A Global Constraint for Modeling Time in Scheduling Problems for Local Search

Stefano Michelini
CETIC asbl, Combinatorial Algorithmics
e-mail: stefano.michelini@cetic.be

Renaud De Landtsheer
CETIC asbl, Combinatorial Algorithmics
e-mail: renaud.delandtsheer@cetic.be

The Iterative Flattening and Relaxing algorithm (IFlatIRelax) [1] is a heuristic aimed at solving the cumulative scheduling problem, which is a scheduling problem where each machine has a certain capacity, and each task may require more than one capacity unit. The heuristic is an improvement of the Iterative Flattening algorithm by [3], and it works by manipulating the directed acyclic graph (DAG)-based model of the problem, where nodes are associated with tasks and arcs with task precedence relations. More specifically, the algorithm explores the solution space by repeatedly adding feasible arcs between tasks in a minimal conflict set associated with a capacity constraint violation in order to repair feasibility (the *flattening* step), or removing arcs belonging to the *critical path*, i.e., the longest path in the DAG determining the value of the makespan, the objective function of the problem (the *relaxing* step).

In their introduction to constraint-based local search (CBLS) [2], the authors present different ways to tackle scheduling problems within their own COMET framework, including IFlatIRelax. However, in order to do so, the search procedure evaluates the makespan through an ad-hoc formula, breaking a design principle according to which the local search neighbourhoods can only evaluate moves by querying the objective function. Therefore, the search procedure and the model cannot be adapted to solve variants of this problem. An relevant variant is the *flexible job-shop problem* where one must also select the appropriate resource for each operation.

In this work in progress, we address this issue by developing a global constraint whose input is the structure of the DAG, as manipulated by the search procedure and whose output is the makespan. The makespan is maintained incrementally by using the algorithms in [4], and the ad-hoc formula used in [2]to achieve $O(1)$-time complexity whenever possible. This further allows us to formulate the IFlatIRelax as a *Destroy&Repair* search procedure using standard neighbourhood combinators from the OSCAR.cbls framework and generic *flatten* and *relax* neighborhoods. The implementation is included in open source OSCAR.cbls framework [5].

Acknowledgement: This research is supported by the SLS project (Logistics in Wallonia - nr 8457)

References

[1] Michel, L. and Van Hentenryck, P., *Iterative Relaxations for Iterative Flattening in Cumulative Scheduling*, ICAPS. Vol. 4, 2004.

[2] Van Hentenryck, P., and Michel, L., *Constraint-based Local Search*, MIT Press, 2005.

[3] Cesta, A., Oddi, A., and Smith, S. F., *Iterative flattening: A scalable method for solving multi-capacity scheduling problems*, AAAI/IAAI, 742–747, 2000.

[4] Katriel, I., Michel, L. and Van Hentenryck, P., *Maintaining longest paths incrementally*, Constraints 10.2, 159-183, 2005.

[5] OscaR Team. OscaR: Operational research in Scala, 2012. Available under the LGPL licence from `https://bitbucket.org/oscarlib/oscar`