

Project Scheduling with Multi Skilled Resources: a conceptual framework

Massimo Bertolini¹, Gionata Carmignani² and Francesco Zammori¹

¹ Dept. of Industrial Engineering, University of Parma, Parco Area delle Scienze 181/A,
43124, Parma, Italy
massimo.bertolini@unipr.it francesco.zammori@unipr.it

² Dept. of Energy, Systems, Territory and Construction Engineering, University of Pisa,
Largo Lucio Lazzarino 561, 56122 Pisa, Italy
gionata.carmignani@unipi.it

Abstract

Projects' success depends, mostly, on people's motivation and competences. A good plan is essential, but it is insufficient if the project manager is incapable to dynamically reassign people to project's tasks, so as to create multi-skilled teams and to avoid multi-tasking and over-allocation. In this regard, several models dealing with the "Multi Skilled Work Force Scheduling Problem" have been proposed, but unfortunately, most of the works produced so far has not yet found its way into practice. This is mainly because project scheduling and resources allocation are jointly considered, a fact that leads to complex and rigid mathematical formulations and that poses serious constraints on the precision of the input data. Since projects are, by their very nature, uncertain entities, we believe that it is preferable to abandon the over optimistic idea of a global optimum, in favour of a sub-optimal but stable and feasible solution. To this aim the paper proposes a heuristic framework that extends the well-known "Dynamic Scheduling" approach. Specifically, the problem is tackled in a hierarchical way: project scheduling is solved first and resource allocation is solved next, considering tasks durations as fixed constraints. In doing so, our focus is on the resources allocation phase, and the objective is to assure an almost perfect matching between resources' skills and tasks requirements, so as to assure project quality and, also, a harmonious development of the workforce. Possible approaches, based on mathematical programming, which could be easily implemented in project management software, are presented and discussed.

Keywords: Heuristics, Linear Programming, Project Scheduling, Multi Skilled Resources.

1. Introduction

Nowadays, more than ever, to survive in a ruthlessly marketplace, focusing on efficiency, resilience and flexibility is not enough, and companies must strive to innovate and to constantly improve their human capital. In this regard, Project Management (PM) has become critical in every business, not only in the Engineering-To-Order (ETO) ones (Vanhoucke, 2012). PM is an integrated managerial approach that makes uses of a vast set of instruments to plan and schedule tasks, to estimate resources efforts, to assess projects progress and to allocate, motivate and control human resources.

Anyhow, among the PM's activities, Project Scheduling (PS) and Resource Allocation (RA) are certainly the most important and tackling ones; indeed a proper schedule may do the difference between a successful and a failed project and, most of the times, project's success depends on people's motivation and competences (Lock, 2007).

Pioneering works on PS date back to the 50s when the Resource Constrained Project Scheduling Problem (RCPS) was formulated for the very first time. Initially, the main objective was to minimize the makespan or to maximize project's quality, without violating neither technological nor capacity constraints. Since then, this classic NP-hard problem has been extended under a plethora of different constraints and objectives (Artigues *et al.*, 2007). However, and quite surprisingly, resources have always been assumed as homogeneous and, consequently, tasks have always been considered in a *single mode*, with fixed duration and fixed resources requirements (Tiwarly *et al.*, 2009). It is easy to see that these assumptions contradict industrial practice, where, conversely, productivity depends on workers' skills and workers may be assigned to tasks for which they are not specialized. To our knowledge, only a few recent works have differentiated resources in terms of productivities and skills, in the so-called Skilled Work Force Scheduling Problem (SWFSP). Although most of the SWFSP

works concern staffing of software development projects (Barreto *et al.*, 2008; Chang *et al.*, 2008; André *et al.*, 2011), some examples can be found also in other fields (Heimerl and Kolisch, 2009; Valls *et al.*, 2009; Firat and Hurkens, 2011), with interesting papers combining the concept of resources allocation and projects selection (Yoshimura *et al.*, 2006; Gutjahr *et al.* 2008; Zaraket *et al.*, 2014). Clearly, also the SWFSP is a NP-hard problem and so, unfortunately, all proposed approaches relay on mathematical formulations that are far too complicated to be appealing for industrial practitioners (Herroellen and Leus, 2005, Acuña *et al.*, 2006). In our opinion, the misalignment between theory and practice can be traced in the fact that, the RCPSP and, lately, the SWFSP have been derived from the job-shop scheduling area; as such, the majority of the proposed algorithms operate in an integrated way, by considering, jointly, both technological and resources constraints. This allows finding optimal or quasi-optimal solutions, but it also poses serious constraints on the quality of the input data: due to complexity, even small perturbations of the inputs may lead to considerable variations of the end results (Penz *et al.*, 2001). This condition is fairly acceptable for job shop scheduling, but it is not for project scheduling: whereas, a job shop is a repetitive process, a project is a unique/unrepeatable event and the availability of reliable data is pure utopia.

To solve these criticalities, we believe that it is preferable to abandon the over optimistic search of a global optimum, in favor of a sub-optimal but robust and feasible solution. We also believe that the solution should be generated using an easy and intuitive approach, fully aligned to the decision making process followed by a project manager. Indeed, a project manager will never make use of an automatically generated solution unless he is confident of the results and, most of all, he can easily understand how results have been generated.

