

# Advanced Topics in Project Management Process

Lucio Bianco and Massimiliano Caramia

Dipartimento di Ingegneria dell'Impresa, Università di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy. e-mail: {bianco,caramia}@disp.uniroma2.it

**Abstract.** In this paper we present the evolution of project scheduling models with emphasis on temporal constraints.

**Keywords:** Project scheduling, Generalized precedence relations, Resource constraints, Feeding precedences.

## 1 Introduction

In today's competitive global environment it is crucial to deliver quality product on time and within an established budget. "Management by projects" has become the principal way to pursue this goal since it is a powerful way to integrate organizational functions and motivate groups to achieve higher levels of performance and productivity. Therefore it is not surprising to see that, in the last few decades, project management has become a hot topic and has been recognized as a central management discipline. Major industrial companies now use project management as their principal management style. Nevertheless the impression is that the contribution that project management can make to management at all is still underrated and poorly known. Moreover, in spite of numerous books on the argument, a very limited number of these put emphasis on the mathematical tools and on the quantitative methods necessary to make project management more effective in dealing with increasing complexity organizations. To better understand this aspect, it is useful to recall that project management is a process which involves the planning, scheduling and control of project activities to achieve performance, cost and time objectives for a given scope of work, while using resources efficiently and effectively (see Lewis, 1995). The *planning phase* involves the following crucial steps: identify the project activities, their interdependencies, and estimate times, resources and costs necessary to perform them. The *scheduling phase* involves the construction of a project base plan which specifies, for each activity, the precedence and resource feasible start and completions times, the amounts of the various resource types necessary during each time period, and, as result, the budget. The *control phase* focuses on the comparison between the project base plan and the actual performance once the project has started. If this comparison reveals some difference on the schedule, or on the budget or also on some technical specifications, a corrective action must be taken to get the project back on the right track.

In this paper we mainly focus on project scheduling phase often identified, on the basis of management experience, as the most critical with respect to project failure (Goldratt, 1997, Lewis, 1995). In fact, especially in the case of product innovation and new product development projects, delays in the completion time are more heavily penalized than budget overruns (Goldratt, 1997).

Suri (1998) reports an experience of Hewlett-Packard related to a high-growth market which confirms the greater importance of time schedule than the cost. In fact, Hewlett-Packard found that a delay of six months in a new product project could cause a loss of 33% in profits while, if the project overran its cost by 50% but was completed on time, the loss was only 3.5%. For this reason we consider the project scheduling problem assuming that all input data are known in advance (deterministic case) and that the objective is minimizing project duration. Of course other objective functions, like resource levelling, resource cost, tardiness or earliness, could be considered. In particular, we analyse the evolution of project scheduling models with emphasis on temporal constraints. We start from the well known basic case of resource unconstrained project scheduling with Finish-to-Start precedence constraints and zero time lag (PERT, CPM). Then we analyse the problems arising when other types of precedence constraints with minimum and maximum time lags are considered, that is the “Generalized Precedence Relations” (GPRs).

The next step is Resource Constrained Project Scheduling Problem (RCPSP) with GPRs. With respect to the previous problems, that from the computational point of view are polynomial, the RCPSP is NP-hard and not optimally solvable for large instances, within a reasonable computing time.

Finally a further extension of the classical RCPSP, arising when the activity durations are not defined and known in advance, is considered. This problem happens in that production planning environment, like make-to-order manufacturing, where the resources are often shared among different simultaneous activities in proportion variable over time.

In this case the duration of each activity is not univocally defined, and consequently the traditional Finish-to-Start precedence constraints, so as the GPRs, cannot be used any longer. Then, we need to introduce the so called “Feeding Precedences” (FP) (Kis, 2006) defining in such way a new problem: Resource Constrained Project Scheduling Problem with Feeding Precedences (RCPSP-FP).

## **2 The Resource Unconstrained Project Scheduling Problem**

A project consists of a number of events and activities that have to be performed according to a set of precedence constraints. Each event refers to a stage of accomplishment of a certain activity (in general its starting time or its finishing time). Each activity has a duration and typically requires resources for its execution.

In order to represent activities and their relationships it is a common practice to use networks; the latter allow one to represent precedence relationships more efficiently than other representation techniques like Gantt charts (Clark, 1954), track planning (Herroelen, 1998), and line balance planning (Lumsden, 1968).

