Mistral 2.0

XCSP3 Competition 2017

Emmanuel Hebrard[1] and Mohamed Siala[2]

[1] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
hebrard@laas.fr
[2] Insight Centre for Data Analytics
Department of Computer Science, University College Cork, Ireland
mohamed.siala@insight-centre.org

Mistral is an open source constraint programming library written in C++ and available on GitHub (https://github.com/ehebrard/Mistral-2.0). It implements a modelling API, however, it can also read instance files in XCSP3 or FlatZinc format. Moreover, it is fully interfaced with Numberjack [6] which provides a Python API for modelling and solving combinatorial optimization problems using several back-end solvers.

Solver Engine

The solver engine relies on standard mechanisms, using a priority constraint queue and a domain event stack to implement the propagation procedure. Moreover, it supports dynamic type change for variables: domains are initially implemented using intervals or Boolean types whenever possible, and can be changed to (bit)sets during search when a propagator requires it. The backtracking mechanism is implemented using a trail in a standard way.

Propagators

Several classic global constraints are implemented, such as LexOrdering [4], bound consistency propagator for AllDifferent [9] and Gcc [10]. Moreover, less standard constraints were implemented within the context of research articles on constraint propagation, such as the AtMostSeqCard constraint for car-sequencing [12] or a VertexCover constraint [3] to reason about cliques, independant set or vertex covers.

Search Strategy

The search heuristic used for the XCSP3 competition is based on *Last Conflit* [8], using a variant of *Weighted Degree* [2] as default strategy: in the case of failure raised by a propagator of a global constraint, an *explanation* of the conflict is computed and only the weight of the variables participating in the conflict is incremented. This heuristic is fully described in [7]. Moreover, given the next variable x to branch on, the solver chooses the value that was assigned to x in the best solution found so far, if possible, or the minimum value in the domain of x otherwise.

Applications of Mistral

Mistral was used to implement a state-of-the-art method for disjunctive scheduling which improved several best known results on classic benchmarks [5]. More recently, some clause learning methods were implemented in Mistral, still improving the results on disjunctive scheduling [1] and car-sequencing problems [11]. These methods were not used within the context of the XCSP3 competition.

References

1. Christian Artigues, Emmanuel Hebrard, Valentin Mayer-Eichberger, Mohamed Siala, and Toby Walsh. SAT and hybrid models of the car sequencing problem. In *Proceedings of the 11th International Conference on Integration of AI and OR Techniques in Constraint Programming – CPAIOR*, pages 268–283, 2014.
2. Frédéric Boussemart, Fred Hemery, Christophe Lecoutre, and Lakhdar Sais. Boosting systematic search by weighting constraints. In *Proceedings of the 16th European Conference on Artificial Intelligence – ECAI*, pages 146–150, 2004.
3. Clément Carbonnel and Emmanuel Hebrard. Propagation via kernelization: The vertex cover constraint. In *Proceedings of the 22nd International Conference on Principles and Practice of Constraint Programming – CP*, pages 147–156, 2016.
4. Alan M. Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, and Toby Walsh. Propagation algorithms for lexicographic ordering constraints. *Artif. Intell.*, 170(10):803–834, 2006.
5. Diarmuid Grimes and Emmanuel Hebrard. Solving variants of the job shop scheduling problem through conflict-directed search. *INFORMS Journal on Computing*, 27(2):268–284, 2015.
6. Emmanuel Hebrard, Eoin O'Mahony, and Barry O'Sullivan. Constraint programming and combinatorial optimisation in numberjack. In *Proceedings of the 7th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems – CPAIOR*, pages 181–185, 2010.
7. Emmanuel Hebrard and Mohamed Siala. Explanation-based weighted degree. In *Proceedings of the 14th International Conference on Integration of AI and OR Techniques in Constraint Programming – CPAIOR*, pages 167–175, 2017.
8. Christophe Lecoutre, Lakhdar Sas, Sbastien Tabary, and Vincent Vidal. Reasoning from last conflict(s) in constraint programming. *Artificial Intelligence*, 173(18):1592 – 1614, 2009.
9. Alejandro López-Ortiz, Claude-Guy Quimper, John Tromp, and Peter van Beek. A fast and simple algorithm for bounds consistency of the alldifferent constraint. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence – IJCAI*, pages 245–250, 2003.

10. Claude-Guy Quimper, Peter van Beek, Alejandro López-Ortiz, Alexander Golynski, and Sayyed Bashir Sadjad. An efficient bounds consistency algorithm for the global cardinality constraint. In *Proceedings of the 9th International Conference Principles and Practice of Constraint Programming – CP*, pages 600–614, 2003.
11. Mohamed Siala, Christian Artigues, and Emmanuel Hebrard. Two clause learning approaches for disjunctive scheduling. In *Proceedings of the 21st International Conference on Principles and Practice of Constraint Programming – CP*, pages 393–402, 2015.
12. Mohamed Siala, Emmanuel Hebrard, and Marie-José Huguet. An optimal arc consistency algorithm for a particular case of sequence constraint. *Constraints*, 19(1):30–56, 2014.