To this aim, the paper proposes a heuristic framework that extends the well-known Dynamic Scheduling (DS) approach (Vanhoucke, 2012). More specifically, by tackling the problem in a hierarchical way, PS is solved first and RA is solved next, considering tasks durations as fixed constraints. In doing so, our main focus is on RA and the objective is to assure an almost perfect matching between resources' skills and tasks requirements, so as to assure project quality and a harmonious development of the workforce. Indeed, the only possible way to overcome the *cost-quality-time* trade-off of PM, is that to exploit the capability, the motivation, the knowledge and the skills of the available resources. Only if the right resources are allocated to the project's tasks and only if the workloads are evenly distributed in time (without over allocation and/or multitasking) a project can be successfully completed (Certa *et al.*, 2007).

2. A consolidate approach: Dynamic Scheduling for project planning

Dynamic Scheduling (DS) is a step by step hierarchic framework typically adopted to solve project management problems. The underlying idea is to decompose a complex problem into less complex ones, which can be solved iteratively and sequentially. At each step a partial solution is generated and, if feasible, it is used as the input of the following step, so that a new and more refined one can be generated. After projects have been selected and prioritized, tasks are scheduled, satisfying both time requirements and technological constraints. Only afterwards, individual resources are assigned to tasks and, in case of over-allocation, the project is levelled and the final baseline is obtained. It is exactly the fact that tasks' scheduling and resources' allocation are considered one at a time that greatly simplifies the problem and makes it appealing also at the operational level: a project manager would never dare to allocate individual resources to tasks, unless an uncapacitated project plan has been generated first. The main steps of the DS approach are detailed below.

Project Selection (PS) - The first strategic choice concerns the selection of the projects to be activated. Many tools can be used, but generally the decision is based on the alignment of the projects with the overall strategy of the company and on their economical/financial performance. Next, selected projects are ordered and scheduled in time considering both

financial needs (i.e., cash flows) and the requirement of critical resources that could act as bottlenecks. Generally, before freezing the solution, a feasibility check is made using a Rough Cut Project Planning module, which verifies resources availability at a very aggregate level.

Uncapacitated Project Planning (UPP) - At a tactical level, projects are subdivided into tasks and, for each one of them, standard durations, start and finished dates and precedence constraints are defined, so that a Gantt chart and a preliminary budget can be made. It is important to note that the Gantt is generated in an uncapacitated way, considering an infinite availability of renewable resources. This is the reason why the resulting Gantt is generally referred as the Uncapacitated Baseline Schedule (UBS). However, although resources are considered limitless, there is the need to indicate, for each task, the number of standard resource (i.e., number of engineers, workers, electricians, artisans, etc.), which will be needed to complete the task in the allotted time. This has a twofold purpose. First of all, if the makespan is too long, standard durations may be reduced through the addition of extra standard resources. Secondly, and perhaps more important, a Capacity Requirement Planning module can check the availability of the required resource pools (over monthly or weekly time buckets) and verify the (aggregate) feasibility of the UBS.

Resource Allocation (RA) and Project Levelling (PL) - In this step, generic resources are substituted with individual ones i.e., people are selected from the resource pools to which they belong to and they are associated to project's tasks. In doing so, specific calendar and working days are considered. Once again, in case of unacceptable over allocations, resources can be substituted by equivalent ones, or, if needed, the whole project or part of it can be levelled using slack times, using over-times or pushing ahead the starting date of some tasks. Anyhow, after performing simulations and what-if-analysis, a final solution is chosen and the frozen Capacitated Baseline Schedule CBS is obtained. This baseline details tasks sequence, allocated resources, and expected costs (i.e., the detailed budget of the project).

Project Control (PC) - Lastly, at the operating level, the CBS is used to monitor the ongoing progress of the project. Time and cost variances are computed and used to reschedule the project's tasks.

3. An optimization engine for detailed resource allocation

Notwithstanding its consolidated practical use, the DS framework presents large space for improvement, especially in the RA step that is, by sure, the most challenging one. When resources are assigned to tasks, a first tackling problem is to find a schedule that avoids (or at least limits) over-allocations and, if possible, multi-tasking.

However, a schedule that achieves these objectives may not be enough to assure project's quality and due dates compliance. Indeed, resource allocation must assure, also, an almost perfect matching between resources' skills and tasks requirements; possible misalignments increase the training time of project's teams, it is one of the major causes of delays, of costs rising, and, lastly, of project's failures.

Also, in order to achieve a consolidated competitive edge, a good and robust schedule should exploit people's potentialities not only in the short, but also in the medium-long term. Fostering motivation and pursuing a well-balanced development of the work-force are critical issues of competitiveness, as knowledge and human resources represent the real long term value of a company. Thus, a good project manager should not commit the more challenging tasks always to the same experts. Job enlargement and job enrichment should be encouraged by assigning, from time to time, under-skilled resources to challenging tasks (under the supervision of senior workers), so as to enhance learning on the field. By doing so, less skilled workers will be valorised and a positive synergy will be obtained among team's members.

Unfortunately, at present, all the above mentioned decisions are totally delegated to the expertise of the project manager. Indeed, RA models proposed so far are definitely too hard to

be of any practical interest and none of them has found its way into practice, yet. Similarly, the aid offered by project management software is fairly limited and manager's expertise remains essential. RA is generally narrowed to over-allocations identification and levelling, with just the best of the shelf software implementing, at a very low level, the concept of skilled resources (Kastor and Sirakoulis, 2009).

