An Optimal Constraint Programming Approach to the Open-Shop Problem

Arnaud Malapert
École des Mines de Nantes, LINA
École Polytechnique de Montréal, CIRRELT

Hadrien Cambazard
Cork Constraint Computation Centre
Department of Computer Science
University College Cork, Ireland

Christelle Guéret
École des Mines de Nantes
IRCCyN UMR CNRS 6597
Nantes, France

Narendra Jussien
École des Mines de Nantes
LINA UMR CNRS 6241
Nantes, France

André Langevin and **Louis-Martin Rousseau**
École Polytechnique de Montréal
CIRRELT
Montréal, Québec, Canada

Abstract

This is a summary of the journal article (Malapert et al. 2012) published by *Journal on Computing* entitled *"An Optimal Constraint Programming Approach to the Open-Shop Problem"*. The article presents an optimal constraint programming approach for the Open-Shop scheduling problem, which integrates recent constraint propagation and branching techniques with new upper bound heuristics. Randomized restart policies combined with nogood recording allow to search diversification and learning from restarts. This approach is compared with the best-known metaheuristics and exact algorithms, and shows better results on a wide range of benchmark instances.

Open-Shop problems are at the core of many scheduling problems involving unary resources such as Job-Shop or Flow-Shop problems, which have received an important amount of attention because of their wide range of applications. Among the many techniques proposed in the literature, Constraint Programming (CP) is among the most successful. In shop problems, n jobs, consisting each of m tasks, must be processed on m machines. A machine can process only one task at a time. The processing orders of tasks which belong to a job can vary: global order (flow-shop); order per job (job-shop); no order (open-shop). In Open-Shop problems, the tasks of a job can be processed in any order, but only one at a time. The processing times are known in advance and constant. We consider the construction of non-preemptive schedules of minimal makespan C_{max} which is NP-Hard for $m \geq 3$.

The study and classification of models and search algorithms show that one of the major challenge of solving optimally these problems is to provide good solutions as quick as the metaheuristics. The most famous exact method (Brucker et al. 1997) consists of fixing precedences on the critical path of heuristic solutions computed at each node. It has been improved since by using intelligent backtracking (Guéret, Jussien, and Prins 2000) and consistency techniques (Dorndorf, Pesch, and Huy 2001). More recently, (Laborie 2005) applied a method for cumulative scheduling on Open-Shop problems and (Tamura et al. 2006) applied a method that encodes Constraint Satisfaction/Optimization problems with integer linear constraints into a Boolean Satisfiability Testing problem. Many metaheuristics algorithms have been developed in the last decade to solve the Open-Shop problem. The most recent and successful metaheuristics are: Genetic Algorithm (Prins 2000), Construction and Repair (Chatzikokolakis, Boukeas, and Stamatopoulos 2004), Ant Colony Optimization (Blum 2005) and Particle Swarm Optimization (Sha and Hsu 2008).

Constraint programming techniques have been widely used to solve scheduling problems. A *Constraint Satisfaction Problem* (CSP) consists of a set V of variables defined by a corresponding set of possible values (the domains D) and a set C of constraints. A solution of the problem is an assignment of a value to each variable such that all constraints are simultaneously satisfied. Constraints are handled through a propagation mechanism which allows the reduction of the domains of variables and the pruning of the search tree. The propagation mechanism coupled with a backtracking scheme allows the search space to be explored in a complete way. Scheduling is probably one of the most successful areas for CP thanks to specialized global constraints, which allow modelling resource limitations and temporal constraints.
Constraint programming models in scheduling usually represent a non-preemptive task by a triplet of non-negative integer variables representing the start, processing time and end of the task.
We now present our constraint programming model to tackle Open-Shop problems. First, we state *disjunctive* global constraints which model the fact that a single machine or job

---

can be processed at any given time. Then, *precedence* constraints are tackled in the decision and propagation process through a dedicated global constraint. We also state additional dedicated constraints such as *forbidden intervals* and *symmetry breaking* constraints.

We introduce a new algorithm characterized by its: *simplicity*, most of its components are available in constraint solvers; *flexibility*, two constraint models offer different trade-offs among propagation strength, speed and simplicity; *genericity*, it is applicable to any disjunctive scheduling problem. The algorithm relies on recent constraint filtering algorithm and (randomized) branching techniques with new upper bound heuristics. The algorithm starts with a randomized constructive heuristic (without propagation) which initializes the upper bound so that the selection and propagation of initial choices are improved.

