A new grey-box approach to solve challenging workforce planning and activities scheduling problems

Stefano Lucidi[1,2] and Ludovica Maccarrone[1,2]

[1] Department of Computer, Control, and Management Engineering,
University of Rome "La Sapienza ", Italy
lucidi, maccarrone@dis.uniroma1.it
[2] ACTOR s.r.l., via Nizza 45, 00198 Rome, Italy
stefano.lucidi, ludovica.maccarrone@act-OperationsResearch.com

Keywords: Workforce management, Project scheduling, Grey-box optimization, MILP.

1 Introduction and motivations

Nowadays, a key success factor for many large enterprises is the ability of properly managing labor cost and timetables. This is the reason why workforce planning and scheduling tools are now getting more and more developed.

Two are the typical issues arising in such applications: the first is related to the medium and long-term goal of estimating the amount of workers that the company will require in future periods. The second, mostly linked to short-term operations, involves the assignment of human resources to activities in order to meet deadlines and industrial plans.

In practice, to conduct a complete analysis and evaluate the effectiveness of a solution it is important to take into account both time and financial objectives, considering not only the need of reducing durations and delays but also the ability to do so within reasonable budgets. The result is a trade-off problem looking at the same time at avoiding resource underutilization and incapacity to comply with due dates.

In the following, we present a new approach to solve the workforce scheduling problem in complex applicative contexts such as manufacturing and logistics, characterized by the simultaneous processing of several activities, the occupation of wide areas, the coexistence of independent workloads, the use of advanced machineries and, above all, the employment of different types of operators, having various abilities and experience levels.

Standard approaches usually address this issue by defining distinct planning, scheduling and allocation problems. However, within the considered context, the problem of providing the right number of workers with the right skills at the right time is inherently linked to the schedule of the activities. For this reason, we rather propose a strategy to tackle all these aspects at the same time, taking into account a reasonable time horizon. As a result we obtain a large problem requiring not only a proper representation of processes complexity, but also a feasible assignment of operators to tasks and an optimized activities scheduling.

In what follows, the structure of the problem is formalized and a specialized simulation-based decomposition framework is proposed.

2 Problem description

Hereinafter, we will consider systems where one or more processes are executed. Roughly, a process can be described in terms of two basic definitions: the skills employed and the component activities. A skill represents the ability of an operator to perform certain tasks, thus identifying a worker type. An activity can be any non-interruptible elementary time-consuming operation requiring skilled operators to be completed.

Activities may be linked by some precedence constraints, but have variable starting times that can be modified in order to create optimal schedules satisfying logical and s-trategical restrictions. Indeed, activities may be subject to release and deadline constraints, and are affected by workforce availability limitations.

A basic assumption of our approach is that the number of skills required by each activity is not fixed and therefore there exist many feasible combinations of operators guaranteeing tasks completion. In particular, allowing to vary the workforce assignments between a lower and an upper limit, we evidently admit variability to operations processing times. Such aspect heavily characterizes our procedure. Assuming it is not possible to derive analytic functions expressing the link between allocated skills and time to complete the activities, we have based our solution method on the use of a set of ad hoc simulators, having as input a vector of worker availabilities and as output a duration estimate.

The result is a simulation-optimization problem facing a typical trade-off between different objectives. On the one hand it aims at reducing the employment cost, minimizing the number of necessary skilled operators, on the other, it encourages an optimal activities scheduling, trying to parallelize the tasks and decrease the overall completion time.

2.1 Mathematical formulation

In order to introduce the general mathematical formulation we have developed for this problem, we first need to list some basic definitions. We will consider m activities and n different skills. Let $A = \{1, ..., m\}$ be the set of indexes for activities, and $S = \{1, ..., n\}$ be the set of indexes for skills. Each activity is non-preemptable and is characterized by a variable processing time, a release date r_j and a due date d_j. Both r_j and d_j are real parameters and can be set to zero and infinity to nullify the associated constraints.

Precedence relations are given by the set Q of ordered index pairs, such that $(j_1, j_2) \in Q$ means that the execution of activity j_2 must start after the end of activity j_1. The same concept can be expressed by an *activity-on-node* graph whose nodes correspond to activities, and arcs represent sequence constraints. From this perspective, a necessary condition to guarantee consistent precedence relations is that the graph contains no cycles.

