Structural and Experimental Comparisons of Formulations for a Multi-Skill Project Scheduling Problem with Partial Preemption

Christian Artigues, Pierre Lopez, Oliver Polo Mejía

Structural and Experimental Comparisons of Formulations for a Multi-Skill Project Scheduling Problem with Partial Preemption

Christian Artigues, Pierre Lopez, and Oliver Polo Mejía

LAAS-CNRS, Université de Toulouse, CNRS, France
{christian.artigues, pierre.lopez, oliver.polo-mejia}@laas.fr

Keywords: RCPSP, Multi-skill, Partial preemption, Mixed-integer programming, Constraint programming.

1 Problem statement

Preemptive scheduling problems assume that all resources are released during preemption periods, and that they can be used to perform other activities. However, in certain cases, constraints require that a subset of resources remains allocated to the activity when it has been interrupted, to ensure safety for example. Suppose one must execute an experimental activity that requires an inert atmosphere for its execution. In practice, one can stop this activity and allow the technicians and some of the equipment to be used in other activities. However, safety and operational constraints force us to preserve the inert atmosphere even when the activity is stopped (before its end). In other words, one cannot release the equipment that ensures the inert atmosphere during the preemption periods. Traditional preemptive schedule models cannot represent this behaviour since they assume that all resources are released during the preemption periods. Until now, the only way to model this activity, while respecting safety requirements, was to declare it as "non-preemptive". However, this decision can increase the project makespan, especially when the activities have restrictive time windows and the availability/capacity of the resources vary over time. We call *partial preemption* the possibility of only releasing a subset of resources during the preemption periods.

We are concerned here in multi-skill project scheduling problem (MSPSP) (Bellenguez and Néron 2012). We present in this extended abstract a new variant of the MSPSP that uses the concept of partial preemption. The variant of the problem under study is then called Multi-Skill Project Scheduling Problem with Partial Preemption (MSPSP-PP). To the best of our knowledge, it has not been studied yet in the scientific literature.

In the MSPSP-PP, if an activity is interrupted, we release only a subset of resources while seizing the remainder. We can then classify the set $\overline{I}$ of activities to be scheduled into three types according to the possibility of releasing the resources during the preemption periods: 1) Non-preemptive activities ($\overline{NP}$), if none of the resources can be released; 2) Partially preemptive activities ($\overline{PP}$), if a subset of resources can be released; and 3) Preemptive activities ($\overline{P}$), if all resources can be set free. In our case, the partial preemption is only related to mono-skilled resources, and we made the hypothesis that resources can always be released during preemption periods.

Our objective in the MSPSP-PP is to find a feasible schedule that minimises the total duration of the project ($C_{\max}$). Finding a solution consists in determining the periods during which each activity is executed and also which resources will execute the activity in every period; all this, while respecting the resources capacity and the activities characteristics. We must schedule these activities on renewable resources with limited capacity; they can be cumulative mono-skilled resources (machines or equipment) or disjunctive multi-skilled resources (technicians) mastering a given number of skills. Multi-skilled resources can respond to more than one skill requirement per activity and may execute it partially (except for non-preemptive activities where technicians must perform the whole activity). An activity is defined by its duration (D_i), its precedence relationships (set $\overline{E}$), its requirements of resources, its requirements of skills, the minimum number of technicians needed to perform it, and the subset of preemptive resources. Activities might or not have either a deadline or a release date. Figure 1 illustrates an example of an MSPSP-PP instance and a possible solution.

	Duration	Skill needed	Machine	Dead line	Release date	Activity type
Act 1	4	(S1,1)	(M1,1)	-	-	Preemptive
Act 2	2	(S3,1),(S4,1)	(M1,1)	5	3	Non-preemptive
Act 3	4	(S2,1)	(M1,1)	-	-	Partially preemp (M1 can not be released)

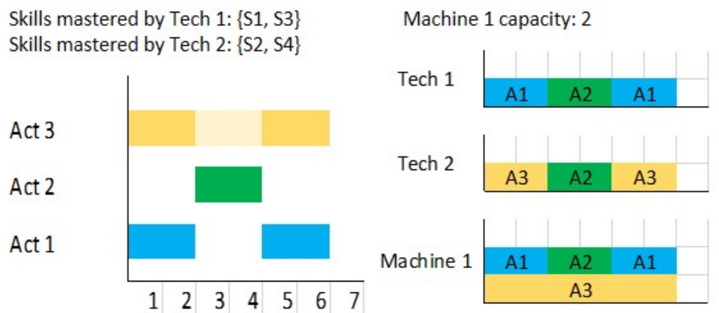

Fig. 1. Example of an MSPSP-PP instance.