During the search, the filtering is based on the disjunctive graph model and disjunctive global constraints, whereas the branching is conducted by adding precedences to the disjunctive graph with the profile strategy. Besides, randomized restart policies combined with nogood recording allow search diversification and learning from restarts. Indeed, nogoods are recorded at each restart to prevent exploring the same part of the search space again. Restart policies are based on the following observation: the longer a backtracking search algorithm runs without finding a solution, the more likely it is that the algorithm is exploring a barren part of the search space. These choices are motivated by the fact that propagation techniques are very effective once a tight upper bound is known, but only slow down search otherwise, and initial choices made by the branching are both the least informed and the most important, as they lead to the largest subtrees and the search can hardly recover from early mistakes.

We have shown that restarting alone can greatly improve the solution of Open-Shop problems but lacks robustness. Restarting basically helps to find good upper bounds quickly, but once those are known, longer runs are eventually needed to prove optimality. The balance between restarting quickly to improve the upper bound and searching more to prove optimality is difficult to achieve. Enhancing the restarting policy with nogood recording compensates for this drawback and improves the resolution significantly.

In a complementary research (Grimes, Hebrard, and Malapert 2009), we introduce another model combining simple filtering methods with a new variable selection strategy inspired by the weighted degree strategy, and empirically show that the complex inference methods and search strategies can, surprisingly, often be advantageously replaced by this naive model. The weighted degree strategy includes a learning component from the failures caused by the constraints during the search. The naive model is frequently faster, but less robust and only adapted to our weighted degree strategy.

The proposed algorithm outperforms other exact algorithms and metaheuristics published so far on a wide range of benchmarks for the Open-Shop problem.

These contributions have been implemented within the Choco constraint solver, a java library for constraint programming, which can be used for teaching, research and real-life applications. They are now publicly available to the academic and industrial communities (choco.mines-nantes.fr).

References

Blum, C. 2005. Beam-ACO: hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Computers & Operations Research* 32(6):1565–1591.

Brucker, P.; Hurink, J.; Jurisch, B.; and Wöstmann, B. 1997. A branch & bound algorithm for the open-shop problem. In *GO-II Meeting: Proceedings of the second international colloquium on Graphs and optimization*, 43–59. Amsterdam, The Netherlands: Elsevier Science Publishers B. V.

Chatzikokolakis, K.; Boukeas, G.; and Stamatopoulos, P. 2004. Construction and Repair: A Hybrid Approach to Search in CSPs. *SETN* 3025:2004.

Dorndorf, U.; Pesch, E.; and Huy, T. P. 2001. Solving the open shop scheduling problem. *Journal of Scheduling* 4:157–174.

Grimes, D.; Hebrard, E.; and Malapert, A. 2009. Closing the Open Shop: Contradicting Conventional Wisdom. In Gent, I. P., ed., *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming (CP 2009)*, volume 5732 of *Lecture Notes in Computer Science*, 400–408. Springer.

Guéret, C.; Jussien, N.; and Prins, C. 2000. Using intelligent backtracking to improve branch-and-bound methods: An application to Open-Shop problems. *European Journal of Operational Research* 127:344–354.

Laborie, P. 2005. Complete MCS-Based Search: Application to Resource Constrained Project Scheduling. In Kaelbling, L. P., and Saffiotti, A., eds., *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, 181–186. Professional Book Center.

Malapert, A.; Cambazard, H.; Guéret, C.; Jussien, N.; Langevin, A.; and Rousseau, L.-M. 2012. An Optimal Constraint Programming Approach to the Open-Shop Problem. *INFORMS Journal on Computing* 24(2):228–244.

Prins, C. 2000. Competitive genetic algorithms for the open-shop scheduling problem. *Mathematics Methods of Operations Research* 52(3):389–411.

Sha, D. Y., and Hsu, C.-Y. 2008. A new particle swarm optimization for the open shop scheduling problem. *Computers & Operations Research* 35(10):3243–3261.

Tamura, N.; Taga, A.; Kitagawa, S.; and Banbara, M. 2006. Compiling Finite Linear CSP into SAT. In Benhamou, F., ed., *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP 2006)*, volume 4204 of *Lecture Notes in Computer Science*, 590–603. Springer.