A project network is a graph  $G = (N, A)$  consisting of a set  $N$  of nodes and a set  $A$  of arcs. There are two possible representations of a project by means of a graph  $G$ . One is the *Activity-On-Arc-representation* (AOA) which uses the set of arcs  $A$  to represent the activities and the set of nodes  $N$  to represent the events; the other is the *Activity-On-Node representation* (AON) which uses the set of nodes  $N$  to denote the activities and the set of arcs to represent the precedence relations, i.e., an arc  $(i, j)$  in  $A$  means that activity  $i$  constrains or is constrained by activity  $j$ .

If we assume that resources are unlimited, or are available in huge amounts, a project network is in general used into two fundamental techniques: CPM (Critical Path Method) and PERT (Program and Evaluation Review Technique). These two methodologies study the temporal development of a project taking into account only precedence relations among activities of the so called Finish-to-Start type, i.e., if an arc  $(i, j)$  exists in an AON representation then activity  $j$  cannot start before the finishing time of activity  $i$ . The temporal analysis gives answers to some important questions like how long the project will take, i.e., what is the completion time (makespan) of the project; how early a particular activity may be started; how far an activity can be delayed without causing a delay of the entire project.

It is well known in project scheduling that PERT/CPM methods can be applied under two assumptions. The first one is that resources are available in infinite amounts, and the second is that the precedence relationships between two activities are only of the Finish-to-Start type with time-lags equal to zero; this implies that an activity can start only as soon as all its predecessors activities have finished.

In this context, one can define an acyclic project network and compute the minimum project completion time as the length of the *critical path*, i.e., the longest path from the initial activity (source node) to the final activity (sink node) in such an activity network (see, e.g., Moder et al., 1983, and Radermacher, 1985).

The computation of the critical path can be accomplished by means of the well-known forward pass recursion algorithm (see, e.g., Kelley, 1963), that is a classical label setting algorithm for longest path calculation. The computational complexity of this algorithm is  $O(m)$ , where  $m$  is the number of arcs of the network; when the graph has a high density, the complexity tends to  $O(n^2)$ , where  $n$  is the number of activities.

However, in a project, it is often necessary to specify other kinds of temporal constraints besides the Finish-to-Start precedence relations with zero-time-lags. Accordingly to Elmaghraby and Kamburoski (1992), we denote such constraints as *Generalized Precedence Relations* (GPRs). GPRs allow one to model minimum

and maximum time-lags between a pair of activities (see, e.g., Dorndorf, 2002, and Neumann et al., 2002).

Four types of GPRs can be distinguished: Start-to-Start ( $SS$ ), Start-to-Finish ( $SF$ ), Finish-to-Start ( $FS$ ) and Finish-to-Finish ( $FF$ ).

A minimum time-lag ( $SS_{ij}^{\min}(\delta)$ ,  $SF_{ij}^{\min}(\delta)$ ,  $FS_{ij}^{\min}(\delta)$ ,  $FF_{ij}^{\min}(\delta)$ ) specifies that activity  $j$  can start (finish) only if its predecessor  $i$  has started (finished) at least  $\delta$  time units before.

Analogously, a maximum time-lag ( $SS_{ij}^{\max}(\delta)$ ,  $SF_{ij}^{\max}(\delta)$ ,  $FS_{ij}^{\max}(\delta)$ ,  $FF_{ij}^{\max}(\delta)$ ) imposes that activity  $j$  can be started (finished) at most  $\delta$  time slots beyond the starting (finishing) time of activity  $i$ .

In order to better understand GPRs we report some examples. If, for instance, a company must place a pipe (activity  $j$ ) in a given region, it is necessary to prepare the ground (activity  $i$ ) in advance. This situation can be represented by a constraint  $SS_{ij}^{\min}(\delta)$ , since the start of activity  $j$  must be  $\delta$  units of time after the starting time of activity  $i$ . In another project, if a company must supply a client with a certain number of products which must be also assembled within 100 days, this relationship can be modeled as  $SF_{ij}^{\max}(100)$ , which says that the assembly process (activity  $j$ ) must finish at most 100 days after the starting time of the delivery (activity  $i$ ) of the products.

