is under 15000, CHB is activated and used behalf of VSIDS.

IV. CHBR_glucose_tuned

We've tuned CHB parameters based on 24 combination tests. Some instances work better than default parameter values. We've changed initial value of α, minimum of α, and $multiplier$ for small problems.

$if(2000 < numberofvariables < 7000)$
$\alpha = 0.4, \alpha_{min} = 0.03, multiplier = 0.5$

V. tb_glucose

Ties happen frequently in VSIDS. To break these, we update VSIDS scores after we obtain learned clauses. After a clause is obtained, we add 1 / (LBD of a clause) for each variables in that clause. We call this TBVSIDS.

VI. tc_glucose

This is a hybrid version of CHBR_glucose and tb_glucose. We use TBVSIDS as a default decision heuristic and use CHB when the number of variables is under 15000.

References

[1] Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S. Chaff: Engineering an Efficient SAT Solver. In Proceedings of the 38th Design Automation Conference, pp 530–535, 2001.
[2] Dershowitz, Nachum and Hanna, Ziyad and Nadel, Alexander. A Clause-Based Heuristic for SAT Solvers. Theory and Applications of Satisfiability Testing, pp 46–60, 2005.
[3] Goldberg, Evgueni and Novikov, Yakov. BerkMin: A Fast and Robust Sat-Solver. Design, Automation, and Test in Europe, pp 465–478, 2008.
[4] L.Ryan. Efficient algorithms for clause-learning SAT solvers. Matser's thesis, Simon Fraser University, 2004.
[5] Hui Liang, J., Ganesh, V., Poupart, P., Czarnecki, K. Exponential Recency Weighted Average Branching Heuristic for SAT Solvers Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[6] Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. (2008).