In order to solve these criticalities and to automatize, at least partially, the RA step, we propose using an optimization engine, which should be integrated in a project management software to automatically generate the Assignments Matrix $A[x_{ij}]$, containing the assignment rate x_{ij} of resource i to task j , where $x_{ij}=1$ implies full assignment, $x_{ij}=0.5$ implies half assignment and values greater than one mean over allocations.

A conceptual representation of the optimisation engine, with a list of possible inputs, outputs, objectives and constraints, is sketched in Figure 1.

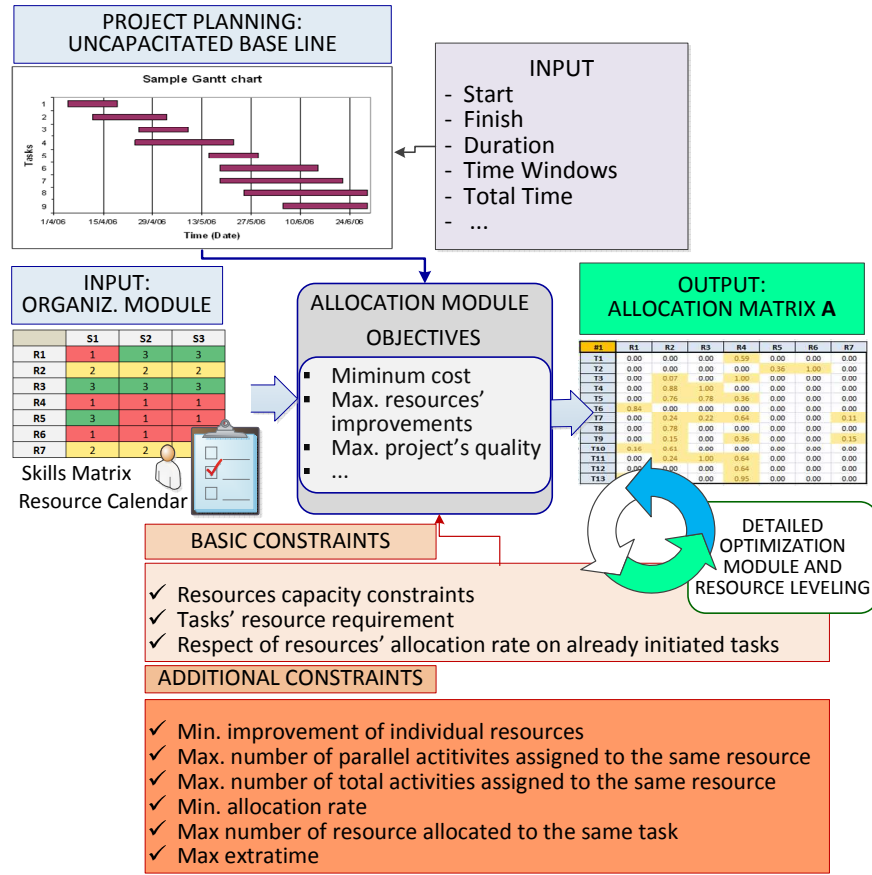


Figure 1. The optimization engine.

To this aim, regardless of the used objective function and of the adopted solution approach, for the optimization engine to work, there is the need to track and to formalize the specializations of the workers and their past field experiences, at least in terms of technical, executive and social/relational skills. Indeed, to generate a feasible solution, all scheduling constraints (expressed mainly in terms of resources' capacity and tasks' requirements) have to be fulfilled and so the optimization module must receive, as input: (i) the uncapacitated baseline schedule (containing tasks' durations, start dates and number of standard resources required by each task) and (ii) detailed information concerning resources' skills and task requirements. The latter input can be formalized with two skill matrices, one for the resources, namely Resources' skills matrix R , and one for the tasks, namely Tasks' requirement matrix T . More precisely, let $i \equiv \{1, \dots, n\}$, $j \equiv \{1, \dots, m\}$ and $k \equiv \{1, \dots, s\}$ denote resources, tasks and skills. Then $T[t_{jk}]$ is a $(m \times k)$ matrix and its generic elements t_{jk} indicates the level of skill k that is required to perform

task j in a standard way. Similarly, $\mathbf{R}[r_{ik}]$ is a $(n \times k)$ matrix and its generic elements r_{ik} indicates the level of skill k possessed by resource i . Once these matrices have been defined by the project manager, the idea is to match the values of $\mathbf{T}$ with that of $\mathbf{R}$, so as to identify the resources that, having all the skills required by a certain task, in intensity greater or equal than the minimum admissible level, can be assigned to that task. How to do so will be detailed in the Section 4, using a simple example as guideline. The same example will also be used to show how, an optimization module as the one of Figure 1, could be practically implemented using a simple linear programming model.

4. A numerical application

For the sake of clarity we will consider the simple project shown in the Gantt chart of Figure 2. The project is made of seventeen tasks, all tasks are connected through Finish-To-Start precedence constraints and the critical path, in red, is composed of five sequential tasks with a total length of forty time units (i.e., days). Also note that Task T₆ is a milestone (a dummy activity with null duration) that indicates the project end.