GPRs can be represented in a so-called *standardized* form by transforming them e.g. into minimum Start-to-Start precedence relationships by means of the so-called Bartusch et al.'s transformations (Bartusch et al., 1988). Thus, applying to a given AON activity network with GPRs such transformations leads to a *standardized activity network* where associated with each arc is a label  $\ell_{ij}$  representing the time-lag between the two activities  $i$  and  $j$  (De Reyck, 1988). If more than one time-lag  $\ell_{ij}$  between  $i$  and  $j$  exists, only the largest  $\ell_{ij}$  is considered.

It should be noted that standardized networks, unlike the networks without GPRs, may contain cycles. This implies that a Resource Unconstrained Project Scheduling Problem (RUPSP) with GPRs and minimum completion time objective:

- cannot be solved by the classical forward and backward recursion algorithm mentioned before;
- can be solved by computing the longest path from the source node to the sink node, whose length is the sum of the time-lags associated with its arcs, using an algorithm working on “general” networks with a worst-case complexity  $O(n \cdot m)$  (see, e.g., Ahuja et al., 1993);
- does not admit feasible solutions when the topology of the network and the arc-length function induce one or more directed cycles of positive length (Bartusch et al., 1988).

In Bianco and Caramia (2008, 2010), it has been proposed a new network formulation and a new mathematical program for the RUPSP with GPRs which can be easily solved in  $O(m)$  time by means of dynamic programming. Starting from this new complexity result, it has been further proved that, by exploiting the dual formulation of the proposed mathematical program, the minimum

completion time can be also computed, with the same computational complexity  $O(m)$ , by finding an augmenting path of longest length in the proposed acyclic network in which a unit capacity is installed on each arc. An experimental study comparing the performance of the proposed approach and the traditional one confirmed the better performance of the former method also in practice.

### 3 The Resource Constrained Project Scheduling Problem with GPRs

The basic Resource Constrained Project Scheduling Problem (RCPSP) deals with scheduling project activities subject to Finish-to-Start precedence constraints with zero time-lags and renewable resources under the minimum project completion time objective. The RCPSP is NP-hard in strong sense (see, e.g., Blazewicz et al., 1983). When this kind of temporal constraints are taken into account, an activity can start only as soon as all its predecessors have finished, and, therefore, the resource constraints exist only between two or more independent activities.

As we mentioned in the previous section, when no resource constraint is concerned, GPRs can be represented in a so called “standardized” form by transforming them, for instance, in minimum Start-to-Start precedence relationships obtaining a possibly cyclic network (see Bartusch et al., 1988), or can be represented by means of an acyclic network by transforming them into Finish-to-Start relations (Bianco and Caramia, 2008, 2010).

When resources come into play, the RCPSP with GPRs is strongly NP-hard; also the easier problem of detecting whether a feasible solution exists is intractable (NP-complete, Bartusch et al., 1988). To the best of our knowledge, the exact procedures presented in the literature are the branch-and-bound algorithms by Bartusch et al. (1988), Demeulemeester and Herroelen (1997b), and De Reyck and Herroelen (1998). The paper by Bartusch et al. (1988) reports a limited computational experience on a case study; the paper by Demeulemeester and Herroelen (1997b) is conceived to work on minimum time-lags only; the third approach, that works with both minimum and maximum time-lags, presents results on projects with 30 activities and percentages of maximum time-lags of 10% and 20% with respect to the total number of generalized precedence relations.

Also lower bounds are available for this problem. In particular, two classes of lower bounds are well known in the literature, i.e., constructive and destructive lower bounds. The first class is formed by those lower bounds associated with relaxations of the mathematical formulation of the problem (for instance, the critical-path lower bound and the basic resource-based lower bound; see, e.g., Demeulemeester and Herroelen, 2002). Destructive lower bounds, instead, are obtained by means of an iterated binary search based routine as reported, e.g., in Klein and Scholl (1999). Also De Reyck and Herroelen (1998) proposed a lower bound for the RCPSP-GPRs denoted with  $lb3 - gpr$  that is the extension of the lower bound  $lb_3$  proposed by Demeulemeester and Herroelen (1997a) for the RCPSP.

