Scheduling of activities in heavy maintenance centers of SNCF

Rahman Torba[1,2], Stéphane Dauzère-Pérès[1], Claude Yugma[1],
Cédric Gallais[2], Juliette Pouzet[2]

[1] Mines Saint-Etienne, CNRS, UMR 6158 LIMOS, Gardanne, France
dauzere-peres, yugma, r.torba@emse.fr
[2] SNCF, Saint-Denis, France
cedric.gallais, juliette.pouzet, rahman.torba@sncf.fr

Mots-clés : *scheduling, multiple projects, multiple skills, heavy maintenance, MILP, meta-heuristics*

1 Introduction

In this work, a real-life multi-skill resource-constrained multi-project scheduling problem is modeled and solved. The problem corresponds to an industrial problem in ten different dedicated factories of SNCF, the french national railway company, where the heavy maintenance of the rolling stock is carried out. Two objective functions are considered independently : (i) Minimization of the sum of weighted tardiness of the projects (SWTP) and (ii) Minimization of the sum of weighted duration of the projects (SWDP).

The well-known Resource-Constrained Project Scheduling Problem (RCPSP) belongs to the class of hard optimization problems and, usually, instances of more than 60 activities cannot be solved with exact methods in reasonable computational times ([3]). Furthermore, most companies produce several products that usually share the same resources. The survey of [4] shows that 84% of companies are working on multiple projects. The multi-project version of the RCPSP is closer to real world applications. Yet, most papers are solving the single project version of the RCPSP. The objective function is often the minimization of the makespan, i.e the completion time of the last activity. The multi-project extension results in more complex problems due to the scarcity of shared resources, the interactions and the competition between different projects, specific project characteristics, deadlines and more elaborate objective functions.

To schedule heavy maintenance operations on rolling stock, a *time-indexed* MILP (Mixed-Integer Linear Programming) model with several industrial constraints, such as precedence constraints with lag times, *time-dependent* resource capacity constraints and machine assignment, is proposed. In the model, each rolling stock unit is considered as a project, maintenance operations as activities requiring a certain number of human resources with different skills and one specific machine to be performed. Since resources have multiple skills and several rolling stock units are maintained simultaneously, the problem studied in this paper corresponds to the multi-skill resource-constrained multi-project scheduling problem (MSRCMPSP) (see e.g. [5] and [1]).

To solve large industrial instances of several hundred projects and several thousand operations, two greedy algorithms based on priority rules (PRs) have been initially implemented. The priority rules are computed using a global precedence graph defined from the precedence graph of each project (a graph with critical paths for each project). The solutions provided by the greedy algorithms are improved via a memetic algorithm.

2 Solution approaches

Two constructive heuristics for the RCPSP problem were adapted and implemented to solve the MSRCMPSP problem : (i) Serial Scheduling Generation Scheme (SSGS) and (ii) Parallel Scheduling Generation Scheme (PSGS) ([2]). The main differences compared to the classical SSGS and PSGS are the assignment of multi-skill resources and the computation of priority rules, since the priorities depend on the tightness of each project. Seven priority rules are implemented and tested on 27 real industrial instances. Table (1) summarizes the numerical results of SSGS. The LS priority rule which selects the activity with the minimal Latest Start time outperforms the other priority rules. The average gap (Avg Gap(%)) of the other priority rules is quite significant, in particular when SWTP is considered as objective function.

	SWDP			SWTP		
PR	#Best	Avg Gap(%)	Max Gap(%)	#Best	Avg Gap(%)	Max Gap(%)
EF	0	8.9	13.0	0	54.4	83.3
ES	1	4.0	7.9	0	40.0	78.7
LF	4	3.9	9.1	5	22.2	60.2
LS	**22**	**0.4**	**4.8**	**22**	**0.9**	**8.3**
Rand	0	20.3	31.4	0	78.8	94.3
SA	0	36.7	58.9	0	88.2	95.2
SST	0	17.2	26.2	0	71.9	91.4

TAB. 1 – SSGS : Performance of priority rules and gaps from the best solution

To improve the solutions of the greedy algorithms, a memetic algorithm (MA) has been proposed, which combines a genetic algorithm (GA) and a simulated annealing algorithm (SA) used for local search. The detailed framework of the MA and the numerical results will be presented and discussed in the conference. In particular, the MA is compared to the GA, the SA and the greedy algorithms.

3 Conclusions and perspectives

The MA has been tested and validated on 27 large industrial instances from three different heavy maintenance centers. The MA outperforms the MILP model, the GA, the SA and the greedy algorithms in terms of solution quality and convergence.

Perspectives on considering the stochastic version of the problem with uncertain processing times and uncertain additional activities will be discussed in the conference.

Références

[1] Odile Bellenguez and Emmanuel Néron. Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. In *International conference on the practice and theory of automated timetabling*, pages 229–243. Springer, 2004.

[2] Rainer Kolisch. Serial and parallel resource-constrained project scheduling methods revisited : Theory and computation. *European Journal of Operational Research*, 90(2) :320–333, 1996.

[3] Oumar Koné, Christian Artigues, Pierre Lopez, and Marcel Mongeau. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1) :3–13, 2011.

[4] Antonio Lova, Concepción Maroto, and Pilar Tormos. A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European journal of operational research*, 127(2) :408–424, 2000.

[5] A Alan B Pritsker, Lawrence J Waiters, and Philip M Wolfe. Multiproject scheduling with limited resources : A zero-one programming approach. *Management science*, 16(1) :93–108, 1969.