The complexity of the MSPSP with partial preemption can be established using the classical RCPSP (Resource-Constrained Project Scheduling Problem) as a starting point. For each instance of the RCPSP, we can match an instance of the MSPSP with partial preemption, where all resources are mono-skilled, and none of the resources can be preempted. Thus, we can define the RCPSP as a particular case of the MSPSP with partial preemption. Since the RCPSP has been proved to be strongly NP-hard (Błazewicz *et al.* 1983), we can, therefore, infer that the MSPSP with partial preemption is also strongly NP-hard.

We propose five formulations for the MSPSP-PP using Mixed-Integer/Linear Programming (MILP) and Constraint Programming (CP).

2 MILP formulations

We present below five time-indexed formulations of the problem over a discretized horizon $\overline{H}$. These formulations generalize the ones presented in (Polo *et al.* 2018) and (Polo *et al.* 2019). All models are based on on/off binary variables $Y_{i,t}$ stating whether an activity i is in process in time period t, on/off binary variables $O_{j,i,t} = 1$ if technician j is assigned to activity i during period t, binary variable $S_{j,i} = 1$ if technician j is assigned to non-preemptive activity i (this variable is used to express that any technician assigned to a non-preemptive activity must remain assigned until its completeness). For any partially preemptive activity i, an on/off binary variable $Pp_{i,t} = 1$ if activity i is preempted in time period t. For the three first models, step binary variable $Z_{i,t} = 1$ if partially preemptive or non-preemptive activity i starts in time period t or before and step binary variable $W_{i,t} = 1$

if partially preemptive or non-preemptive activity i ends in time period t or after. We only provide a subset of the constraints of the first model (MSPP1a): precedence constraints (1) and the constraints (2–8) that link variables Y, Pp, Z, W, S, O and $C_{\max}$, the project makespan. The other constraints are standard resource constraints and operator availability constraints.

$$\begin{aligned}
&D_i * (1 - Y_{l,t}) \geq \sum_{t'=t}^{|\overline{H}|} Y_{i,t'} && \forall (i,l) \in \overline{E}, \forall t \in \overline{H} && (1)\\
&Z_{i,t} \geq Y_{i,t'} && \forall i \notin \overline{P}, \forall t \in \overline{H}, \forall t' \leq t && (2)\\
&W_{i,t} \geq Y_{i,t'} && \forall i \notin \overline{P}, \forall t \in \overline{H}, \forall t' \geq t && (3)\\
&Pp_{i,t} = Z_{i,t} + W_{i,t} - Y_{i,t} - 1 && \forall i \in \overline{PP}, \forall t \in \overline{H} && (4)\\
&Z_{i,t} + W_{i,t} - Y_{i,t} = 1 && \forall i \in \overline{NP}, \forall t \in \overline{H} && (5)\\
&O_{j,i,t} \geq S_{j,i} + Y_{i,t} - 1 && \forall i \in \overline{NP}, \forall j \in \overline{J}, \forall t \in \overline{H} && (6)\\
&O_{j,i,t} \leq S_{j,i} && \forall i \in \overline{NP}, \forall j \in \overline{J}, \forall t \in \overline{H} && (7)\\
&C_{\max} \geq t * Y_{i,t} && \forall i \in \overline{I}, \forall t \in \overline{H} && (8)
\end{aligned}$$

Given the variables $W_{i,t}$ and $Z_{i,t}$, we can replace the precedence constraints (1) by a disaggregated version below, yielding the second model (MSPP1b), while the third model (MSPP1c) includes both constraints (1) and (9).

$$Z_{l,t} + W_{i,t} \leq 1 \qquad \forall (i,l) \in \overline{E}, \forall t \in \overline{H} \tag{9}$$

We also propose two mixed continuous-time/discrete-time models (MSPP2a and MSPP2b), replacing binary variables $W_{i,t}$ and $Z_{i,t}$ by continuous time variables G_i and F_i representing the start and completion times of activity i, respectively. We replace constraints (1–5) by:

$$\begin{aligned}
&F_i + 1 \leq G_l && \forall (i,l) \in \overline{E} && (10)\\
&Pp_{i,t} \leq 1 - Y_{i,t} && \forall i \in \overline{PP}, \forall t \in \overline{H} && (11)\\
&F_i - G_i + 1 \leq D_i + \sum_{t \in H} Pp_{i,t} && \forall i \in \overline{PP} && (12)\\
&F_i - G_i + 1 \leq D_i && \forall i \in \overline{NP} && (13)\\
&F_i \geq t * Y_{i,t} && \forall i \in \overline{I}, \forall t \in \overline{H} && (14)\\
&G_i \leq t * Y_{i,t} + (1 - Y_{i,t}) * |\overline{H}| && \forall i \in \overline{I} && (15)
\end{aligned}$$

