

# OscAR.cbls 6.0: Review after 12 years of Continuous Development

Thomas Fayolle, Quentin Meurisse, Renaud De Landtsheer,  
Fabian Germeau, Stefano Micheli

CETIC, Department of Combinatorial Algorithmics

e-mail: {thomas.fayolle, quentin.meurisse, renaud.delandtsheer,  
fabian.germeau, stefano.micheli}@cetic.be

## 1 Introduction

OscAR.cbls [1] is a library, written in Scala, dedicated to the implementation of Constraint-Based Local Search [2] engines. This library contains a collection of generic constraints and neighborhoods and provides tools to implement new ones.

In 2013, Oscar.cbls 1.0 was presented in Orbel [3]. The library has just released its version 6.0 and has been subject to a complete refactoring. After more than 12 years of continuous development, this is an opportunity to review some of its main components and some of the major changes performed.

## 2 Variable and Constraints

OscAR.cbls lets you define three types of variables: integers, sets of integers and sequences of integers. The integer sequence is an ordered sequence of integers and is often used to represent routes in vehicle routing problems.

These variables can be used to define constraints: elements that take variables as input and update the value of other variables as output. These output variables can then be used as inputs to other constraints, which allows us to build a problem-modeling graph. When a variable is modified, it notifies all the constraints that use it as input, informing them of its modification. This information enables the constraint's output values to be updated incrementally.

Between each notification, sequences retain changes in the form of symbolic delta [4]. When they send a notification, they inform their constraints which nodes have been inserted, removed or moved since the last notification.

When the decision variables are modified during the search, these modifications are transmitted along the problem modeling graph, using a propagation mechanism. The propagation affects each node in the graph only once.

Thanks to Scala's object paradigm, a constraint abstraction can manage the integration to the propagation structure. So, implementing a new constraint can be achieved by focusing only on the incremental updates of the outputs.

### 3 Neighborhoods and Combinators

Oscar.cbls also contains several neighborhoods that are standard in the local search literature. On an array of integer variables, it contains, for example, the *Assign* or the *Swap*. It also contains standard neighborhoods for vehicle routing, such as *OnePointMove*, *2-opt* or *3-opt*.

In addition to the standard neighborhoods, Oscar.cbls also contains neighborhood combinators that allow you to create new neighborhoods from existing ones. For example, you can create a Cartesian product of neighborhoods. Let  $A$  and  $B$  be two neighborhoods, and let  $S$  be the set of solutions. For any solution  $s$  and any neighborhood  $V$ , let  $neigh(V, s)$  be the set of solutions reachable from solution  $s$  using neighborhood  $V$ . The product of  $A$  and  $B$  ( $A \times B$ ) explores the solutions defined by the set  $\cup_{a \in neigh(A, s)} neigh(B, a)$ , i.e., the set of solutions that can be reached using neighborhood  $B$  from all solutions obtained using neighborhood  $A$  from solution  $s$ .

### 4 Conclusion and Future Works

The Oscar.cbls library contains a collection of constraints and neighborhoods. A specific language is also used to define an optimization model and a search procedure to find a solution to this problem using local search. Its modularity makes it possible to build solvers for a wide range of problems (*Warehouse location problem*, *Vehicle Routing Problem*, etc.).

The library's modularity also means that other heuristics (*VLSN* [5], etc.) or meta-heuristics (simulated annealing, etc.) can be added. To speed up performance, it is also planned to include primitives that facilitate the use of distributed computing in the search procedure.

**Acknowledgment:** This work is supported by FEDER/FTJ Wallonie 2021-2027 - project n° 933 - VirtualLab\_Cetic.

### References

- [1] Oscar website. <https://github.com/cetic/oscar-cbls>
- [2] Laurent Michel, and Pascal Van Hentenryck. Constraint-based Local Search. *MIT Press*, 2009
- [3] Renaud De Landtsheer, and Christophe Ponsard. Oscar. cbls: an open source framework for constraint-based local search. *Proceedings of ORBEL*, 2013
- [4] Renaud De Landtsheer, Yoann Guyot, Gustavo Ospina, Fabian Germeau, and Christophe Ponsard. Reasoning on sequences in constraint-based local search frameworks. *CPAIOR Proceedings 15*, 2018
- [5] Sébastien Moushuy, Pascal Van Hentenryck, and Yves Deville. Constraint-based Very Large-Scale Neighborhood search. *Constraints : an international journal*, 17(2):87-122, 2012