

The CP-SAT-LP Solver

Laurent Perron ✉

Google, Paris, France

Frédéric Didier ✉

Google, Paris, France

Steven Gay ✉

Google, Paris, France

Abstract

The CP-SAT-LP solver is developed by the Operations Research team at Google and is part of the OR-Tools [8] open-source optimization suite. It is an implementation of a purely integral Constraint Programming solver on top of a SAT solver using Lazy Clause Generation [11]. It draws its inspiration from the chuffed solver [4], and from the CP 2013 plenary by Peter Stuckey on Lazy Clause Generation [12].

The CP-SAT-LP solver improves upon the chuffed solver [4] in two main directions. First, it uses a simplex alongside the SAT engine. Second, it implements and relies upon a portfolio of diverse workers for its search part.

The use of the simplex brings the obvious advantages of a linear relaxation on the linear part of the full model. It also started the integration of MIP technology into CP-SAT-LP. This is a huge endeavour, as MIP solvers are mature and complex. It includes presolve – which was already a part of CP-SAT –, dual reductions, specific branching rules, cuts, reduced cost fixing, and more advanced techniques. It also allows to integrate tightly the research from the Scheduling on MIP community [3, 1, 9] along with the most advanced scheduling algorithms [13]. This has enabled breakthroughs in solving and proving hard scheduling instances of the Job-Shop problems [5] and Resource Constraint Project Scheduling Problems [6, 2].

Using a portfolio of different workers makes it easier to try new ideas and to incorporate orthogonal techniques with little complication, except controlling the explosion of potential workers. These workers can be categorized along multiple criteria like finding primal solutions – either using complete solvers, Local Search [7] or Large Neighborhood Search [10] –, improving dual bounds, trying to reduce the problem with the help of continuous probing. This diversity of behaviors has increased the robustness of the solver, while the continuous sharing of information between workers has produced massive speedups when running multiple workers in parallel.

All in all, CP-SAT-LP is a state-of-the-art solver, with unsurpassed performance in the Constraint Programming community, breakthrough results on Scheduling benchmarks (with the closure of many open problems), and competitive results with the best MIP solvers (on purely integral problems).

2012 ACM Subject Classification Applied computing → Operations research

Keywords and phrases Constraint Programming, Operations Research, Sat Solver

Digital Object Identifier 10.4230/LIPIcs.CP.2023.3

Category Invited Talk

References

- 1 David Applegate and William Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2):149–156, 1991. doi:10.1287/ijoc.3.2.149.
- 2 Christian Artigues, Sophie Demasse, and Emmanuel Neron. *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. ISTE/Wiley, 2008. URL: <https://hal.science/hal-00482946>.
- 3 Egon Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic Discrete Methods*, 6(3):466–486, 1985.



© Laurent Perron, Frédéric Didier, and Steven Gay;
licensed under Creative Commons License CC-BY 4.0

29th International Conference on Principles and Practice of Constraint Programming (CP 2023).

Editor: Roland H. C. Yap; Article No. 3; pp. 3:1–3:2

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- 4 Geoffrey Chu, Peter J. Stuckey, Andreas Schutt, Thorsten Ehlers, Graeme Gange, and Kathryn Francis. the chuffed solver, June 2023. URL: <https://github.com/chuffed/chuffed>.
- 5 Junwen Ding, Zhipeng Lü, Chu-Min Li, Liji Shen, Liping Xu, and Fred Glover. A two-individual based evolutionary algorithm for the flexible job shop scheduling problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2262–2271, 2019.
- 6 Rainer Kolisch and Arno Sprecher. Psplib - a project scheduling problem library: Or software - orsep operations research software exchange program. *European Journal of Operational Research*, 96(1):205–216, 1997. doi:10.1016/S0377-2217(96)00170-1.
- 7 Bjørnar Luteberget and Giorgio Sartor. Feasibility jump: an lp-free lagrangian mip heuristic. *Mathematical Programming Computation*, 15, March 2023. doi:10.1007/s12532-023-00234-8.
- 8 Laurent Perron and Vincent Furnon. Or-tools, March 2023. URL: <https://developers.google.com/optimization/>.
- 9 Maurice Queyranne. Structure of a simple scheduling polyhedron. *Math. Program.*, 58:263–285, 1993. doi:10.1007/BF01581271.
- 10 Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In Michael J. Maher and Jean-Francois Puget, editors, *Principles and Practice of Constraint Programming - CP98, 4th International Conference, Pisa, Italy, October 26-30, 1998, Proceedings*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer, 1998. doi:10.1007/3-540-49481-2_30.
- 11 Peter J. Stuckey. Lazy clause generation: Combining the power of sat and cp (and mip?) solving. In Andrea Lodi, Michela Milano, and Paolo Toth, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 5–9, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- 12 Peter J. Stuckey. Those who cannot remember the past are condemned to repeat it. In Christian Schulte, editor, *Principles and Practice of Constraint Programming*, pages 5–6, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- 13 Petr Vilim. Timetable edge finding filtering algorithm for discrete cumulative resources. In Tobias Achterberg and J. Christopher Beck, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 230–245, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.