Our problem formulation involves three main types of decision variables. First, the total number of operators made available for each skill is represented by a vector $y \in \mathbb{N}^n$, such that y_i denotes the availability of resource i. Second, integer variables x_{ij} are required to indicate the number of workers with skill i assigned to activity j. Finally, the starting-time continuous variables t_j are introduced for each activity j, thus making possible the scheduling.

Alongside these definitions, two auxiliary variables γ_{jc} and θ_{jc} are used in our mathematical model, where j and c both belong to A; their meaning will be soon clear.

Ultimately, minding our assumption on the dependence among operators assignments and time to complete the activities, we can identify the output of the j-th simulator with the symbol $\tau_j = \phi_j(x_{1j}, \ldots, x_{ij}, \ldots, x_{nj})$, so expressing the processing time of activity j as an unknown function of the variable skill allocations.

We can therefore formalize the problem in a bilevel programming formulation having as upper-level and lower-level objectives two generic functions. Their global effect can be thought of as the combination of two conflicting components: the first accounting for the workforce cost, the second expressing a time objective. As an example of this trade-off, we can consider a situation where variables y_i are, at the same time, pushed down to lower salaries expenses, and pushed up to relax resource constraints and obtain better results in activities scheduling, improving, for example, the overall *makespan*, the sum of projects completion times or the average finish time of activities.

We propose the following formulation:

$$
\begin{aligned}
\min_{x,y,\tau,t,\gamma,\theta} \quad & f_1(x,y,\tau,t,\gamma,\theta) && && (1)\\
s.t. \quad & l_{ij} \le x_{ij} \le u_{ij} && i \in S,\;\; j \in A && (2)\\
& x_{ij} \le y_i && i \in S,\;\; j \in A && (3)\\
& y_i \le \sum_{j \in A} x_{ij} && i \in S && (4)\\
& \tau_j = \phi_j(x_{1j},..,x_{nj}) && j \in A && (5)\\
& y_i \in \mathbb{N} && i \in S && (6)\\
& x_{ij} \in \mathbb{N} && i \in S,\;\; j \in A && (7)\\
& \tau_j \in \mathbb{R}^+ && j \in A && (8)\\
& (t,\gamma,\theta) \in \arg\min_{t,\gamma,\theta} \quad f_2(t,\gamma,\theta) && && (9)\\
& \qquad s.t. \quad t_j \ge r_j && j \in A && (10)\\
& \qquad\qquad t_j \le d_j - \tau_j && j \in A && (11)\\
& \qquad\qquad t_{\tilde{\jmath}} \ge t_{\hat{\jmath}} + \tau_{\hat{\jmath}} && (\hat{\jmath},\tilde{\jmath}) \in Q && (12)\\
& \qquad\qquad \sum_{j \in A} x_{ij}\gamma_{jc} \le y_i && i \in S,\;\; c \in A && (13)\\
& \qquad\qquad t_c - t_j \ge M(\gamma_{jc} - 1) && j \in A,\;\; c \in A && (14)\\
& \qquad\qquad t_c - t_j \le \tau_j + M(1 - \gamma_{jc}) - \varepsilon && j \in A,\;\; c \in A && (15)\\
& \qquad\qquad t_c - t_j \ge -M\theta_{jc} + \tau_j - \frac{\tau_j}{2}\gamma_{jc} && j \in A,\;\; c \in A && (16)\\
& \qquad\qquad t_c - t_j \le M(1 - \theta_{jc}) + \frac{\tau_j}{2}\gamma_{jc} - \varepsilon && j \in A,\;\; c \in A && (17)\\
& \qquad\qquad t_j \in \mathbb{R}^+ && j \in A && (18)\\
& \qquad\qquad \gamma_{jc} \in \{0,1\} && j \in A,\;\; c \in A && (19)\\
& \qquad\qquad \theta_{jc} \in \{0,1\} && j \in A,\;\; c \in A && (20)
\end{aligned}
$$

The upper and lower level objective functions are respectively contained in (1) and (9). In (2) are the bounds for variables x_{ij}. Constraints (3) and (4) express two concepts: the availability of operators with skill i must be (i) enough to guarantee that each activity can be independently executed (e.g. if scheduled in sequence with the others), and (ii) not more than the total amount of resources that would be needed if all the activities were parallelized. Relation (5) brings processing time simulations into the problem. Constraints (10) and (11) give release date and deadline limits, while inequalities (12) describe the precedence relations between activities.

