Optsat, Abcdsat and Maple_simp : Speed up Solving Satisfiable Instances

Jingchao Chen
School of Informatics, Donghua University
2999 North Renmin Road, Songjiang District, Shanghai 201620, P. R. China
chen-jc@dhu.edu.cn

***Abstract*—In order to speed up solving satisfiable instanceswe use a simple rephasing technique to our previous SAT solvers such as abcdsat, maple_simp and optsat.**

I. Introduction

Decision variable phasing policies are very important in CDCL solvers. Here we use a simple phasing policy similar to the phasing policy of [1], [2] to improve the speed of our SAT solvers.

II. A Simplified Rephasing Technique

We use a technique similar to the rephasing technique in CaDiCaL [1] and Relaxed_LCMDCBD [2] to the speed of solvers on satisfiable instances. Unlike CaDiCaL and Relaxed_LCMDCBD, we delete the local search solver for rephasing. This technique is actually a extension of the bit-eoncoding phase selection strategy in [3]. The simplified rephasing technique used this year may be described as follows. When the CDCL search reaches a conflict, the trail stack has the partial assignment of variables. We apply the unit propagation of CDCL and the random flip to extend the partial assignment to a full assignment. We save three full assignments to handle the variable phase problem. When the trail stack reaches a maximal length, the extended full assignment is called max_trail assignment. We get the current partial assignment based on the trail stack every 1500 restarts, and extend it to the full assignment, which is called current full assignment. The current full assignment with minimized unsatisfied clauses is called the best assignment. We maintain always the update of max_trail assignment, current full assignment and best assignment. When selecting the phase of decision variables, we use max_trail assignment, current full assignment and best assignment with 30%, 30% and 10% probability, respectively. The probability that the other assignment such as all true, all false, random, reverse etc is used is 30%.

III. Maple_simp21

Maple_simp21 participates the main track. It is an improved version of Maple_simp20 [4]. The main difference between them is that Maple_simp21 adds the simplified rephasing technique mentioned above.

IV. Optsat *m21*

This solver is submitted to the main track. It is an improved version of Optsat *m20* [4]. Different from Optsat *m20*, Optsat *m21* eliminates the hyper binary resolution in-processing. Another difference is Optsat *m21* adds the simplified rephasing technique mentioned above.

V. Optsat *R21*

This solver is submitted to the main track. It is a recursive version of Optsat *m21* mentioned above. The difference between them is that Optsat *R21* adds in-processing, which includes subsumption and variable elimination. The in-processing here is implemented by calling recursively the pre-processing subroutine of Optsat *m21*.

VI. abcdsat *i21*

This solver is submitted to the incremental library track. It is an improved version of abcdsat *i20* [4]. Compared with abcdsat *i20*, abcdsat *i21* adjusts some parameters, and adds adds the simplified rephasing technique mentioned above.

VII. abcdsat *p21*

This solver is submitted to the parallel track. It is an improved version of abcdsat *p18* [5], rather than last yeras version abcdsat *p20* [4]. Abcdsat *p21* uses at most 64 threads. 61 out of 64 threads solve the subproblem $F \wedge p$, where p and F are a pivot and the original problem respectively. The other 3 threads solve either the original problem or the simplified problem. The other difference from abcdsat *p18* is that abcdsat *p21* contains the simplified rephasing technique mentioned above.

References

[1] Biere, A.: Cadical at the sat race 2019. SAT RACE 2019, p.8
[2] Zhang, X., Cai, S.: Relaxed Backtracking with Rephasing, Proceedings of SAT competition 2020, pp.15–16
[3] Chen J.: A bit-eoncoding phase selection strategy for satisfiability solvers, in Proceedings of Theory and Applicationsf Models of Computation (TAMC'2014), LNCS, vol. 8402, Chennai, India, 2014, pp. 158-167
[4] Chen, J.: Optsat, Abcdsat and Solvers Based on Simplified Data Structure and Hybrid Solving StrategiesProceedings of SAT competition 2020, pp. 25–26
[5] Chen J.: AbcdSAT and Glucose hack: Various Simplifications and Optimizations for CDCL SAT solvers, Proceedings of SAT Competition 2018, pp.10-12