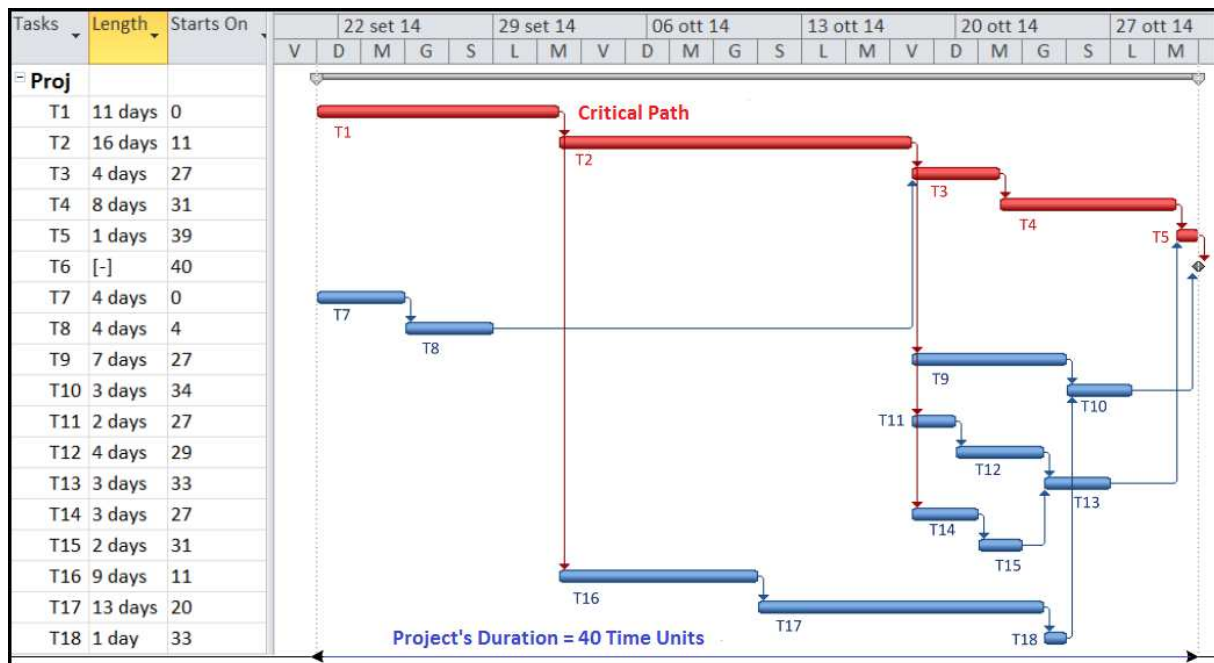


Figure 2. The project's Gantt chart.

The Resources skills matrix $\mathbf{R}$ and the Tasks requirements matrix $\mathbf{T}$ are shown by Table 1 and 2, respectively.

There are five resources differentiated in terms of three skills, and skill levels are quantified on a Likert-Type scale ranging from one to five. Table 1 also shows the classification label {strong - average - weak} and the cost for unit of time c_j , of each resource.

RESOURCE	SKILL 1	SKILL 2	SKILL 3	KIND	COST
R ₁	4.5	4.1	4.1	Strong	11.6
R ₂	3.5	3.9	3	Average	10.2
R ₃	3.2	3.9	3.6	Average	9
R ₄	2.1	2.9	2.3	Weak	8
R ₅	2.8	2.3	3.8	Weak	8.8

Table 1. Resources skills matrix $\mathbf{R}$.

Similarly, Table 2 shows the ideal skill levels required by each task j , its duration and the number of resources r_j , with adequate skill levels, that are needed to complete it in the allotted time. As above mentioned, appropriate values of r_j must be defined by the project manager during the uncapacitated project planning step of the DS approach. Also note that, typically, r_j would be integer, but fractional values could also be used in case of partial allocation. For instance, a strong and fully allocated resource, with skills levels of 4.5, 4.1, and 4.1 points, can complete T_1 in eleven units of time. Conversely, T_2 can be completed in sixteen days by a weak and partially allocated resource, with skill levels of 2.1, 2 and 2.4 points.

TASK	SKILL 1	SKILL 2	SKILL 3	r_j	Duration
T_1	4.5	4.2	4	1.0	11
T_2	2.1	2.0	2.4	0.5	16
T_3	2.1	2.3	2.4	1.0	4
T_4	2.1	2.0	2.3	0.5	8
....					
T_{16}	3.1	3.0	3.1	1.0	9
T_{17}	3.1	3.5	3.1	1.0	13
T_{18}	3.1	3.7	3.1	1.0	1

Table 2. Tasks requirements matrix T .

4.1 Resource's Productivity Rates

In order to understand which resources can be assigned to a task, there is the need to match the values of the matrixes R and T . In case of perfect matching (between resource's skills and task's requirements) it is doubtless that the resource can be assigned to the task. But what happens if a perfect match does not exist? To answer this question let $G_{ijk} = (r_{ik} - t_{jk})$ be the skill gap between resource i and task j , with respect to skill k . We define:

- *Standard resource* - a resource that has a perfect match with a task i.e., $G_{ijk} = 0 \forall k$;
- *Over skilled resource* - a resource for which $G_{ijk} \geq 0 \forall k$;
- *Under skilled resource* - a resource for which $\exists k$ s.t. $G_{ijk} < 0$.

Clearly, the assignability problem does not arise neither for standard nor for over skilled resources. Instead, the possibility to use under skilled resources depends on the value of the negative gap G_{ijk} and, generally, for each skill k , there will be a negative threshold value G_k , below which a resource cannot be assigned to a task without compromising its quality. This concept could be easily translated in a binary ($n \times m$) Incidence or Boolean matrix $B[b_{ij}]$, whose generic element b_{ij} is one if $G_{ijk} \geq G_k \forall k$ and it is zero otherwise.