In order to understand the meaning of constraints from (13) to (17), it is first necessary to clarify the role of binary variable γ. For each couple of activities (j, c), we have that γ_{jc} is equal to 1 if j is in progress when c is starting, 0 otherwise. Thus, we make use of the following double implication, which is guaranteed by inequalities (14)–(17) where ε and M are two appropriate small and large constants:

$$\gamma_{jc} = 1 \Leftrightarrow t_j \le t_c < t_j + \tau_j$$

Then, constraints (13) indicate the relation between available and allocated operators, i.e. the sum of resources simultaneously occupied cannot exceed the total number of workers, for each skill i.

Finally, (6)–(8) and (18)–(20) define variables domains. Notice that activities durations τ are black-box values varying on the positive side of the real axis. This assumes a particular meaning when the structure of lower-level formulation is analyzed: indeed, if we consider each τ_j to be externally calculated (once a value for every x_{ij} and y_i is fixed by the upper-level decision-maker) and f_2 to be the overall *makespan*, we can prove our problem to fall under the standard definition of Resource Constrained Project Scheduling Problem (see Artigues, Demassey and Néron (2008)), with additional due date constraints.

However, due to the a priori unknown values of processing times, an appropriate comparison of our formulation with existing ones makes sense only by considering analogous approaches, as those proposed by Artigues, Michelon and Reusser (2003) and Koné, Artigues, Lopez and Mongeau (2011), that admit continuous starting time variables and do not recourse to time horizon discretization. In this respect, it is worth making two observations: the first is that, similarly to authors just cited, we have developed a MILP formulation of the problem (that is evident when looking at upper-level variables as constants). The second, instead, captures the difference between our and previous approaches. In particular, by exploiting the relations between pairs of activities, we are able to formulate the same problem in a new way which differs and in some cases outperforms existing methods in terms of total amount of variables and constraints.

Anyway, the solution of the RCPSP constituting our lower-level optimization is not the only source of complexity in our procedure. The presence of black-box values calculated by simulators is an important issue to be addressed. For this reason, we propose a decomposition approach modeling the problem from a new grey-box optimization perspective.

3 Simulation-Optimization framework

Our solution framework is composed of three main nested blocks as shown in Figure 1. The most external one is a *black-box* optimization formulation working on variables y_i and x_{ij}, subject to constraints (2)–(4) and (6)–(7). Its objective function, denoted by $\tilde{f}$, has the structure of (1) and is calculated every time from the results of inner blocks.

In turn, the second module, represented by the resource constrained scheduling formulation described above, is (approximately) solved at every iteration, immediately after the execution of the third block, that takes the x_{ij} as inputs, runs a parallel simulation for each activity j, and returns the processing times τ_j.

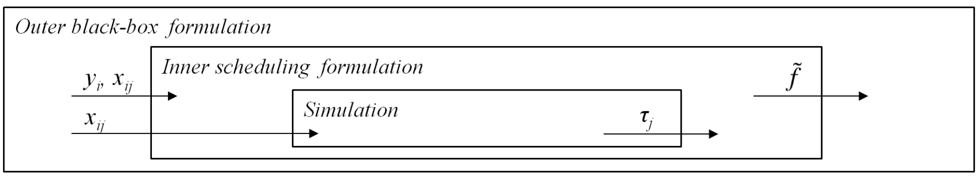

Fig. 1. Framework structure

References

Artigues C., Demassey S., Néron E., 2008, Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications, ISTE, London, UK.

Artigues C.,Michelon P.,Reusser S., 2003, "Insertion techniques for static and dynamic resource-constrained project scheduling", *European Journal of Operational Research*, Vol.149 pp.249-67.

Koné O., Artigues C., Lopez P., Mongeau M., 2011, "Event-based MILP models for resource-constrained project scheduling problems", *Computers & Operations Research*, Vol.38 pp.3-13.