A MODEL FOR SCHEDULING OF ACTIVITIES AND ALLOCATION OF CONSTRAINED RESOURCES IN PROJECTS

Clarisse da Silva Vieira
Universidade Federal de Minas Gerais – Escola de Engenharia – Departamento de Engenharia Mecânica
Av. Antônio Carlos, 6627, Pampulha – CEP 31270-901 – Belo Horizonte – MG
cosvieira@terra.com.br

Ricardo Luiz Utsch de Freitas Pinto
Universidade Federal de Minas Gerais – Escola de Engenharia – Departamento de Engenharia Mecânica
Av. Antônio Carlos, 6627, Pampulha – CEP 31270-901 – Belo Horizonte – MG
utsch@demec.ufmg.br

Carlos Roberto Venâncio de Carvalho
Universidade Federal de Minas Gerais – Escola de Engenharia – Departamento de Engenharia de Produção
Av. Antônio Carlos, 6627, Pampulha – CEP 31270-901 – Belo Horizonte – MG
carlos@dep.ufmg.br

Abstract. *This paper proposes a linear programming formulation for the Resource Constrained Project Scheduling Problem (RCPSP). Broadly, this problem may be formulated considering all activities processed within their time windows and requiring a constant amount of renewable resources to be executed. The processing times (durations) of the activities and the amounts of resources are given and assumed to be non negative integer valued. Furthermore, precedence relations are defined between the activities. The scheduling problem considered in this paper uses the concept of a disjunctive graph for the Job Shop where the processing time and the amount of resources are continuous variables. The simultaneous allocation of renewable resources is modeled through binary variables. A mathematical artifice is proposed to turn linear the relations between the processing time and amount of resources. The consistence of the proposed model is illustrated with numeric results.*

Keywords: *Scheduling, Resource Constrained, Linear Programming.*

1. INTRODUCTION

The planning and control of large projects are difficult and important problems of modern enterprises. One of the important problems of the projects management is the Resource Constrained Project Scheduling Problem (RCPSP), which involves the scheduling of the activities in the project to minimize its total duration subject to precedence constraints and constant availability constraints on the required set of resources (Demeulemeester and Herroelen, 1996).

For Oguz and Bala (1994), the RCPSP is an important and challenging problem for both practitioners and researchers. It is important because it has a wide variety of application areas such as design of production facilities, installation of computers systems, large scale construction projects, scheduling of radio and television broadcasts and new product development. It is challenging because it is an NP-complete problem (see Garey and Johnson, 1979 and Blazewicz *et al.,* 1983).

A more detailed of the RCPSP is shown in the Section 2 as well as a bibliographical revision of some of the main models proposed in the literature and of the resolution methods. The Section 3 proposes a new formulation for the studied problem. An example of the scheduling problem is modeled and resolved in Section 4. Finally, the Section 5 presents the conclusion.

2. THE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

The RCPSP, in agreement with Brucker *et al.* (1998), may be formulated as follows: given n activities $i = 1, ..., n$ and r renewable resources. A constant amount of R_k units of resource k is available at any time. Activity i must be processed for p_i time units, where preemption is not allowed. During this time period a constant amount of r_{ik} units of resource k is occupied. The values R_k, p_i and r_{ik} are supposed to be non-negative integers. Furthermore, precedence relations are defined between the activities. The objective is to determine starting times S_i for the activities $i = 1, ..., n$, in order to minimize $C_{\max} = \max_{i=1}^{n} C_i$, where $C_i = S_i + pi$ is the completion time of activity i, such that:

- at each time t the total resource demand is less than or equal to the resource availability for each resource type,
- the given precedence constraints are fulfilled.

Several models of Linear Programming are proposed for that problem, as for instance, Artigues *et al.* (2003), Brucker and Knust (2000), Carlier and Néron (2003), Mingozzi *et al.* (1998) and Christofides *et al.* (1987).