Nonetheless, we believe that a straight use of the incidence matrix B would be inadequate for a full characterisation of the RA problem. Indeed, the skill gap G_{ijk} is related, not only to the quality of a task, but also to its duration; as a matter of fact, one can expect that, to complete a task, an under skilled resource requires more time than an over skilled one. This can be formalized substituting the incident matrix I with a more substantial ($n \times m$) Productivity matrix $P[p_{ij}]$, whose generic element p_{ij} corresponds to the productivity rate of resource i on task j . Typically the productivity rates p_{ij} will be defined by the project manager, more or less subjectively, but depending on the value of the skill gap G_{ijk} and provided that the following constraints are fully respected:

- $p_{ij} = 1$ if $G_{ijk} = 0 \forall k$;
- $p_{ij} > 1$ if $G_{ijk} \geq 0 \forall k$;
- $p_{ij} < 1$ if $G_{ijk} \geq G_k \forall k$ and $\exists k$ s.t. $G_{ijk} < 0$;
- $p_{ij} = 0$ if $\exists k$ s.t. $G_{ijk} < G_k$.

Alternatively, since p_{ij} must be an increasing function of G_{ijk} (i.e., the higher the positive gap the higher the productivity rate and vice versa), the productivity rates could be automatically computed by fitting a parametric S-shaped curve.

For instance, taking a value of two points as the maximum productivity rate, a possible analytical form could be:

$$\begin{cases} p_{ij} = \frac{2}{1 + \exp(-\alpha \cdot \bar{G}_{ij})} \text{ if } G_{ijk} \geq G_k \forall k \\ p_{ij} = 0 \text{ otherwise} \end{cases} \quad (1)$$

Where α is a positive shape parameter that determines the slope of the curve and $\bar{G}_{ij}$ is an aggregated value of the skill gaps G_{ijk} of resource i and task j , on all skills k .

For instance, using a weighted average to compute $\bar{G}_{ij}$, with weights obtained as in Eq. (3), setting $\alpha = 1.1$ and $G_k = 0.4 \forall k$, the productivity matrix of Table 3 can be obtained.

$$\bar{G}_{ij} = \sum_k w_{jk} G_{ijk} \quad (2)$$

$$w_{jk} = \frac{t_{jk}}{\sum_{k=1}^s t_{jk}} \quad (3)$$

	R ₁	R ₂	R ₃	R ₄	R ₅
T ₁	1.0	0.6	0.6	0.0	0.0
T ₂	1.8	1.6	1.6	1.1	1.4
T ₃	1.8	1.6	1.6	1.1	1.4
T ₄	1.8	1.6	1.7	1.2	1.4
...	...	...	...	...	...
T ₁₆	1.6	1.2	1.3	0.7	1
T ₁₇	1.5	1.1	1.2	0.6	0.8
T ₁₈	1.5	1.1	1.1	0.6	0.8

Table 3. Productivities matrix P .

4.2. The Scheduling Formula

The productivity rates should determine, not only the subset of the resources that can be allocated to a task, but also the actual time needed to complete it. This could seem a non-sense, since we have said that, according to the DS approach, tasks are scheduled first and resources are allocated next, without affecting either tasks' duration or project length. However, the real effect of the productivity rates is not a change in the length of a task, but rather it is a change in the assignment rate of the resources allocated to it. Specifically, if a resource is over skilled, then the activity duration will not be reduced, but rather it is the resource that will be partially allocated i.e., $x_{ij} < 1$; vice versa an under skilled resources will be over allocated i.e., $x_{ij} > 1$.

To better clarify this concept let us consider the scheduling formula of Eq. (4) which is commonly used to quantifies the Standard Work Load (SWL) of a task, after the uncapacitated baseline schedule has been created and the project manager has defined both the length l_j and the number of standard resources r_j of each task j .

$$SWL_j = l_j \cdot \sum(\text{Resources Assignment Rates}) = l_j \cdot \sum_i(x_{ij}) = (l_j \cdot r_j) \quad (4)$$

When, during the RA steps, individual resources are assigned to tasks, both SWL and l are considered fixed constraints and so the assignment rates x_{ij} are evaluated as follows:

$$\sum_i (x_{ij} \cdot p_{ij}) = r_j \quad (5)$$

To make an example let us consider task T_1 and resources R_1 and R_2 . From Table 3 and Eq.(4) it follows that T_1 has a standard work load of eleven man-days; also, it is easy to see that both resources can be assigned to T_1 ; yet, whereas R_1 is a standard resource (i.e., $p_{1,1} = 1$), R_2 is an under skilled one (i.e., $p_{2,1} = 0.6$). Thus, from Eq. (5), in order to complete T_1 in time, the assignment rate of R_1 should be $x_{1,1} = 1$, whereas that of R_2 should be $x_{2,1} = 1.68$.

Note that an assignment of 168% corresponds to 13.5 hours per day, which is unfeasible even admitting some overtime. Thus, to make the time schedule feasible, R_2 could be joined with R_3 , a resource that has the same productivity rate $p_{3,1} = 0.6$. If so, their assignment rates turn into $x_{2,1} = x_{3,1} = 0.83$ or, equivalently, to 6.7 hours per day.

