A Conceptual Framework for Project Scheduling with Multi-skilled Resources

Francesco Zammori
Department of Industrial Engineering
University of Parma
Parma, Italy

Massimo Bertolini
Department of Industrial Engineering
University of Parma
Parma, Italy

Abstract— **The success of a project depends, mostly, on the ability to create multi-skilled teams and to assign them to project's tasks, without creating multi-tasking and over-allocation. In this respect, the *Multi-skilled Work Force Scheduling Problem* has attracted a great interest, both for academicals and practitioners. Unfortunately, most of the academic works produced so far, has not yet found its way into practice, mainly because of a complex and rigid mathematical formulation, which poses a serious constraint on the precision of the input data. To solve this criticality, we abandon the over optimistic idea of a global optimum and we propose a hierarchical framework that extends the well-known *Dynamic Scheduling* approach. The focus is on the resource assignment phase, with the objective to allocate multi-skilled resources in a quasi-optimal way, so as to assure project quality and a harmonious development of the workforce.**

Keywords-heuristics; multi skilled resources; project scheduling

I. Introduction

New technologies and globalization, reduced product life cycles and the constant growth of internet technology, make the market competitive and turbulent. To respond to this ruthlessly competitive environment there is a solo option: accept the challenge, focusing on innovation and human resources management, or perish. Thus, investing in the direction of greatest engineering strength has become essential and Project Management (PM) has become critical, in terms of *costs-quality-time,* in every business [1].

Regardless of whether PM is considered a general managerial philosophy, or rather an integrated set of operating tools for planning, motivating, and controlling resources, it is unquestionable that Project Scheduling (PS) plays a major role within it, since a good schedule may do the difference between the success and the failure of a project [2]. Pioneering works on PS date back to the 50s when the *Resource Constrained Project Scheduling Problem* (RCPSP) was firstly introduced, with the objective to define a schedule, feasible in term of technological and capacity constraints, which optimizes makespan and/or project quality. Since then, this famous NP-hard problem has been extended under a plethora of different constraints and objectives and has been tackled with many heuristics procedures [3]. However, the availability of skilled resources is rarely included in the constraints of a RCPSP [4]. This fact is quite surprising, because the capability to assign multi-skilled workers to appropriate tasks and to compose teams effectively is an issue of great practical relevance. Possible misalignments, between workers skills and tasks requirements, increases the training time of project's teams and is one of the major causes of delays and project's failure [5]. Also, those few works that considered the availability of skilled resources are far too complicated to be appealing for industrial practitioners [6]. Indeed, in order to reach a global optimum, tasks' scheduling and resources allocation are jointly considered and this leads to a complex mathematical formulation, which poses a serious constraint on the precision of the input data; a condition that is unacceptable, because projects are unique and unrepeatable events and the availability of reliable data is utopic.

To solve these criticalities, we abandon the over optimistic idea of global optimization and we propose (at present only at a conceptual level) the basic architecture of a hierarchical framework, which is based on the well-known *Dynamic Scheduling* approach [1]. Specifically, our focus will be on the resource assignment phase, with the objective to allocate multi-skilled resources in a quasi-optimal way, so as to assure project quality, but also the harmonious development of the skills of the workforce. To assure robustness and intuitiveness, so as to encourage its industrial use and to facilitate its integration with PM software, scheduling procedure will make use of rule-based or, eventually, of integer programming based heuristics.

II. Dynamic Scheduling

Dynamic Scheduling (DS) is a hierarchic framework that allows solving, in a simplified way, complex RCPSP problems. The underlying idea is straightforward: starting from a problem based on simplified hypotheses and aggregated data, the solution is iteratively refined through a series of successive steps. At the end of each step a partial solution is obtained and, after a feasibility check, it is frozen and used as input for the following step. So, the original problem is decomposed into less complex sub-portions, which are solved iteratively and sequentially. Schematically we could say that, first the project manager schedules tasks, so as to satisfy time requirements and technological constraints, and next he allocates available resources to the previously scheduled tasks. Although this procedure cannot assure a global optimum, it generally leads to robust and feasible solutions, as empirically demonstrated by a long industrial use. More specifically, the main steps of the framework, shown by Figure I, are the following ones.

