Modeling and Solving Combinatorial Constrained Problems with PyCSP^3 and ACE

Christophe Lecoutre

CRIL, University of Artois & CNRS, France

lecoutre@cril.fr

Mots-clés : *Modeling Library, Constraint Solver*

1 Introduction

Solving optimization problems remains a difficult task, especially when instances are large and/or when optimality proofs are wished. Actually, the difficulty is two-fold : modeling in an appropriate way the problems, and solving their instances efficiently. Main paradigms for optimization, namely, mathematical programming, meta-heuristics and constraint pogramming (including SAT), are somewhat complementary ; each paradigm having its own successful stories.

In this paper, we succintly present the main ingredients of the complete tool chain we propose for handling combinatorial constrained problems, from a CP (Constraint Programming) perspective :

— PyCSP^3 : a Python library for modeling constrained problems

— ACE : a constraint solver written in Java

Note that an intermediate format, XCSP^3, is used to represent problem instances while preserving structure of models.

2 PyCSP^3

PyCSP^3 [8] is a Python library for writting, in a declarative manner, models of constraint satisfaction and optimization problems. Models are compiled into the intermediate format XCSP^3 [3, 4], and can be directly solved with two embedded solvers ACE and Choco.

More specifically, you can build models for :

— CSP (Constraint Satisfaction Problem)

— COP (Constraint Optimization Problem)

Interestingly, on pycsp.org you can find more than 60 Jupyter notebooks for gently learning CP (Constraint Programming) :

— understanding 25 popular constraints

— understanding, step by step, 34 models of classical problems

Code. PyCSP^3 code is available

— on Github : https://github.com/xcsp3team/pycsp3

— as a PyPi package : https://pypi.org/project/pycsp3

3 ACE

ACE (AbsCon Essence) is an open-source constraint solver, developed in Java. ACE [7] focuses on :

— integer variables, including 0/1 (Boolean) variables,
— state-of-the-art table constraints, including ordinary, starred, and hybrid table constraints,
— popular global constraints (AllDifferent, Count, Element, Cardinality, Cumulative, etc.),
— search heuristics, as e.g., dom/wdeg [2, 10], last-conflict [6], BIVS [5], solution-saving [9],
— mono-criterion optimization

ACE is derived from the constraint solver AbsCon that has been used as a research platform in our team at CRIL during many years. Many ideas and algorithms have been discarded from AbsCon, so as to get a constraint solver of reasonable size and understanding.

ACE is a competitive solver as shown by the results of the 2022 XCSP^3 competition [1] (note that ACE is not officially ranked because it would rise some conflicts of interest). We believe that its competitiveness is due to its search components (and their combinations), showing that "search is not dead".

Code. ACE code is available
— on Github : https://github.com/xcsp3team/ace

Références

[1] Gilles Audemard, Christophe Lecoutre, and Emmanuel Lonca. Proceedings of the 2022 XCSP3 competition. *CoRR*, abs/2209.00917, 2022.

[2] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais. Boosting systematic search by weighting constraints. In *Proceedings of ECAI'04*, pages 146–150, 2004.

[3] Frédéric Boussemart, Christophe Lecoutre, Gilles Audemard, and Cédric Piette. XCSP3-core : A format for representing constraint satisfaction/optimization problems. *CoRR*, abs/2009.00514, 2020.

[4] Frédéric Boussemart, Christophe Lecoutre, and Cédric Piette. XCSP3 : an integrated format for benchmarking combinatorial constrained problems. *CoRR*, abs/1611.03398, 2016.

[5] J.-G. Fages and C. Prud'homme. Making the first solution good ! In *Proceedings of ICTAI'17*, pages 1073–1077, 2017.

[6] C. Lecoutre, L. Sais, S. Tabary, and V. Vidal. Reasonning from last conflict(s) in constraint programming. *Artificial Intelligence*, 173(18) :1592–1614, 2009.

[7] Christophe Lecoutre. ACE, a generic constraint solver. *To Appear*, 2022.

[8] Christophe Lecoutre and Nicolas Szczepanski. PyCSP3 : modeling combinatorial constrained problems in Python. *CoRR*, abs/2009.00326, 2020.

[9] J. Vion and S. Piechowiak. Une simple heuristique pour rapprocher DFS et LNS pour les COP. In *Proceedings of JFPC'17*, pages 39–45, 2017.

[10] H. Wattez, C. Lecoutre, A. Paparrizou, and S. Tabary. Refining constraint weighting. In *Proceedings of ICTAI'19*, pages 71–77, 2019.