Multi-project scheduling problems with shared multi-skill resource constraints

Meya HAROUNE[1,2], Cheikh DHIB[2], Emmanuel NERON[1], Ameur SOUKHAL[1],Hafedh MOHAMED BABOU[2] and Mohamedade NANNE[2]

[1] Laboratoire d'Informatique Fondamentale et Appliquée de Tours LIFAT EA 6300 ROOT ERL-CNRS 7002, France(2)
meya.haroune@etu.univ-tours.fr,{emmanuel.neron, ameur.soukhal}@univ-tours.fr
[2] Unité de recherche Documents Numériques et Interaction de l'Université de NouakchottAl-Asriya DNI, Mauritanie
dhib.cheikh@iscae.mr, hafedh.mohamed-babou@esp.mr, farouk@una.mr

Keywords: Operational research, Project scheduling, Personnel constraints, Skills management, ILP, Heuristics, Tabu search.

1 Introduction

Companies are increasingly concerned by the organization of their projects, taking into account skills of their teams, in order to answer to customer requests. Thus, in this context, several project managers can compete for renewable resources (human or machines) to carry out their projects and to avoid delay that may be expensive (Dhib *et. al.* (2016)).

In this paper, we study a problem of multi-project scheduling where several project managers manage one or more projects. Projects share the same resources. Shared resources are human teams available with limited capacities and various skills. Our aim is to propose to the project managers a schedule of their activities within a fixed time horizon. This schedule must take into account the completion time of each phase of the project to reduce the weighted cost of realization of the projects (payment of the penalties). Our goal is to study a relevant industrial model, by integrating the particularities and the characteristics of persons and activities, as realistic as possible.

Project scheduling problems have been widely studied and the dedicated literature on this subject is very important. The most well known are RCPSP (for Resource Constrained Project Scheduling Problem) Artigues *et. al.* (2008), the MM-RCPSP (for Muli-modeProjects Scheduling Problem) Bianco *et. al.* (1998), MSPSP (for Multi-skill Project Scheduling Problem)Bellenguez-Morineau (2006). Some other works focused on scheduling in a multi-project context. Concerning the studied problem, it has been introduced by Néron *et. al.* (2011), that corresponds to a real case of multi-project planning. This problem can be encountered in IT companies.

In the next section, the studied problem and the used notations are formally presented.

2 Problem description

Multiple projects are running simultaneously and share common human resources. Each project $K_{l,l=1...,L}$ consists of a set of independent and preemptive activities. An activity J_i is characterized by an estimated load p_i expressed in day. There are release dates r_i, due dates d_i and late penalties w_i expressed in week.

Each person M_j, has a quota per project $Q_{j,l}$, determining his/her maximum rate of participation to the project K_l. The periods of availability per week of each person, are known and are divided between the different projects in proportion to the rate of his/her intervention. Each resource has a skill level for performing an activity. Thus, the processing time (nominal load) of an activity is computed according to the efficiency of the person in charge of its processing. It is defined for activity J_i and person M_j as follows: $p_{i,j} = (2 - v_{i,j})p_i$, where $v_{i,j}$ is the given efficiency level of resource M_j in carrying out the activity J_i, $0 \leq v_{i,j} \leq 1$. Durations and availabilities are measured by a half of a day as time unit.

An activity is assigned to a single person throughout its realization, even if it is spread over several weeks. At a time t, a person M_j can work only on one activity. The number of activities on which a person may be assigned during the same week can not exceed the given value b_j. So, this limitation reduces the number of context changes and thus increases the efficiency of the person. However, a minimum (resp. maximum) load C_i^{min} (resp. C_i^{max}) is defined for each activity J_i to frame the quantities of its realization per week. One solution to this problem is to check whether there is a possible schedule within the fixed time horizon of persons to different project activities. The schedule of different activities within the week is not considered here. We are interested in determining the time spent per week and per person on each activity while respecting the constraints mentioned above and minimizing the total weighted tardiness, $Z = \sum w_i T_i$, where T_i is the number of weeks of delay of activity J_i.

The originality of this model is the consideration of the notion of skill and the minimum/maximum activities loads. This problem is $\mathcal{NP}$-hard since the activity scheduling problem on uniform parallel machines noted $Qm \mid r_i, pmtn \mid \sum w_i T_i$ known as $\mathcal{NP}$-hard, is a special case of the studied problem (Brucker *et. al.* (1997)).