It is also important to note that this time schedule is just a basic possibility; other combinations can be used and resources can rearrange their daily work as they prefer. For example, it may be advisable to work full time at the beginning of a task and, next, to progressively reduce the effort as the work is almost done. For instance, the use of “non-working-day”, as in the example of Table 4, is a very robust solution, because this arrangement creates a sort of protection (i.e., a time buffer) that can be used in case of unforeseen problems; obviously, if everything goes well, resource can perform other activities during the “non-working-times”.

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11
R_2	6.65 h	6.65 h	6.65 h	6.5 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h
R_3	6.65 h	6.65 h	6.65 h	6.5 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h	6.65 h
R_2	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	6 h	4 h	0 h
R_3	8 h	8 h	8 h	8 h	8 h	8 h	8 h	8 h	6 h	4 h	0 h

Table 4. Basic and modified time schedule.

Lastly we observe that, the length of task T_1 has been reduced (by one day) only apparently. Although the work will be performed for ten days, the eleventh one is used as a time buffer. Thus, the planned end of T_1 is not modified and even if everything goes well, all subsequent tasks will not begin before the eleventh day.

4.3. The Optimization Model

Now we have all the elements needed to formulate a linear programming model that can solve the RA problem, in terms of total cost minimization.

$$\left\{ \begin{array}{l} \min C = \sum_{i=1}^n \sum_{j=1}^n x_{ij} l_j c_i \\ \text{s.t.} \\ \sum_{i=1}^n x_{ij} p_{ij} = r_j \quad \forall j \\ \sum_{i=1}^n x_{ij} \leq 1 \quad \forall j \in J_t, \forall t \in \{t_1^S, t_2^S, \dots, t_m^S\} \\ x_{ij} \geq 0 \quad \forall i, \forall j \end{array} \right. \quad (6)$$

where:

- l_j, t_j^E, t_j^S are, respectively, the length, the start and the ending time of task j ;
- $J_t = \{j : t_j^S \leq t; t_j^E > t\}$ is the set of the tasks that are active in t (i.e., that start before t and end after t). In other words J_t contains all the tasks that are in parallel, at least in a neighborhood of t ;
- The first set of constraints - *Task's requirements constraints* - is used to assure the respect of the SWL of each task;
- The second set of constraints - *Resources' capacity constraints* - is used to avoid resources' over-allocations. Indeed, the cumulated assignment rates $\sum x_{ij}$ of the same resource i on the parallel tasks $j \in J_t$ is forced to be lower or equal than one;
- The last set of constraints - *Non negativity constraints* - imposes the positivity of the assignments rates x_{ij} .

To better explain the Resources' capacity constraints, let us consider the project's Gantt of Figure 3. In this case, although there are eighteen tasks, some of them start at the same time t , and so there are only eleven different starting times to be considered. More specifically, in order to identify parallel tasks is sufficient to create the sets $J_t = \{j : t_j^S \leq t; t_j^E > t\}$ for each one of the starting times $\{t_1^S = 0, t_2^S = 4, \dots, t_{11}^S = 39\}$. For example, for $t_1^S = 0$ only T1 and T7 are active at $t = 0$. These tasks are in parallel and so we have that: $J_0 = \{1, 7\}$ and that $(x_{i,1} + x_{i,7}) \leq 1 \forall i$.

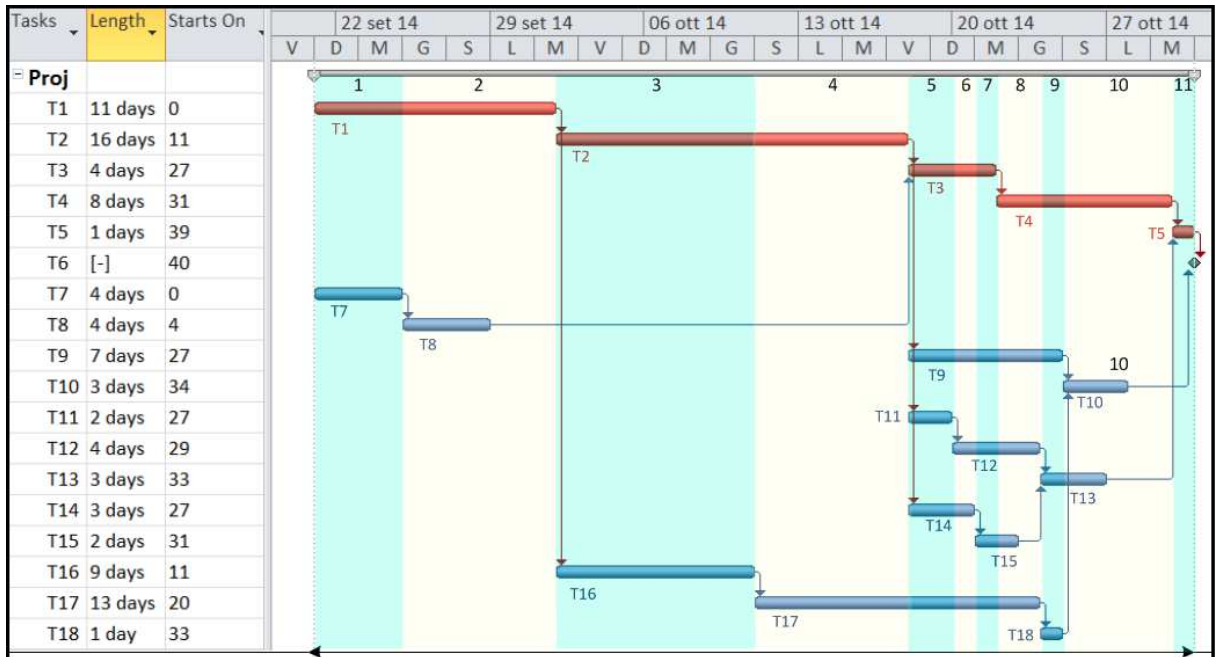


Figure 3. The over allocation constraints.