The formulation presented by Artigues *et al.* (2003) is an extension of the classic mathematical model of the Job Shop scheduling problem. The difficulty of the RCPSP comes from the resource limitation constraints, preventing some activities requiring the same resource from being scheduled simultaneously. These constraints can be state by defining each resource k as the union of R_k resource units, such that a given resource unit cannot be allocated at the same time to more than one activity. Hence, in any feasible solution, a resource unit allocated to an activity i has to be directly transferred after the completion of i to a unique activity j. However since all the units of the same resource are equivalent, one has only to know the number of units directly transferred from one activity to another. Then, two other decision variables are defined in the problem. One of them is the number of units of resource k directly transferred from an activity i to an activity j, denominated f_{ijk}. The other is a binary variable x_{ij} is equal to 1 if activity j is constrained to start after the completion of activity i, and equal to 0 otherwise.

The models of Brucker and Knust (2000) and Carlier and Néron (2003) consider the amounts of allocated resources known and they admit the preemptions of the activities. Mingozzi *et al.* (1998) relax, also, partially the precedence relations. In the formulation of Brucker and Knust (2000), based on the formulation proposed by Mingozzi *et al.* (1998), the activities can be interrupted and each part of the activity can be executed simultaneously inside of intervals of time I_t, forming subsets X_{jt}. Than, if an activity i belongs to X_{jt}, the processing time of all activities contained in that subset will be x_{jt}. Each interval of time can contain more than a subset X_{jt}, however the execution time of the activities, inside that interval, cannot exceed interval duration. In the model proposed by Carlier and Néron (2003), the activities are processed in consecutive intervals of time $[t_1, t_2]$, $[t_2, t_3]$, . . ., $[t_L, t_{L+1}]$. The processing time of the part of the activity i that is processed within a given time interval $[t_L, t_{L+1}]$ cannot be larger than this interval. The sum of all the executed parts of the activity i should be the same of the total time for processing that activity.

The model proposed by Christofides *et al.* (1987) considers that the processing times and the resource requirements of every activity are known and independent of the moment in which the activity is processed. The same happens with the available total amount of each resource type. The Christofides *et al.* (1987) formulation relaxes the integrality requirements. It can be strengthened adding constraints which are redundant with respect to the original integer problem but not with respect to the relaxed linear problem.

As RCPSP are complex, many authors developed heuristics procedures to solve them, as explicit Valls *et al.* (2003). The oldest studies in that line appeared in the decade of 60 and they were accomplished by Wiest (1967) and Fendly (1968). Since that, many others appeared, as for instance, proposed them by Boctor (1990) and Oguz and Bala (1994). There are also many commercial softwares based on heuristic routines.

Already other works, like Schrage (1971) and Kolisch *et al.* (1995), present the development of exact algorithms to solve the RCPSP. In agreement with Oguz and Bala (1994), Pritsker *et al.* (1969) formulated the first model of mathematical programming model. In the exact procedures, one of the inconveniences is that the variables number of the model grows very quickly with the size of the problem. However, numerous treatments to solve versions of the scheduling problem have been developed by several authors. Among the most competitive, according to Valls *et al.* (2003), are proposed by Brucker *et al.* (1998) and Mingozzi *et al.* (1998). However, according to Klein and Scholl (1999), only problems of small and medium instances, among thirty activities, can be resolved in a satisfactory way.

3. THE PROPOSED LINEAR MODEL

The classic problem studied by the authors, mentioned previously, considers the processing time of each activity of the project and the amounts of necessary resources to execute them as being constant given that assume values non-negative integers. As it verifies Konstantinidis (1998), these considerations are usual in many other works and publications.

On the other hand, in the planning area of the production, problems that integrate the problem of the definition of the sizes of lots into the scheduling problem of operations in machines are known. Some examples of these problems are studied by Carvalho (1998).

In that way, the model proposed, as well as the general cases presented in Dauzère-Pèrés *et al.* (1998), consider the processing time of the activity i as being a continuous variable, in other words, could vary in function of the amount of resource allocated. That new formulation is a generalization of the problem presented by Artigues *et al.* (2003) and consists in the determination of the best processing times of the activities according to the allocation of the available resources.

3.1. Definition of the problem

Let

- $N = \{1, ..., n\}$: set of the activities i of the project,
- $s = 0$ and $t = n + 1$: two dummy activities such that represent, respectively, the beginning and the end of the all activities of the project, such that, the processing time for those activities is constant and equal a zero, $P_s = P_t = 0$,
- $R = \{1, ..., m\}$: set of renewable resources,
- R_k: available maximum amount of the resource $k \in R$,
- $p: N \rightarrow IR_+$: function to represent the processing time P_i of each activity of the project.

