Exact: evaluating cutting-planes learning at the PB'24 competition

Jo Devriendt
Nonfiction Software
Koksijde, Belgium
jo.devriendt@nonfictionsoftware.com

***Abstract*—EXACT is a cutting-planes learning integer programming solver built upon the foundations of CDCL-CUTTINGPLANES and ROUNDINGSAT. This document summarizes the main features of the version of EXACT submitted to the 2024 PB competition.**

I. Introduction

EXACT is a cutting-planes learning integer programming solver. It supports integer linear constraints, integer multiplicative constraints and reifications of integer linear constraints as input. EXACT eagerly translates these constraints during parsing to 0-1 linear inequalities, also known as *pseudo-Boolean* (PB) constraints. As such, the core search routines deal only with Boolean variables and linear constraints, which allows a tight conflict-driven cutting-planes learning (CDCPL) depth-first search loop based on the division method introduced in ROUNDINGSAT [1]. In fact, EXACT is a fork of ROUNDINGSAT, which participated in the last PB competition in 2016 under the name of CDCL-CUTTINGPLANES.

Complementary to their CDCPL core, EXACT and ROUNDINGSAT feature

- simplex LP solving integration [2] with SOPLEX[1] as backend solver;
- watched unit propagation [3];
- hybrid core-guided optimization [4], dynamically interleaving top-down and bottom-up optimization;
- arbitrary-sized coefficients, transparently switching to the most efficient internal representation for individual constraints;
- limited support for an early version of the VERIPB [5] proof format, to log certificates of unsatisfiability;

EXACT improves upon its predecessor through a myriad of refactorings and extensions. Chief amongst these are

- various changes to the cutting-plane conflict analysis routine with the goal to learn stronger constraints from a conflict;
- in-processing using probing, dominance breaking, binary implication analysis and cardinality detection;
- support for integer variables and non-linear constraints, i.a., parsing .lp and .mip formats;
- a fully stateful Python interface with support for assumptions, unsat core generation, objective modification, solution counting and solution intersecting (*full propagation*);
- a branch[2] with full support for VERIPB's proof format version 2.

II. Submissions

Four versions of EXACT were submitted to the following tracks:

- *exact*: DEC-LIN, DEC-NLC, OPT-LIN, OPT-NLC, PARTIAL-LIN, SOFT-LIN
- *exact_no_soplex*: DEC-LIN, DEC-NLC, OPT-LIN, OPT-NLC, PARTIAL-LIN, SOFT-LIN
- *exact_veripb2*: DEC-LIN-CERT, OPT-LIN-CERT
- *exact_veripb2_no_soplex*: DEC-LIN-CERT, OPT-LIN-CERT

The regular *exact* and *exact_no_soplex* submissions stem from a single commit,[3] and the VeriPB submissions *exact_veripb2* and *exact_veripb2_no_soplex* stem from another.[4]

To simplify the internal proof generation routines, the VeriPB submissions disable dominance breaking inprocessing and replace core-guided optimization by a simpler lower bound assumption routine. As such, we expect them to perform worse than the regular versions.

exact_no_soplex and *exact_veripb2_no_soplex* disable the use of the SOPLEX LP solver, which we expect again to reduce performance. However, in internal tests, we noticed that disabling SOPLEX may be beneficial on certain instances that were otherwise unsolvable. This PB competition is an opportunity to evaluate how significant the performance loss is, and whether EXACT, without LP solving support, can solve instances that no other approach can solve.

References

[1] J. Elffers and J. Nordström, "Divide and conquer: Towards faster pseudo-Boolean solving," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*, Jul. 2018, pp. 1291–1299.

[2] J. Devriendt, A. Gleixner, and J. Nordström, "Learn to relax: Integrating 0-1 integer linear programming with pseudo-Boolean conflict-driven search," in *Proceedings of the 17th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR '20)*, ser. Lecture Notes in Computer Science, vol. 12296. Springer, Sep. 2020, pp. xxiv–xxv.

1. https://github.com/scipopt/soplex
2. https://gitlab.com/nonfiction-software/exact/-/tree/veripb_2?ref_type=heads
3. https://gitlab.com/nonfiction-software/exact/-/commit/43b6ba712c68eab97bd8805e04eddb4447c58459
4. https://gitlab.com/nonfiction-software/exact/-/commit/5c7ed0d5f0df8c2892cc85a26fdbc6c2e6e6f786

[3] J. Devriendt, “Watched propagation of 0-1 integer linear constraints,” in *Proceedings of the 26th International Conference on Principles and Practice of Constraint Programming (CP ’20)*, ser. Lecture Notes in Computer Science, vol. 12333. Springer, Sep. 2020, pp. 160–176.

[4] J. Devriendt, S. Gocht, E. Demirović, J. Nordström, and P. Stuckey, “Cutting to the core of pseudo-Boolean optimization: Combining core-guided search with cutting planes reasoning,” vol. 35, no. 5, May 2021, pp. 3750–3758. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16492

[5] J. Elffers, S. Gocht, C. McCreesh, and J. Nordström, “Justifying all differences using pseudo-Boolean reasoning,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI ’20)*, Feb. 2020, pp. 1486–1494.