Example: Let's consider two projects (K_1, K_2) to be planned over a three-week horizon. Two persons (M_1, M_2) work on these projects. The data are given in Table 1. In this table we particularly specify the project to which the activities belongs, resources that are able to perform the activity and the efficiencies of the resources to perform an activity. Table 2 presents the characteristics of the resources, represented by their weekly availability and their quotas per project.

Table 1. Characteristics of activities

		J_1	J_2	J_3	J_4
Interval $\{r_i, d_i\}$		{1,3}	{2,3}	{2,2}	{1,2}
Load (p_i)		8	4	7	6
C_i^{max}		6	6	5	8
Penalty(w_i)		10	20	15	10
Project		K_1	K_1	K_2	K_2
Skill		$\{M_1,M_2\}$	$\{M_2\}$	$\{M_1\}$	$\{M_1,M_2\}$
Efficiencie $v_{i,j}$	M_1	1.0	-	0.5	0.0
	M_2	0.5	1.0	-	1.0

Table 2. Characteristics of resources

	M_1	M_2	
W_1	8	6	Availability
W_2	10	10	
W_3	5	8	
K_1	50%	70%	Quotas
K_2	60%	60%	

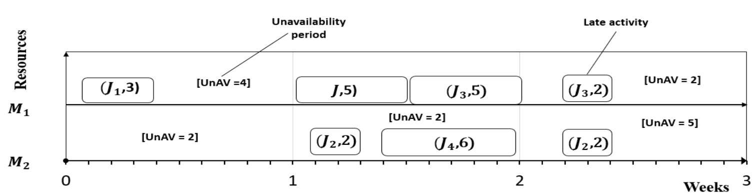

Fig. 1. Example of feasible solution

Figure 1 presents a feasible solution. Only the activity J_3 is late, i.e. a one-week delay ($T_3 = 1$). Hence, the total cost due to this delay is 15.

3 Heuristics approaches

To optimally solve the studied problem, an integer linear programming is proposed by Meya *et. al.* (2019). This ILP solves small and medium sized instances up to 40 activities over 8 weeks in reasonable time. To solve large size instances two heuristics are developed: a local search algorithm and a tabu search algorithm.

3.1 2-phases heuristics

The 2-phase algorithm operates in 2 phases: With the first phase, we seek to build an initial solution, i.e. a possible assignment of operators to the different activities of each project. For this purpose, we use a combination of a priority rule (to determine which activity should be considered first) and algorithm developed to solve bin packing problem (to determine which person should execute the current activity). Different priority rules have been tested: weighted earliest due dates WEDD; shortest (resp. longest) processing time SPT (LPT) where the efficiency coefficients to perform activity is taken into account. To assign activities to persons, two known algorithms are considered: First-fit algorithm (the first operator found with availability large enough to process the activity is chosen); Best-fit algorithm (the operator with the smallest large enough availability on project of the activity is chosen). The preliminary results show that the combination of WEDD and Best-fit algorithm outperforms all other heuristics. In the second phase, with a given assignment, we should determine a compatible schedule of each person over time. In this second phase, we should solve $\mathcal{NP}$-hard problem. So, heuristic based on a maximum flow with minimum cost (MF-MC) is proposed. The graph is defined by three levels of nodes plus two dummy nodes. Note that this graph is built by person (see figure 2 where 3 activities belong to 2 projects are assigned to one operator).

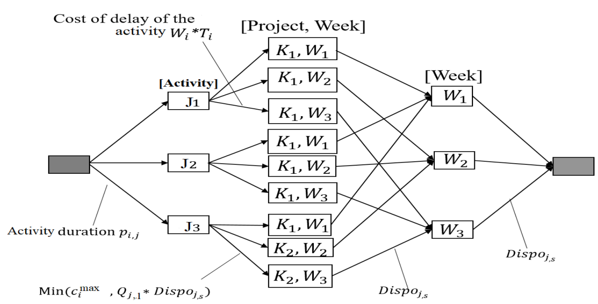

Fig. 2. Max Flow-Min Cost: 3 activities assigned to one person

In order to minimize the total weighted tardiness, we add costs on the edges between 'Activity' node and node 'project, week'. This cost represents the cost of delay of the activity multiplied by the index of the week.