4.4. Model solution

The proposed linear optimization model can be optimally solved using the Simplex method; the obtained optimal assignment matrix A is shown in Table 5.

It is interesting to note that, in the optimal solution there are only four resources with positive assignment rates, while one resource (i.e., R_4) is never used. This is because, in the model, skills are used to define the productivity rates p_{ij} and, in turn, to determine the optimal assignment rates x_{ij} . Consequently, due to the cost minimization objective, resources having a high value of the "productivity to cost ratio" (p_{ij}/c_i), as R_3 and R_2 , tend to be allocated first

and/or more frequently. Instead, low skilled and quite costly ones, as R_4 , tend to be used less frequently and only when all the other more efficient resources are already saturated.

	T ₁	T ₂	T ₃	T ₄	T ₅	T ₇	T ₈	T ₉	T ₁₀	T ₁₁	T ₁₂	T ₁₃	T ₁₄	T ₁₅	T ₁₆	T ₁₇	T ₁₈
R₁	1.0	-	-	-	-	-	-	1.0	0.94	-	-	-	-	-	0.1	-	-
R₂	-	-	0.32	-	-	0.71	-	0.1	-	0.45	0.34	-	-	-	-	0.16	0.5
R₃	-	0.3	0.3	0.3	0.61	1.0	0.44	-	-	-	-	0.3	-	-	0.7	0.7	0.4
R₄	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R₅	-	-	-	-	-	-	-	-	-	0.22	0.31	-	0.7	0.7	-	-	-

Table 5. The optimal assignment matrix.

However, as previously noted, assigning challenging tasks always to the same experts is a strategy that pays off only in the short term. Conversely, to properly valorise less qualified workers and to create a positive synergy among the workers, from time to time also under-skilled resources should be assigned to some challenging tasks (under the supervision of a senior). In order to incorporate this feature in the model, an additional matrix is needed. This is the ($s \times s$) Skills relationships matrix $S[s_{kz}]$, whose generic element s_{kz} quantifies the positive link between skill k and skill z . In other words, S describes how different skills may help decreasing the learning time to become proficient in other areas. Thus, by matching the values of S with that of T and R , it is possible to define an upgrading rate per unit of time u_{ijk} relative to skill k of resource i assigned to task j (i.e., how much resource i can improve on skill k if assigned to task j).

This makes it possible to add constraints - *Skills improvements constraints* - such as Eq. (7), so as to ensure that, during the project, some resources may achieve an improvement, greater than an objective value K_k , on certain skills.

$$\sum_j (u_{ijk} \cdot x_{ij} \cdot d_j) \geq K_k \quad (7)$$

As for the productivity rate, also the upgrading rates must depend on the skill gaps. However, this time, the potential improvement should be high in case of slightly negative skill gaps and should be low or even null in case of slightly positive skill gaps. Also, the upgrading rate should be null when the absolute value of the gap $|G_{ijk}|$ is high. Indeed, a resource that is excellent in a skill cannot improve anymore and, similarly, a resource that is too poor cannot learn on the field. This behaviour can be easily modelled using the following quadratic function:

$$k_{ijz} = -\beta_1 \cdot (G_{ijk})^2 - \beta_2 \cdot G_{ijk} + \gamma \quad (8)$$

Adding these constraints the optimization model remains linear and non-integer and so it can be still optimally solved using the Simplex method. For instance, the results of Table 6 were obtained using the following values $\beta_1 = \beta_2 = 0.4$, $\gamma = 0.1$, and introducing the following three constraints that force the model to allocate resource R_4 , too:

- $\sum_j (u_{4j1} \cdot x_{4j} \cdot d_j) \geq 0.5K_{1,Max} \geq 0.12$
- $\sum_j (u_{4j2} \cdot x_{4j} \cdot d_j) \geq 0.5K_{2,Max} \geq 0.16$
- $\sum_j (u_{4j3} \cdot x_{4j} \cdot d_j) \geq 0.5K_{3,Max} \geq 0.28$

Where $K_{k,max}$ is the (maximum) increase of skill k that could be achieved if R_4 were fully allocated to all the tasks of the project.

Assignments that have change are clearly shown in Table 6. As it can be seen, R_4 has been allocated to T_2 and to T_{17} . More specifically, R_4 works alone on T_2 , which is an easy task, but

R₄ has been coupled with the senior resource R₃ on T₁₇, which is a quite tough task. This is certainly a better balanced solution.

	T ₁	T ₂	T ₃	T ₄	T ₅	T ₇	T ₈	T ₉	T ₁₀	T ₁₁	T ₁₂	T ₁₃	T ₁₄	T ₁₅	T ₁₆	T ₁₇	T ₁₈
R ₁	1.0	-	-	-	-	-	-	1.0	0.94	-	-	-	-	-	-	-	-
R ₂	-	-	0.25	-	-	0.71	-	0.1	-	0.63	0.34	-	-	-	-	-	0.5
R ₃	-	-	0.4	0.3	0.61	1.0	0.44	-	-	-	-	0.3	-	-	0.8	0.6	0.4
R ₄	-	0.45	-	-	-	-	-	-	-	-	-	-	-	-	-	0.5	-
R ₅	-	-	-	-	-	-	-	-	-	-	0.31	-	0.7	0.7	-	-	-

Table 6. The optimal assignment matrix with skill improvement constraints.