It remains to express the fact that partial preemption variables $Pp_{i,t}$ must be equal to 0 outside the execution interval of i. We either use the following constraints using variables Y (MSPP2a):

$$Pp_{i,t} \leq \sum_{t'=1}^{t} Y_{i,t'} \;; \qquad Pp_{i,t} \leq \sum_{t'=t}^{|\overline{H}|} Y_{i,t'} \qquad \forall i \in \overline{PP}, \forall t \in \overline{H} \tag{16}$$

or the following ones using variables F and G (MSPP2b):

$$F_i \geq t * Pp_{i,t} \;; \qquad G_i \leq t * Pp_{i,t} - (1 - Pp_{i,t}) * |\overline{H}| \qquad \forall i \in \overline{PP}, \forall t \in \overline{H} \tag{17}$$

In Section 3, we compare the proposed MILP formulations in terms of LP relaxation strength and we provide a computational comparison with the constraint programming formulation described in (Polo *et al.* 2019).

3 Structural and computational comparisons

3.1 Structural comparison of the MILP formulations

Using the transformation $G_i = |\overline{H}| - \sum_{t \in \overline{H}} Z_{i,t} + 1$ and $F_i = \sum_{t \in \overline{H}} W_{i,t}$ we show that the constraints in model MSPP2 involving the F_i and G_i variables are all implied by the constraints of model MSPP1 augmented with the transformation. As the computational experiments show that there are instances where MSPP1 has a strictly better LP relaxation than MSPP2, this yields the following result:

Theorem 1. *Formulation MSPP1b is tighter than formulations MSPP2a and MSPP2b.*

We could not prove a dominance relation between MSPP1a and MSPP1b, which was corroborated by the experiments fox which some bounds provided by the MSPP1a relaxation are better than those provided by the MSPP1b relaxation and vice-versa, which justifies the proposal of MSPP1c.

3.2 Computational experiments on the MILP and CP formulations

For computational tests, we use CPLEX 12.7 and CP Optimizer 12.7 for solving the MILP models and the CP models, respectively (using the default configuration and limiting the number of threads used by the solvers at 8). The computation time was limited to 10 minutes. We use the four sets of 30 activities instances of (Polo *et al.* 2019), each of them having 50 instances and a different proportion of preemption types. The activity durations are between 5 to 10 time units. There are up to 15 skills, 8 cumulative resources, 8 technicians (multi-skilled resources) divided into two teams, 20% of activities with time windows, the density of precedence relationships is low, and an average optimum $C_{\max}$ between 70 and 90 time units.

First of all, the MILP formulations are only efficient when the number of preemptive activities is high. For these instances, model MSPP1b provides the larger number of optimal solutions but model MSPP2b is faster and has a better average gap. There was no perceptible advantage for the integrated model MSPP1c. These computational results confirm one more time that a theoretically stronger formulation does not necessarily imply better practical performance. The MILP MSPP2b model outperforms the CP one when the percentage of preemptive activities is high, proving the optimality of a higher number of instances, and giving a lower average gap. CP, on the other hand, gives better results when this percentage is low. One could then say that the two methods are complementary.

Future research should be done in order to develop a hybrid method that better exploit the characteristics of each instance.

References

Bellenguez-Morineau O., E. Néron, 2008, “Multi-mode and multi-skill project scheduling problem”, In *Resource-constrained project scheduling: models, algorithms, extensions and applications*, C. Artigues, S. Demassey, E. Néron (Eds.), Wiley-ISTE, pages 149-160.

Błazewicz J., J. K. Lenstra and A. H. G. Rinnooy Kan, 1983, “Scheduling subject to resource constraints: classification and complexity”, *Discrete applied mathematics*, Vol. 5, pp. 11-24.

Polo-Mejía O., M.-C. Anselmet, C. Artigues and P. Lopez, 2018, “Mixed-integer and constraint programming formulations for a multi-skill project scheduling problem with partial preemption”, In *12th International Conference on Modeling, Optimization and Simulation (MOSIM 2018)*, Toulouse, France, pages 367-374.

Polo-Mejía O., C. Artigues, P. Lopez and V. Basini, 2019, “Mixed-integer/linear and constraint programming approaches for activity scheduling in a nuclear research facility”, *International Journal of Production Research*, In Press. DOI:10.1080/00207543.2019.1693654.