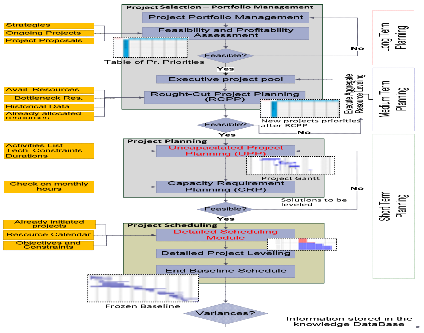

Project Selection – Portfolio Management
Project Portfolio Management
Feasibility and Profitability Assessment
Strategies
Ongoing Projects
Project Proposals
Table of Pr. Priorities
Feasible?
No
Yes
Long Term Planning
Executive project pool
Avail. Resources
Bottleneck Res.
Historical Data
Already allocated resources
Rought-Cut Project Planning (RCPP)
Execute Aggregate Resource Leveling
Medium Term Planning
New projects priorities after RCPP
Feasible?
No
Yes
Project Planning
Activities List
Tech. Constraints
Durations
Uncapacitated Project Planning (UPP)
Project Gantt
Check on monthly hours
Capacity Requirement Planning (CRP)
No
Solutions to be leveled
Feasible?
Yes
Short Term Planning
Already initiated projects
Resource Calendar
Objectives and Constraints
Project Scheduling
Detailed Scheduling Module
Detailed Project Leveling
End Baseline Schedule
Frozen Baseline
Variances?
Information stored in the knowledge DataBase

FIGURE I. DYNAMIC SCHEDULING

Project Selection - The Company has to decide, among a portfolio of potential projects, which ones should be activated. Obviously, besides being feasible and profitable, to be selected a project must be fully aligned with the overall strategy of the company. We also note that feasibility is checked with a RCPP module that assesses, at a very aggregate level, the availability of critical (i.e., bottleneck) resources.

The output of this step is the ordered list of the projects that will be activated in the next future.

Project Planning - For each selected project, a Gantt chart is generated using standard durations and scheduling tasks (generally as soon as possible) in a way that satisfies all technological constraints. If the makespan is too long, the project manager can reduce the standard duration of critical tasks, by accelerating their completion rate through the addition of extra resources. It is important to stress that, at this level, renewable resources are considered limitless and so the resulting Gantt is generally referred as the *uncapacitated baseline schedule*. Nonetheless, the project manager has to indicate, for each task, the number of resources, taken from available resource pools, that are judged necessary to complete the task in the established time (for instance 1 civil engineer and 3 workers may be needed to complete task 1 in 2 weeks). In this way the CRP module can check the availability of the required resource pools (over monthly or weekly time buckets) and verify the (aggregate) feasibility of the plan.

Project Scheduling - In case of feasibility, the project manager completes the schedule by substituting generic resources with individual ones. People are selected from the resource pools to which they belong to and are associated to project's tasks (i.e., *who does what*). Possible over-allocations are solved replacing resources with equivalent ones, using slack times, using over-times or, eventually, levelling the whole project or some parts of it. Anyhow, after performing simulations and what-if-analysis, a final solution is chosen and the base schedule is obtained.

Project Control - The base line is used to monitor the ongoing progress of the project. Time and cost variances are used to reschedule the project and their causes are investigated and recorded in the Company's knowledge database.

Notwithstanding its consolidated practical use, the DS framework presents large space for improvement. Indeed, being a hierarchical approach, mistakes made at a certain level will inevitably impact on all the lower levels and so, before moving from a level to the next one, it is vital to be sure of the decisions taken so far. This issue is particularly critical for the preliminary project definition (i.e., gathering of all the data need to build the project network) and for resources allocation. Both phases are totally left to the judgment and to the experience of the project manager; even the aid offered by PM software is fairly limited, with just a few applications in the area of resource management and resource leveling [7]. Due to these issues, in the next Sections we will discuss, at a conceptual level, how the resource allocation phase could be automated, at least partially, and fully integrated in PM software, so as to reduce the effort required by the project manager and the probability of making mistakes.

III. RESOURCE ALLOCATION OPTIMIZATION ENGINE

To enhance the project scheduling step, we propose using an optimization engine as the one shown by Figure II. The engine receives as input the project's baseline and the skills' matrices and, based on a set of reconfigurable heuristics and/or constructive procedures for multi-skilled resource assignment, it generates the *Assignments* matrix $\boldsymbol{A}[x_{ij}]$ containing the allocation rate x_{ij} of each individual resource i to each task j. This is made in a way that satisfies a set of constraints and that "maximizes" a set of selectable objective functions. In case of multi-projects, the list of the ongoing projects and of the already allocated resources is an additional input.

