Simulated Annealing for Discrete Lot-Sizing and Scheduling

Sara Ceschia*
DPIA, University of Udine, sara.ceschia@uniud.it

Luca Di Gaspero
DPIA, University of Udine, luca.digaspero@uniud.it

Andrea Schaerf
DPIA, University of Udine, andrea.schaerf@uniud.it

Keywords: Lot-sizing, Simulated Annealing

Lot-sizing and scheduling (LSS) is a production planning problem that is very relevant in production economics. Several versions of the problem have been proposed in the literature, depending on the specific application domain. In this work, we consider the discrete, single machine, single level, multi-item version of the general LSS problem, denoted by the acronym MI-DLS-CC-SC. This version of the problem is also called Pigment Sequencing (PS) by Pochet and Wolsey [1] (§14.4), and it has been recently included in the problem library CSPlib [2] (prob058).

For the PS problem, we propose a metaheuristic approach based on Simulated Annealing, along with a statistically-principled tuning procedure. Due to the scarcity of public instances and the unavailability of really challenging ones, we also developed a parametrized instance generator. Using our generator, we created instances with different features, in terms of items, periods, and production level.

In order to assess the quality of our solver, we compare it with the state-of-the-art search methods, which are, to the best of our knowledge, the three MILP models designed and implemented in Mosel by Pochet and Wolsey [1]. To this aim, starting from their original source code, we reimplemented the three models in CPLEX (v. 12.5), and ran them on the same environment of our solver.

The outcome is that CPLEX, using the most effective of the three models of Pochet and Wolsey (model 3), in 1 hour is able to find the optimal solution on about 44.4% of the instances, a nearly optimal one on other 37.7% (i.e., a gap inferior to 3% respect to the lower bound), and it is far from the lower bound in the remaining ones. Our solver finds a nearly optimal solution for all instances in a reasonable time (about 75s). More specifically, our average solutions are at most 1.33% worse than the best/optimal solution, and at most 2.89% worse than the lower bound. On the CSPlib benchmarks, both model 3 in CPLEX and our method always reach the optimal solution with average computational times of 0.33 and 4.5 seconds, respectively.

All instances and best solutions are available for verification and future comparisons at the website opthub.uniud.it, along with the online solution validator.

References

[1] Y. Pochet, L. A. Wolsey, *Production planning by mixed integer programming*. Springer Science & Business Media. 2006

[2] I. P., Gent, T. Walsh, *CSPlib: a benchmark library for constraints* (`www.csplib.org`). Proceedings of the 5th International Conference on Principles and Practices of Constraint Programming (CP-99), pp. 480–481, Springer-Verlag, Lecture Notes in Computer Science 1713, 1999.