3.2 Local search algorithm

Two neighborhood functions have been developed to improve the assignment of activities. The first function $V1$ tries to reassign late activities to other person, where the second function $V2$ tries to free up time for a person performing a late activity (an activity on-time can be reassigned to another person). To determine a compatible schedule of each person over time, we recall MF-MC algorithm.

3.3 Tabu search algorithm

The basic idea of our developed tabu search is described as follow. Initial assignment of activities to operators is given by the best 2-phases heuristic. Two neighbourhood operators are used: the first one is $V1$. According to the second neighbourhood operator, denoted $V3$, late or incomplete activity is switched with another activity. Swapping is only possible if the execution time of these two activities overlaps. The best solution is kept as a starting point for the next iteration. At each iteration, the number of neighbors generated depends on the quality of the solution selected to be the current solution. If each time the current solution is improved, the neighborhood size is expanded. Otherwise, if after a certain number of iterations, the found best solution is not improved, a diversification strategy is applied. This strategy consists in replacing a current solution with another one generated by applying one of the priority rules previously introduced. We then obtain a new assignment of activities to operators. The tabu search is therefore restarted with this new assignment.

3.4 Experimental results

The generated instances are inspired by a practical case: 2 and 3 projects with 60 and 90 activities per project and between 10 and 15 persons. These projects run simultaneously over a horizon of 12 and 14 weeks. The experiments were performed using Cplex 12.8.0 solver where the calculation time is limited to 3600sec.

Tables 4 and 5 summarize the obtained results where the first column represents the percentage of instances for which heuristics found a feasible solution (reminder that each project should be performed during a given period); The second column gives the percentage of the instances that are optimally solved by heuristics; The third column shows the average deviation of instances not solved to optimal and the last column gives the execution time in seconds. All these values correspond to 34 instances for which ILP found feasible solutions within 3600sec.

Table 3. Local search approach (LS)

Method	% Feas	% Opti	% GAP	Total GAP	Time (m)
InitAssignment	36	22,4	58,4	45	0,06
LS with V1	69	46,2	29	15,3	1,2
LS with V2	64,4	47	23,3	14,5	1,8
LS with V1 & V2	86,3	64,5	22,5	7	3,1

Table 4. Tabu search algorithm (TS)

Method	% Feas	% Opti	% GAP	Total GAP	Time (m)
InitAssignment	36	22,4	58,4	45	0,06
TS with V1	82	53	22,1	13,4	1,5
TS with V3	64,5	59,5	20,1	11,7	3,6
TS with V1 & V3	92,8	73	15	**4,6**	5,2

We can see in both tables that the neighborhood operator V1 is more efficient compared to the other neighborhood operators V2 and V3. We note that TS with V1 and V3 outperforms all other heuristics. Over the 34 instances, the mean deviation is at most 4,6% (see total Gap column where 73% of instances are optimally solved). Over the 27% of the remaining instances, the mean deviation from the optimal solution given by TS with V1 and V3 is at most 15%.

References

Brucker P., B. Jurisch and A. Krämer, 1997, "Complexity of scheduling problems with multipurpose machines", *Annals of Operations Research*, Vol. 70, pp. 57-73.

Dhib Ch., A. Soukhal and E. Néron, 2016, "Solving Methods for a Preemptive Project Staffing and Scheduling problem", *Handbook on Project Management and Scheduling*, Vol. 1&2, Springer.

Artigues C., S. Demmasey and E. Néron, 2008, "Resource-constrained project scheduling problem: Models, Algorithms, Extension and Applications", *Control systems, robotics and manufacturing series*, Wiley.

Bianco, P. Dell'Olmo and M.G. Speranza, 1998, "Heuristics for multi-mode scheduling problems with dedicated resources", *European Journal of Operational Research*, Vol. 107(2), 260-271.

Bellenguez-Morineau 0., 2006, "Méthodes de résolution pour un problème de gestion de projet avec prise en compte de compétences", Thèse de doctorat de l'université de Tours.

Néron E., A. Soukhal, R. Semur, F. Tercinet and S. Bouamer, 2011, "planification de projets -modèle et méthode de résolution", ROADEF(2011).

Haroune M., Dhib Ch., Néron E., A. Soukhal, H. Babou, and F. Nanne, 2019, "Ordonnancement multi-projets à contraintes de ressourcespartagées multi-compétences", ROADEF(2019).