Recently, Bianco and Caramia (2009a) proposed a new lower bound for the RCPSP-GPRs based on a relaxation of the resource constraints among independent activities and on a solution of the relaxed problem suitably represented by means of an AON acyclic network. In particular, for the project scheduling problem with GPRs and scarce resources, the authors exploited the network model proposed in Bianco and Caramia (2008, 2010) and tried to get rid of the resource constraints. The analysis has been restricted to only those pairs of activities for which a GPR exists to determine a lower bound on the minimum project completion time. For each of these pairs it has been verified whether the amount of resources requested exceeds the resource availability, for at least one resource type. In case of a positive answer, some results which allow the reduction of the problem to a new RUPSP with different lags and additional disjunctive constraints have been proved. This last problem can be formulated as an integer linear program whose linear relaxation can be solved by means of a network flow approach (see also Bianco and Caramia, 2008, 2010). Computational results confirmed a better practical performance of the proposed lower bound with respect to the aforementioned ones.

#### 4 Resource Constrained Project Scheduling with Feeding Precedences

More recently, a new variant of the RCPSP appeared in the literature. It is an extension of the classical Resource Constrained Project Scheduling Problem (RCPSP) in which a further type of precedence constraints denoted as *Feeding Precedences* (FP) is introduced. The first authors introducing this problem variant were Kis et al. (2004).

This problem happens in that production planning environment, like make-to-order manufacturing, which commonly requires the so-called project-oriented approach. In this approach a project consists of tasks each one representing a manufacturing process.

Due to the physical characteristics of these processes, the effort associated with a certain activity for its execution may vary over time. An example is that of the human resources that can be shared among a set of simultaneous activities in proportion variable over time. In this case the amount of work per time unit devoted to each activity, and therefore its duration, are not univocally defined.

This kind of problems is in general modelled by means of the so called Variable Intensity formulation, that is a variant of the Resource Constrained Project Scheduling Problem (see, e.g., Kis, 2005). As the durations of the activities cannot be taken into play, the traditional Finish-to-Start precedence relations, so as the generalized precedence relations, cannot be used any longer, and there is the need for the feeding precedences. Feeding precedences are of four types:

- *Start-to-%Completed (S%C)* between two activities  $(i, j)$ . This constraint imposes that the processed percentage of activity  $j$  successor of  $i$  can be greater than  $0 \leq g_{ij} \leq 1$  only if the execution of  $i$  has already started.

- *%Completed-to-Start (%CS) between two activities (i, j)*. This constraint is used to impose that activity  $j$  successor of  $i$  can be executed only if  $i$  has been processed for at least a fractional amount  $0 \leq q_{ij} \leq 1$ .
- *Finish-to-%Completed (F% $C$ ) constraints between two activities (i, j)*. This constraint imposes that the processed fraction of activity  $j$  successor of  $i$  can be greater than  $0 \leq g_{ij} \leq 1$  only if the execution of  $i$  has been completed.
- *%Completed-to-Finish (%CF) constraints between two activities (i, j)*. This constraint imposes that the execution of activity  $j$  successor of  $i$  can be completed only if the fraction of  $i$  processed is at least  $0 \leq q_{ij} \leq 1$ .

For this problem, Bianco and Caramia (2009b) proposed a lower bound based on a Lagrangian relaxation of the resource constraints. Regarding exact approaches, only two exact algorithms have been presented in the literature. One is the branch and cut algorithm by Kis (2006), with the objective of minimizing the external resource cost; the other is the exact algorithm by Bianco and Caramia (2009c) based on branch and bound rules.

## 5 Conclusions

In this paper we presented the main advanced topics emerging in the scheduling phase of the project management process. Our analysis mainly focused on the evaluation of project scheduling models with emphasis on different kinds of temporal constraints.

