HKIS, HCAD, PAKIS and PAINLESS_EXMAPLELCMDISTCHRONOBT in the SC21

Rodrigue Konan Tchinda
Department of Mathematics and Computer Science
University of Dschang
Dschang, Cameroon
rodriguekonanktr@gmail.com

Clémentin Tayou Djamegni
Department of Mathematics and Computer Science
University of Dschang
Dschang, Cameroon
dtayou@gmail.com

***Abstract*—This document describes the sequential solvers HKIS, HCAD and the parallel solvers PAKIS and PAINLESS_EXMAPLELCMDISTCHRONOBT submitted to the 2021 SAT Competition.**

I. INTRODUCTION

The results of the latest SAT competition showed very good performances of the sequential solvers KISSAT [1] and CADICAL [1], [2] in the main track. The highly optimized data structures and algorithms used by these solvers make them particularly efficient. However, this does not exclude the possibility of improvements. We proposed for the 2021 edition of the SAT competition, versions of these solvers that integrate the PSIDS heuristic [3] for choosing polarities of decision variables. Moreover, we submitted a parallel version of KISSAT built on top of the Painless framework [4] as well as a slightly modified version of PAINLESS_EXMAPLELCMDISTCHRONOBT [5].

II. HKIS AND HCAD

HKIS and HCAD are both “hacks” of KISSAT and CADICAL [1], [2] respectively. They all integrate the PSIDS heuristic [3] for selecting a polarity once a branching variable chosen. PSDIS is enabled through the Boolean option `--psids`. These solvers also change the default configuration of the base solvers as follows:

- HCAD is submitted with two configurations:
  - `default` where psids=0, target=2, chrono=0 and walk=0;
  - `psids` where psids=1, target=2, chrono=1 and walk=0.
- HKIS is submitted with three configurations:
  - `default` where psids=0, target=2 and chrono=0;
  - `psids` where psids=1, target=2 and chrono=0;
  - `unsat` where psids=0, target=1, walkinitially=1 and chrono=1.

The `default` configuration of HCAD is submitted to the new CADICAL Hack subtrack of the 2021 SAT Competition.

III. PAINLESS_EXMAPLELCMDISTCHRONOBT

The parallel solver PAINLESS_EXMAPLELCMDISTCHRONOBT is identical to that we submitted to the 2020 SAT Competition [5] except for a slight change where we now load the input formula to the workers in parallel. The version we submitted to the 2021 SAT Competition was configured to launch 24 workers in parallel.

IV. PAKIS

In sequential SAT solvers, it is unlikely to find a single configuration that is the most efficient on all benchmarks of a given set. There are instances that can be easily solved by specific configurations of a given solver that are not necessarily its best configuration. Hence, running multiple configurations of a solver in parallel may help speedup solving times. The goal of PAKIS is to select a number of configurations in order to approximate the performance of the “Best Virtual Configuration” of the solver KISSAT. To achieve this, we used a test set consisting of the new instances submitted to the SAT Race 2019 and selected among a large number of possible configurations those that had the best results for SAT, UNSAT and SAT+UNSAT. Table I gives the configurations for the 24 workers used in PAKIS. The meaning of each of the options in this table can be obtained by running the solver KISSAT with the traditional `--help` option.

In contrast to many parallel SAT solvers, PAKIS does not allow any information sharing. This has some advantages regarding for instance the determinism of the solver and the production of DRAT proofs.

V. ACKNOWLEDGMENTS

We would like to thank the developers of MAPLELCMDISTCHRONOBT [6], MAPLELCMDISTCHRONOBT-DL [7], PAINLESS [4], KISSAT and CADICAL [1].

TABLE I
PaKis workers' configurations

Id	tier1	chrono	stable	walkinitially	target	phase
0	2	1	1	0	1	1
1	2	1	1	0	2	1
2	2	1	0	0	1	1
3	2	0	1	0	1	1
4	2	0	1	0	1	0
5	2	1	1	0	1	0
6	2	0	2	0	1	1
7	2	1	1	0	0	1
8	2	0	1	0	0	0
9	2	1	1	0	0	0
10	2	1	1	1	1	1
11	2	0	1	0	2	1
12	2	0	1	0	2	0
13	3	0	1	0	2	0
14	3	0	1	0	2	1
15	2	1	1	0	2	0
16	2	0	2	0	2	1
17	2	1	1	0	0	1
18	2	0	1	0	0	0
19	2	1	1	0	0	0
20	3	1	1	0	0	1
21	3	1	1	0	2	1
22	2	1	1	1	2	1
23	2	0	0	0	1	1

REFERENCES

[1] A. Biere, K. Fazekas, M. Fleury, and M. Heisinger, "CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020," in *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2020-1. University of Helsinki, 2020, pp. 51–53.

[2] A. Biere, "CaDiCaL at the SAT Race 2019," in *Proc. of SAT Race 20219 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, M. Heule, M. Järvisalo, and M. Suda, Eds., vol. B-2019-1. University of Helsinki, 2019, pp. 8–9.

[3] R. K. Tchinda and C. T. Djamegni, "Padc maplelcmdistchronobt, padc maple lcm dist and psids maplelcmdistchronobt in the sr19," *SAT RACE 2019*, p. 33.

[4] L. Le Frioux, S. Baarir, J. Sopena, and F. Kordon, "Painless: a framework for parallel sat solving," in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2017, pp. 233–250.

[5] R. Konan Tchinda and C. Tayou Djamegni, "Exmaplelcmdistchronobt, upglucose-3.0 padc and painless exmaplelcmdistchronobt in the sc20," in *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2020-1. University of Helsinki, 2020, pp. 17–18.

[6] V. Ryvchin and A. Nadel, "Maple_lcm_dist_chronobt: Featuring chronological backtracking," *SAT COMPETITION 2018*, p. 29, 2018.

[7] S. Kochemazov, O. Zaikin, V. Kondratiev, and A. Semenov, "Maplelcmdistchronobt-dl, duplicate learnts heuristic-aided solvers at the sat race 2019," *SAT RACE 2019*, p. 24.