Specifically, *skills-matrices* are used to track the specializations of the workers and their past field experiences; technical, executive and social/relational skills must be considered and at least two matrices should be defined. These are: the *Tasks' required skills* matrix $\boldsymbol{Ts}[t_{jk}]$ and the *Resources' skills* matrix $\boldsymbol{Rs}[r_{ik}]$. More precisely, let $i \equiv \{1,\ldots, n\}$, $j \equiv \{1,\ldots, m\}$ and $k \equiv \{1,\ldots, s\}$ denote resources, tasks and skills. Then $\boldsymbol{Ts}[t_{jk}]$ is an $m\times k$ matrix and its elements t_{jk} indicate the level of skill k that is required to perform task j in a standard way (generally $0 \le t_{jk} \le 5$). Similarly, $\boldsymbol{Rs}[r_{ik}]$ is an $n\times k$ matrix and its elements r_{ik} indicate the level of skill k that is possessed by resource i.

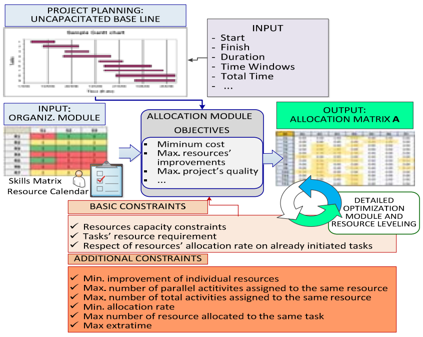

PROJECT PLANNING: UNCAPACITATED BASE LINE
INPUT
- Start
- Finish
- Duration
- Time Windows
- Total Time
- ...
INPUT: ORGANIZ. MODULE
Skills Matrix
Resource Calendar
ALLOCATION MODULE
OBJECTIVES
Miminum cost
Max. resources' improvements
Max. project's quality
...
OUTPUT: ALLOCATION MATRIX A
DETAILED OPTIMIZATION MODULE AND RESOURCE LEVELING
BASIC CONSTRAINTS
✓ Resources capacity constraints
✓ Tasks' resource requirement
✓ Respect of resources' allocation rate on already initiated tasks
ADDITIONAL CONSTRAINTS
✓ Min. improvement of individual resources
✓ Max. number of parallel actitivites assigned to the same resource
✓ Max. number of total activities assigned to the same resource
✓ Min. allocation rate
✓ Max number of resource allocated to the same task
✓ Max extratime

FIGURE II. OPTIMIZATION ENGINE

Evidently, by matching the values of $\boldsymbol{Ts}[t_{jk}]$ with that of $\boldsymbol{Rs}[r_{ik}]$, one can identify the subset of the available resources that possess all the skills required by a certain task, in an intensity greater or equal than the minimum admissible level (i.e., positive skill gap). This can be formalized in a binary $n\times m$ *Incidence* matrix $\boldsymbol{B}[b_{ij}]$, whose elements b_{ij} equal one if $r_{ik} \ge t_{jk}$ $\forall$ k = 1,..,s and zero otherwise. A further enhancement of the model can be obtained by substituting the incidence matrix $\boldsymbol{B}[b_{ij}]$, with an $n\times m$ *Productivity* matrix $\boldsymbol{Pr}[p_{ij}]$, whose values p_{ij} correspond to the productivity rate of resource i on task j. The hypothesis is made that, above a certain threshold limit T_{jk}, also a resource with a negative skill gap can be assigned to a task, but, depending on the negative gap, this resource will require more time to accomplish the task. Resource productivity depends on the matching between resource skills and task's requirements: the greater the positive gap, the faster a resource can accomplish a task and vice versa.

Typically the productivity rates p_{ij} will be defined by the project manager, more or less subjectively, provided that the following constraints are respected:

- $p_{ij} = 1$ if $r_{ik} = t_{jk}$ $\forall$ $k = 1,..,s$
- $p_{ij} > 1$ if $r_{ik} \geq t_{jk}$ $\forall$ $k = 1,..,s$ and $\exists$ k s.t. $r_{ik} > t_{jk}$
- $p_{ij} < 1$ if $r_{ik} \leq t_{jk}$ $\forall$ $k = 1,..,s$ and $\exists$ k s.t. $r_{ik} < t_{jk}$
- $p_{ij} = 0$ if $\exists$ k s.t. $r_{ik} < T_{jk}$