## References

1. Ahuja, K., T. Magnanti, J. Orlin. 1993. Network flows, Prentice Hall, New York.
2. Bartusch, M., R.H. Mohring, F.J. Radermacher. 1988. Scheduling Project Networks with Resource Constraints and Time Windows, *Annals of Operations Research* 16, 201–240.
3. Bianco, L., M. Caramia. 2008. A New Approach for the Project Scheduling Problem with Generalized Precedence Relations. *Proceedings of the 11th International Workshop on Project Management and Scheduling*, Istanbul, April 28-30, 2008.
4. Bianco, L., M. Caramia. 2009a. A New Lower Bound for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations, *Computers and Operations Research*, doi:10.1016/j.cor.2009.07.003.
5. Bianco, L., M. Caramia. 2009b. Minimizing the Completion Time of a Project Under Resource Constraints and Feeding Precedence Relations: a Lagrangian Relaxation Based Lower Bound, RR-03.09 - University of Rome “Tor Vergata”, submitted.
6. Bianco, L., M. Caramia. 2009c. An Exact algorithm to minimize the makespan in project scheduling with scarce resources and feeding precedence relations, RR-07.09 - University of Rome “Tor Vergata”, submitted.
7. Bianco, L., M. Caramia. 2010. A New Formulation of the Resource-Unconstrained Project Scheduling Problem with Generalized Precedence Relations to Minimize the Completion Time, *Networks*, in press.

8. Blazewicz, J., J.K. Lenstra, A.H.G. Rinnooy Kan. 1983. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics* 5 (1), 11–24.
9. Clark, W. 1954. *The Gantt Chart*, Pitman, New York.
10. Demeulemeester, E.L., W.S. Herroelen. 1997a. New benchmark results for the resource-constrained project scheduling problem. *Management Science* 43 (11), 1485–1492.
11. Demeulemeester, E.L., W.S. Herroelen. 1997b. A branch-and-bound procedure for the generalized resource-constrained project scheduling problem. *Operations Research* 45, 201–212.
12. Demeulemeester, E.L., W.S. Herroelen. 2002. *Project Scheduling - A Research Handbook*. Kluwer Academic Publishers, Boston.
13. B. De Reyck. 1998. Scheduling projects with generalized precedence relations - Exact and heuristic approaches, Ph.D. Thesis, Department of Applied Economics, Katholieke Universiteit Leuven, Leuven, Belgium.
14. De Reyck, B., W. Herroelen (1998). A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 111 (1), 152–174.
15. Dorndorf, U. 2002. *Project scheduling with time windows*, Physica-Verlag, Heidelberg.
16. Elmaghraby, S.E.E., J. Kamburowski. 1992. The Analysis of Activity Networks under Generalized Precedence Relations (GPRs), *Management Science* 38 (9), 1245–1263.
17. Herroelen, W. 1998. *Projectbeheersing in de Bouw*, Vlaamse Confederatie Bouw, Vlaamse Ingenieurs-Kamer.
18. Kelley, J.E. 1963. The critical path method: Resource planning and scheduling, *Industrial Scheduling*, J.F. Muth and G.L. Thompson (Editors), Prentice Hall, N.J., 347–365.
19. Klein, R., A. Scholl. 1999. Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling. *European Journal of Operational Research* 112, 322–346.
20. Kis, T., G. Erdős, A. Márkus, J. Váncza. 2004. A Project-Oriented Decision Support System for Production Planning in Make-to-Order Manufacturing, *ERCIM news* 58, 66–67.
21. Kis, T. 2005. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities, *Mathematical Programming* 103 (3), 515–539.
22. Kis, T. 2006. Rcps with variable intensity activities and feeding precedence constraints. In *Perspectives in Modern Project Scheduling*, Springer, 105–129.
23. Goldratt, E. 1997. *Critical Chain*. The North River Press, Great Barrington.
24. Lewis, J.P., 1995. *Project Planning, Scheduling and Control. A Hands On Guide to Bringing Projects in on Time and on Budget*. McGraw-Hill, New York.
25. Lumsden, P. 1968. *Line of Balance Method*, Pergamon Press, Exeter.
26. Moder, J.J., C.R., Philips, E.W. Davis. 1983. *Project management with CPM, PERT and precedence diagramming*, Van Nostrand Reinhold Company, Third Edition.
27. K. Neumann, C. Schwindt, J. Zimmerman. 2002. Project scheduling with time windows and scarce resources, *Lecture Notes in Economics and Mathematical Systems* 508, Springer Verlag, Berlin.
28. Radermacher, F.J. 1985. Scheduling of project networks, *Annals of Operations Research* 4, 227–252.
29. Suri, R. 1998. *Quick Response Manufacturing*, Productivity Press.