The value of P_i is constant for the activities i that need fixed amounts of resources for they be executed. For the activities i that doesn't have those amounts of resources fixened, previously in the project, P_i is calculated in function of the amounts of resources used for the execution of the activity i:

$$P_i \geq p_{ik} \ \forall k \in R, \tag{1}$$

where, p_{ik} is the processing time of the activity i defined in function of a certain amount of resource k. The p_{ik} can be calculated by the equation:

$$p_{ik} = \overline{p}_{ik} - a_{ik} x_{ik}, \tag{2}$$

with,

- $\overline{p}_{ik}$: larger processing time of the activity i, determined when the smaller possible amount $\underline{q}_{ik}$ of the resource k is allocated to execute that activity,
- $a_{ik} \geq 0$: decrease coefficient of the processing time of the activity i, when an unit of the resource k is allocated:

$$a_{ik} = \frac{\overline{p}_{ik} - \underline{p}_{ik}}{\overline{q}_{ik} - \underline{q}_{ik}}, \tag{3}$$

where:

- $\underline{p}_{ik}$: smaller processing time of the activity i, determined when the largest possible amount $\overline{q}_{ik}$ of the resource k is allocated to execute that activity,
- x_{ik}: amount of resource k that will be allocated to accelerate the execution of the activity i.

Over there:

- l_{ik}: maximum total amount of resource k that will be allocate to accelerate the execution of the activity i, or be:

$$l_{ik} = \frac{\overline{p}_{ik} - \underline{p}_{ik}}{a_{ik}}. \tag{4}$$

- $E = \{(i, j) \in NxN\}$: set of pairs of activities to represent the precedence relationships among the activities of the project,
- M_1 e M_2: two arbitrary large integers.

The decision variables are defined as:

- t_i: starting time of activity i,
- x_{ik}: amount of resource k that will be allocated to accelerate the execution of the activity i,
- P_i: processing time of activity i,
- $y_{ij} \in \{0,1\}$ associated to each pair of activities, such that if y_{ij} is equal to 1 if activity j is constrained to start after the completion of activity i, and equal to 0 otherwise,
- f_{ijk}: number of units of resource k directly transferred from an activity i to an activity j.

3.2. Model

The model is:

$$\min \ C_{\max} \tag{5}$$

$$\text{s.t. } y_{ij} = 1 \ \forall (i,j) \in E \tag{6}$$

$$P_i \geq \overline{p}_{ik} - a_{ik} x_{ik} \ \forall \ i \in N, \ \ \forall \ k \in R \tag{7}$$

$$t_j - t_i - P_i - M_1 y_{ij} \geq -M_1 \ \ \forall \ i \in N \cup \{s\}, \ \ \forall \ j \in N \cup \{t\} \tag{8}$$

$$f_{ijk} - M_2 y_{ij} \leq 0 \ \ \forall \ i \in N \cup \{s\}, \ \ \forall \ j \in N \cup \{t\}, \ \ \forall \ k \in R \tag{9}$$

$$\sum_{j \in N \cup \{t\}} f_{sjk} = R_k \;\; \forall\, k \in R \tag{10}$$

$$\sum_{j \in N \cup \{t\}} f_{ijk} = \underline{q}_{ik} + x_{ik} \;\; \forall\, i \in N, \;\; \forall\, k \in R \tag{11}$$

$$\sum_{i \in N \cup \{s\}} f_{ijk} = \underline{q}_{jk} + x_{jk} \;\; \forall\, j \in N, \;\; \forall\, k \in R \tag{12}$$

$$\sum_{i \in N \cup \{s\}} f_{itk} = R_k \;\; \forall\, k \in R \tag{13}$$

$$t_i \geq 0 \;\; \forall\, i \in N \tag{14}$$

$$0 \leq x_{ik} \leq l_{ik} \;\; \forall\, i \in N, \;\; \forall\, k \in R \tag{15}$$

$$f_{ijk} \geq 0 \;\; \forall\, i \in N \cup \{s\}, \;\; \forall\, j \in N \cup \{t\}, \;\; \forall\, k \in R \tag{16}$$