Alternatively, p_{ij} could be obtained, by fitting a parametric *S-shaped* curve. For instance, if the maximum productivity rate $P = 2$, a possible analytical form could be:

$$p_{ij} = 2 \big/ \left(1 + \exp\left\{-\alpha \cdot \Delta G_{ij}\right\}\right) \qquad (1)$$

where α is a shape parameter, ΔG_{ij} is the skills gap that, letting w_{jk} be the importance of skill k for task j, could be obtained as:

$$\Delta G_{ij} = \sum_j w_{jk}\left(r_{ik} - t_{jk}\right) \text{ with } \sum_j w_{jk} = 1 \qquad (2)$$

As mentioned in Section II, in order to define the uncapacitated baseline, the project manager has to indicate, for each task j, the number of standard resources R_j required to complete it in the allotted time. Typically R_j would be integer (i.e., resources are fully allocated and $x_{ij} = 1$), but real values could also be used in case of partial allocation. Also, when skills matrices are used, allocating a standard resource is equivalent to assign to a task a resource whose skills' levels perfectly match with those required by the task i.e., $r_{ik} = t_{jk}$ $\forall$ k, $\Delta G_{ij} = 0$ and $p_{ij} = 1$. Thus, we can write that, for the uncapacitated project plan, the following equation must hold:

$$\sum_i x_{ij} \cdot p_{ij} = \sum_i x_{ij} = R_j \forall j = 1,\ldots,m \qquad (3)$$

Obviously, to assure product quality, the same condition must also be verified when, during the detailed scheduling phase, standard resources are replaced by individual ones. In this case, since p_{ij} values do not necessary equal 1, Eq. (3) can be used to determine the (minimum) allocation percentage x_{ij} of resource i on task j. Indeed, accordingly to the DS hierarchical framework, the hypothesis is made that tasks are scheduled first and that only next resources are allocated, without affecting either tasks' duration or project length. Since tasks durations are considered as fixed constraints, if a resource has a productivity rate greater than one, then the activity duration will not be reduced, but rather it is the resource that will be *"partially allocated"* i.e., $x_{ij} < 1$. It is exactly the fact that tasks' scheduling and resources allocation are considered one at a time (task scheduling first and resources allocation next) 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. Due to these issues, resource allocation can be optimally solved using the linear programming model of Figure III.

$\min C = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} d_j c_i$	Minimum cost
$\sum_{i=1}^{n} x_{ij} p_{ij} = R_j \quad \forall j$	Resources requirement
$\sum_{j \in J_t} x_{ij} \leq 1 \quad \forall i, \forall t = \{t_j^s\}$	Resources capacity
$x_{ij} \geq 0 \ \forall i, j$	Positivity of assignment rates
$x_{ij} = \tilde{x}_{ij}, \forall j \in J_0$	Resources allocated to already started tasks

Where:

c_i is the standard unitary cost of resource i
d_j and t_j^s are the duration and the start time of tasks j
$\tilde{x}_{ij}$ are the resources' assignments of already started tasks
J_0 is the set of tasks started before t_0 and still in progress
J_t is the set of tasks that are active at a certain time t

FIGURE III. BASIC LINEAR PROGRAMMING MODEL

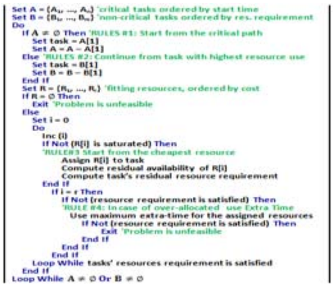
In this case the project is in t_0 and resources must be assigned to the tasks beginning in $[t_0; t_0 + T]$, violating neither capacity constraints nor resource requirements. Note that the *"resources capacity constraints"* assure that the sum of the assignment rates x_{ij} (of the same resource i on parallel tasks $j \in J_t$) is less or equal than 1. Also note that, in the basic linear programming model of Figure III, skills are used only to define the productivity 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) tend to be allocated first and/or more frequently.

