OscaR.cbls : an open source framework for constraint-based local search

Renaud De Landtsheer
CETIC, rdl@cetic.be

Christophe Ponsard
CETIC, cp@cetic.be

Oscar.cbls is a framework for constraint based local search. Constraint-based local search is a technique for solving combinatorial problems including both constraint and optimization problems. Oscar.cbls is based on the reference book [1]. Oscar.cbls is part of the OscaR project also featuring constraint and linear programming. Oscar is freely available under the LGPL V2.1 Open Source license [2, 3]. Oscar.cbls is around 17k LOC big, mostly written in Scala.

1 Introduction

Constraint-based local search (CBLS) is an efficient implementation of local search. *Local search* solves combinatorial problems by maintaining a current solution, and repeatedly moving to a neighbour solution that improves the objective function. The neighbour is selected from a set of solutions obtained by slightly modifying the current solution. This process can be enriched with various meta-heuristics such as taboo, metropolis, simulated annealing, random restart, etc.

2 Core principle of Oscar.cbls

Optimization problems are represented by a model and an objective function. In CBLS, they are made of *incremental variables*, and so-called invariants. OscaR.cbl supports two types of variables, namely integers and set of integers. *Invariants* are code fragments implementing atomic expressions, and maintaining one or more output variables according to the atomic expressions they implement. A typical example of invariant is the `Sum` invariant. It maintains an output integer variable equal to the sum of a specified collection of integer variables. When one or more variable of this collection is updated, the `Sum` invariant updates the output variable accordingly, by efficiently performing some delta-based computation. The update of `Sum` is performed in O("number of updated variables"). Oscar.cbls offers a library of roughly eighty invariants including set operators (union, diff, cardinality, etc) arithmetic invariants (Sum, Prod, Abs, Max, ArgMax, etc.) logical invariants (array access -also called "element"-, clusters , etc).

OscaR.cbls also offers a framework for representing constraint problems. This framework defines the concept of *constraint system* as a container in which constraints are *posted*. Oscar.cbls supports around ten types constraints including basic ($==$, $\leq$, etc.), and global constraints (AllDiff, etc.). Constraints are lagrangian relaxations : every constraints maintains a violation degree that is

zero if the constraint is enforced, and gradually increase to reflect the amount of correction required on the variables to get the constraint enforced.

To ensure fast neighbourhood exploration, OscaR.cbls supports partial propagation. When exploring a neighbourhood, we generally only need to know the value of an objective functions at various neighbour to select the proper neighbour to move to. Models generally involve other aspects such as some variables defining which moves are authorized, or maintaining some auxiliary variables based on violation degree of related to meta-heuristics. These only need to be updated once the proper neighbour is chosen, to start a new neighbourhood exploration. Partial propagation ensures that these auxiliary parts of the model are not updated until necessary.

OscaR.cbls requires one to develop a proper search procedure including neighbourhood exploration, and meta-heuristics. OscaR.cbls is meant to provide as much support as possible to implement these features easily and efficiently.

3 Similar systems and main differences

Comet is the seminal system for Constraint-based local search [1]. It features a differentiation facility that is not implemented in OscaR.cbls. OscaR.cbls relies on partial propagation to provide comparable efficiency. Besides, differentiation as provided by Comet cannot handle intricate models where constraints are posted on variables controlled by invariants. It was available under commercial licence.

Kangaroo provides a partial propagation feature that is more selective than OscaR.cbls [4]. To our knowledge, it is not available.

LocalSolver is a commercial solver implementing CBLS. It supports Boolean and float variables, and does not require the user to specify neighbourhood or meta-heuristics [5].

4 Future

OscaR.cbls is constantly been improved and is already reaching similar performance and scalability level than commercial products. Domain specific packages (routing, scheduling) have been recently added. Future work includes industrial validation and hybridation with other OscaR components.

Références

[1] Laurent Michel Pascal Van Hentenryck. *Constraint-based Local Search*. MIT Press, 2009.

[2] Oscar website. https ://bitbucket.org/oscarlib/oscar.

[3] the lgpl2.1 licence. http ://www.gnu.org/licenses/lgpl-2.1-standalone.html.

[4] M. A. Hakim Newton, Duc Nghia Pham, Abdul Sattar, and Michael Maher. Kangaroo : an efficient constraint-based local search system using lazy propagation. In *Proceedings of CP'11*, pages 645–659, 2011.

[5] F. Gardi R. Megel K. Nouioua T. Benoist, B. Estellon. Localsolver 1.x : a black-box local-search solver for 0-1 programming. *4OR*, 9(3) :299 – 316, 2011.