Cadical_SCAVEL and its friends at the SAT Competition 2021

1st Zhihui Li, 2nd Guanfeng Wu, 3rd Yang Xu, 4th Huimin Fu
School of Mathematics
National-Local Joint Engineering Laboratory of System Credibility
Automatic Verification, Southwest Jiaotong University
Chengdu, China
lizhihui@swjtu.edu.cn, wgf1024@swjtu.edu.cn,
xuyang@swjtu.edu.cn, fhm6688@my.swjtu.edu.cn

Abstract— this document describes six solvers from Scavel: Cadical_SCAVEL01/02, Relaxed_LCMDCBDL_SCAVEL01/02, and two parallel solvers abcd_para18_Scavel and P-MCOMSPS-STR-32-SC at the SAT Competition 2021.

I. INTRODUCTION

The base solvers we used to implement our techniques are cadical-alluip-trail and Relaxed_LCMDCBDL_newTech, obtained from the SAT Competition 2020 [1]. Based on these two very competitive solvers, some minor changes mainly include the following technical solutions: active backtrack, randomization of 2-watched scheme[2],Highlight the role of the stochastic local search solver in the Inception phase, and the adjustment of the program flow with the previous centralized technology.

II. ALGORITHM AND IMPLEMENTATION DETAILS

A. Active Backtrack

When we look at the dynamic growth and backtracking /restart shortening of propagation sequences of a global perspective, we analyze the assignment queues propagated according to BCP that these sequences are initiated by both conflict-causing and non-conflict-causing decisions, at the same time, they are also initiated by continuous conflict-causing decisions and continuous non-conflict-causing decisions.

Based on the empirical analysis, we obtained the variation law of the general solution process of continuous conflict decision and continuous non-conflict decision, and carried out two improvements based on this representational quantity. According to the continuous non-conflict decision model, a new restart standard is determined, the concept of Active Backtrack is proposed, and an Active Backtrack module is added between conventional conflict backtracking and general quick restart. This is different from the existing Trail Saving [3].

B. Randomization of 2-watched Scheme

In a typical CDCL implementation, a data structure called 2-watched scheme is commonly used because the unit Propagation needs to detect unit clauses as efficiently as possible. In cases where more than a unit clause may be examined at the same time, the 2-watched scheme usually presents one of them in order. This order is formed by the clause literals order or the construction of Elements of watches by the previous propagation. Usually the order is fixed unless it is randomized periodically.

C. Highlight the role of the stochastic local search solver

As famous example of combining CDCL with SLS, CaDiCaL and Relaxed_LCMDCBDL use respectively the two stochastic local search (SLS) solvers (probSAT and ccnr) at a specific point in time to get the value of the argument close to the solution space. Usually the SLS solver is fast and efficient when solving hard small-scale problems. For a smaller sample （the number of variables is less than 3000）, We enhance the SLS solver solution time (the number of flips) in an attempt to obtain the result directly in the initial stage , rather than just providing the initial phase for all the variables.

D. External Restart Frame

Quick restart technology is very important, especially important to solving UNSAT incidents. As a major technical module of the CDCL framework, restart is triggered multiple times of the solver function. Before and after restart, the values of the variables score of the decision branch are not changed, and the 2-watched scheme usually presents one of them in order. This order is formed by the clause literals order or the construction of Elements of watches by the previous propagation. Usually the order for every literal 2-watches is fixed. We adjust the program flow with external restart frame to start the solution processes several times based on the number of restarts that have occurred by adding an outer loop around the solve function. Before the solve function is called, we increase the decision branches value of inactive variable, manage the size of three clauses set (core, Iter2 and local) boldly and randomize the existing 2-watched Scheme. They are mainly for eliminating the adverse effect of the former solution stage on the latter as far as possible.

III. SOLVERS

We add the learnt clause used frequency strategy[4] to other parallel solvers to see the effect of this strategy in other parallel solvers. We build our parallel Solvers based on P-MCOMSPS-STR-32[5] and abcdSATi20[6], So the name of parallel solvers are “ P-MCOMSPS-STR-32-SC” and “ abcd_para18_Scavel”.

The Cadical_SCAVEL01/02 solvers in this submission are small amount modifications of CaDiCaL [1] that participated in SAT competition 2020, which implement our techniques of II.C. Especially, Cadical_SCAVEL01 is based on CADICAL-SC2020, and Cadical_SCAVEL01 is based on adical-alluip-trail.

The Relaxed_LCMDCBDL_SCAVEL01/02 solvers in this submission are modifications of Relaxed_LCMDCBDL _newTech [1] that participated in SAT competition 2020. Especially, Relaxed_LCMDCBDL_SCAVEL01 implements our techniques of II. A, and Relaxed_LCMDCBDL_SCA-VEL02 represents our techniques of II. A ~ II. A D.

IV. ACKNOWLEDGMENTS

The author would like to thank the developers of all predecessors of Minisat, MapleLCMDistChronoBT-DL, CaDiCaL, abcdsatptwenty, P-MCOMSPS-STR-32, Relaxed_LCMDCBDL_newTech and all the authors who contributed the modifications that have been integrated. Their solvers are the foundation of our learning and improvement. This work also supported by the Fundamental Research Funds for the Central Universities (No.2682020CX59).

REFERENCES

[1] Hickey R, Feng N, Bacchus F. Cadical-trail, Cadical-alluip, Cadical-alluip-trail, and Maple-LCM-Dist-alluip-trail at SAT Competition 2020[J]. SAT COMPETITION 2020, 10.

[2] M. Moskewicz, C. Conor, Y. Zhao, L. Zhang and S. Malik, Chaff: Engineering an efficient SAT solver, in Proc. DAC’01 (2001).

[3] Ramos A., van der Tak P., Heule M.J.H. (2011) Between Restarts and Backjumps. In: Sakallah K.A., Simon L. (eds) Theory and Applications of Satisfiability Testing - SAT 2011. SAT 2011. Lecture Notes in Computer Science, vol 6695. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21581-0_18

[4] Wu, G., Chen, Q., Xu, Y., & He, X. (2018). A Hybrid Learnt Clause Evaluation Algorithm for SAT Problem. International Journal of Computational Intelligence Systems, 12(1), 250-258..

[5] Vallade, V., Le Frioux, L., Baarir, S., Sopena, J., & Kordon, F. P-MCOMSPS-STR: a Painless-based Portfolio of MapleCOMSPS with Clause Strengthening. SAT COMPETITION 2020, 56.

[6]Chen, J. Optsat, Abcdsat and Solvers Based on Simplified Data Structure and Hybrid Solving Strategies. SAT COMPETITION 2020, 25.