However, to foster job motivation and to get a harmonious improvement of the human capital, the project manager should avoid committing the more challenging tasks always to the same experts. Job enlargement and job enrichment should be searched by assigning, from time to time, under-skilled resources to some challenging tasks (under the supervision of a senior), so as to enhance learning on the field. By doing so, less skilled workers will be valorized and a positive synergy will be obtained among team's members. In order to incorporate this feature in the model, an additional matrix is needed. This is the $s\times s$ *Skills relationships* matrix $\boldsymbol{Sr}[y_{kz}]$, whose (non-null) elements y_{kz} quantify the "positive correlation" between skill k and skill z. In other words, $\boldsymbol{Sr}[y_{kz}]$ describes how different skills may help decreasing the learning time to become proficient in other areas. By matching the y_{kz} values with that of $\boldsymbol{Ts}[t_{jk}]$ and $\boldsymbol{Rs}[r_{ik}]$, it would be possible to define an improvement rate per unit of time (I_{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). Thus, constraints such as Eq. (4) could be introduced in the model to ensure that, during the project, some resources may achieve an improvement greater than K on certain skills.

$$\sum_j I_{ijk} \cdot x_{ij} \cdot d_j \geq K_i \tag{4}$$

Lastly, to avoid an excessive fragmentation of the assignment (i.e., too low assignment rate x_{ij}) additional constraints could be added, such as: maximum number of resource that can be assigned to a task, minimum assignment rate x_{ij}, maximum number of parallel tasks that can be assigned to the same resource, use of over-time, etc. 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.

As an alternative, the optimization engine could make use of 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 [8]. For instance, in order to minimize total cost, four rules could be used, as in the pseudo code of Figure IV. Selected rules could also be combined to define more refined constraints, such as: *"each team should have at least one member of high technical skills"*, *"the team leader must have high executive skills and average technical skills"*, etc. By operating in this way, the scheduling problem turns into a Constraint Satisfaction Problem (CSP) and can be solved in several ways, such as backtracking.



```
Set $A = \{A_1, \ldots, A_a\}$ 'critical tasks ordered by start time
Set $B = \{B_1, \ldots, B_b\}$ 'non-critical tasks ordered by res. requirement
Do
  If $A \neq \emptyset$ Then 'RULES #1: Start from the critical path
    Set task = A[1]
    Set A = A – A[1]
  Else 'RULES #2: Continue from task with highest resource use
    Set task = B[1]
    Set B = B – B[1]
  End If
  Set $R = \{R_1, \ldots, R_r\}$ 'fitting resources, ordered by cost
  If $R = \emptyset$ Then
    Exit 'Problem is unfeasible
  Else
    Set i = 0
    Do
      Inc (i)
      If Not (R[i] is saturated) Then
      'RULE#3 Start from the cheapest resource
          Assign R[i] to task
          Compute residual availability of R[i]
          Compute task's residual resource requirement
      End If
        If i = r Then
          If Not (resource requirement is satisfied) Then
          'RULE #4: In case of over-allocated  use Extra Time
            Use maximum extra-time for the assigned resources
              If Not (resource requirement is satisfied) Then
                  Exit 'Problem is unfeasible
              End If
          End If
        End If
    Loop While tasks' resources requirement is satisfied
  End If
Loop While $A \neq \emptyset$ Or $B \neq \emptyset$
```

FIGURE IV. A RULE BASED HEURISTIC

IV. CONCLUSIONS AND FUTURE WORKS

In this paper we presented an innovative framework capable to automatically allocate resources, assuring project quality, budget compliance and a continuous, integrated and well balanced development of the workforce's skills. The hope is to overcome the delicate *quality-cost-time* tradeoff 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 only at a conceptual level and so further researches will be made to integrate it in PM software and to test it in real industrial cases.

REFERENCES

[1] M. Vanhoucke, Project management with dynamic scheduling, Springer-Verlang, Berlin, Germany, 2012

[2] D. Lock, The Essential of Project Management, Gower Publishing Ltd, London, UK, 2007.

[3] C. Artigues, S. Demassey and E. Néron, Resource-Constrained Project Project Scheduling: Models, Wiltshire, UK, 2007.

[4] A. Certa, M. Enea, G. Galante and C.M. La Fata, 2007 "Multi-objective human resources allocation in R&D projects planning". International Journal of Production Research, vol. 47, No. 13 , pp. 3503–3523,

[5] A. Barreto, M. Barros and C.M.L. Werner, C.M.L, 2008 "Staffing a software project: A constraint satisfaction and optimization-based approach", Computers and Operations Research, vol. 35, pp. 3073-3089.

[6] S.T. Acuña, N. Juristo and A.M. Moreno, 2006 "Emphasizing human capabilities in software development", IEEE are, Vol. 23, pp. 94-101.

[7] Levin, H. (2011). How to allocate resources and build optimal teams, AtTask publication, available online www.attask.com.

[8] Myszkowski, B., Skowronski, M.E. and Podlodowski, L., (2013). Novel heuristic solutions for Multi–Skill Resource–Constrained Project.