Certainly due to the addition of three constraints the overall cost has risen (about 4.5%) but, with the exception of R₅ (that is a highly skilled one), all resources have increased their skills, as clearly shown in Table 7.

	R ₁	R ₂	R ₃	R ₄	R ₅
Skill 1	0.201	0.014	0.113	0.186	-
Skill 2	0.207	0.031	0.032	0.158	-
Skill 3	0.146	0.001	0.032	0.290	-

Table 7. Obtained skills' improvements.

5. Conclusions and future works

In this paper we presented a new framework, based on the dynamic scheduling approach, to automatically allocate resources, so as to assure not only project quality and cost minimization, but also a continuous, integrated and well balanced development of the workforce's skills. The hope is to overcome the delicate *quality-cost-time* project management trade-off, by combining skills management and resource allocation: in this way project's scope and quality could be assured at a lower cost with a harmonious improvement of the human capital. This will increase firms' competitiveness by assuring, in the short term, the possibility to get a competitive edge in terms of reduced costs and time to market, and, in the long term, a well-balanced human resources development.

At the moment, the framework has been developed considering a simple programming model that has the advantage to be linear and non-integer, so that it can be optimally solved with the Simplex methods.

Perhaps, a limit of the model is as excessive fragmentation of the assignments (i.e., very low x_{ij} value can be used), but this can be easily solved by substituting $x_{ij} \geq 0$ with $x_{ij} \geq X_{min}$ in the non-negative constraints of Eq. (6). As a further enhancement to reduce work fragmentation (and multi-tasking), additional constraints could be added to limit both the number of resources that can be assigned to a task and the number of parallel tasks that can be assigned to the same resource. This would certainly increase the precision of the optimization engine, but the basic linear programming model would turn into an integer programming problem. The simplex could not be used and so optimality could not be assured anymore. Thus, in this case, it would be preferable to use heuristics, based on a set of constructive rules, which have a very quick computation time and allow performing what-if analysis, by simply altering the order with which they are executed. Developing and testing these heuristic models, and integrating the optimization engine in commercial project management software, will be the topic of future research activities.

6. References

- Acuña, S.T., Juristo, N and Moreno, A.M., 2006. Emphasizing human capabilities in software development, *IEEE*, 23, 94-101.
- André, M., Baldoquín, M.G. and Acuña, S.T., 2011. Formal model for assigning human resources to teams in software projects, *Information and Software Technology*, 53, 259-275.
- Artigues, C., Demassey, S. and Néron, E. *Resource-Constrained Project Project Scheduling Models*, Wiltshire, UK, 2007.
- Barreto, A., Barros, M. and Werner, C.M.L., 2008. Staffing a software project: a constraint satisfaction and optimization-based approach. *Computers and Operations Research*, 35, 3073-3089.
- Certa, A., Enea, M., Galante, G. and La Fata, C.M., 2007. Multi-objective human resources allocation in R&D projects planning. *International Journal of Production Research*, 47 (13), pp. 3503-3523.
- Chang, C.K., Jiang, H., Di, Y., Zhu, D. and Ge, Y., 2008. Time-line based model for software project scheduling with genetic algorithms. *Information and Software Technology*, 50, 1142-1154.
- Firat, M. and Hurkens. C.A.J., 2011. An improved MIP-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15, 363-380.
- Gutjahr, W.J., Katzensteiner, Reiter, S. P., Stummer, C. and Denk, M., 2008. Competence driven project portfolio selection, scheduling and staff assignment. *European Journal of Operations Research*, 16, 281-306.
- Heimerl, C. and Kolisch, R., 2009. Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets. *International Journal of Production Research*, 48 (13), 3759-3781.
- Herroelen, W. and Leus, R., 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165, 289-330.
- Kastor, A., Sirakoulis, K., 2009. The effectiveness of resource leveling tools for Resource Constraint Project Scheduling Problem. *International Journal of Project Management*, 27, 493-500.
- Lock, D., *The Essential of Project Management*, Gower Publishing Ltd, London, UK, 2007.
- Penz, B., Rapine, C. and Trystram, D., 2001. Sensitivity analysis for scheduling algorithms. *European Journal of Operational Research*, 134, 606-61.
- Tiawary, V., Patterson, J.H. and Mabert, V.A. (2009). Scheduling projects with heterogeneous resources to meet time and quality objectives. *European Journal of Operational Research*, 193, 780-790.
- Valls, V. Pérez A. and Quintanilla, S., 2009. Skilled workforce scheduling in Service Centres. *European Journal of Operational Research*, 193, 791-804.
- Vanhoucke, M., *Project management with dynamic scheduling*. Springer-Verlang, Berlin, Germany, 2012.
- Yoshimura, M., Fujimi, Y., Izui, K. and Nishiwaki, S., 2006. Decision-making support system for human resource allocation in product development projects. *International Journal of Production Research*, 44, (5), 831-848.
- Zaraket, F.A., Olleik, M. and Yassine, A.A., 2014. Skill-based framework for optimal software project selection and resource allocation. *European Journal of Operational Research*, 234, 308-318.