$$y_{ij} \in \{0,1\} \;\; \forall\, (i,j) \in E \tag{17}$$

Equation (5) gives the objective of the scheduling problem. Constraints (6) give the precedence relations within the project. Constraints (7) define the processing time of each activity. Constraints (8) are disjunctive constraints that prevent two activities linked through a resource unit flow from being scheduling simultaneously. Constraints (9) describe the link between variables f_{ijk} and variables y_{ij}, that is the implication $y_{ij} = 0, f_{ijk} = 0,\ k \in R$. Constraints (10) guarantee that the total amount of resource that leaves the activity s is the same to the available total amount of that resource. Constraints (11) and (12) express that the input and output flow of an activity on a resource k must be equal to its required capacity on that resource, i.e. the flow conservation property. Constraints (13) guarantee that the total amount of resource that arrives on the activity t is the same to the available total amount of that resource. Constraints (14), (15), (16) and (17) give the domain of variables.

4. THE EXAMPLE PROBLEM

Consider the example presented in Artigues *et al.* (2003), which consists in a project composed by twelve activities. The activities $s = 0$ and $t = n + 1$ are two dummy activities such that represent, respectively, the beginning and the end of the all activities of the project. The activities are defined for the set $N \cup \{s\} \cup \{t\}$.

The precedence relationships among the activities of the project are defined for the set E of pairs of activities. In that way, if $(i, j) \in E$, the activity i precedes the activity j, in other words, the activity i should be executed before the activity j.

Then the problem is defined by the sets:

- $N \cup \{s\} \cup \{t\}$: = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11},
- E = {(0,1), (0,2), (0,3), (1,4), (2,5), (2,6), (3,5), (4,6), (5,7), (5,8), (5,9), (6,10), (7,10), (8,11), (9,11)}.

The activities of the project, as well as the precedence relationships among the activities, can also be represented by the graph in Fig. 1.

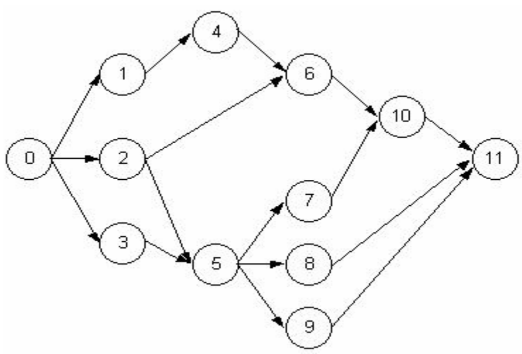

Figure 1. Graph of the example problem

In that graph, the activities are represented by the circles, denominated of node, and the precedence relationships are represented by the arrows, leaving i and arriving in j, that are denominated of arcs.

Generalizing the considered example, the problem data are:

- activities i that doesn't use resources: 0 and 11. For those activities the processing time is constant and equal a zero, $P_s = P_t = 0$,
- activities i that use of some resource, the type of used resource, $k \in R$, the available amounts of each resource type, R_k, the maximum and minimum amounts of used resources, $\overline{q}_{ik}$ and $\underline{q}_{ik}$, and the respective minimum and maximum processing time, $\underline{p}_{ik}$ and $\overline{p}_{ik}$.

  It is possible to obtain other values with these data:
- decrease coefficients of the processing time of activities, a_{ik},
- limits, l_{ik}.

The problem data and the entrances of the model are shown in Tab. 1.

Table 1. Data of example problem.

Activity	Resource	R_k	$\overline{q}_{ik}$	$\underline{p}_{ik}$	$\underline{q}_{ik}$	$\overline{p}_{ik}$	a_{ik}	l_{ik}
0	-	-	0	0	0	0	0	0
1	R_1	4	2	3	1	5	2	1
2	R_1	4	4	4	2	6	1	2
3	R_1	4	3	1	2	2	1	1
4	R_2	3	3	2	1	4	1	2
5	R_2	3	3	1	2	3	2	1
6	R_3	4	4	2	2	4	1	2
7	R_4	5	5	3	3	5	1	2
8	R_4	5	4	2	1	6	1,3	3
9	R_4	5	4	1	1	4	1	3
10	R_5	4	4	2	3	4	2	1
11	-	-	0	0	0	0	0	0

After the computational implementation of that problem, the makespan value of 13 times units is obtained for this instance. Besides, the values of the starting time of activities as well as of the amounts of resources that will be allocated and of the processing times in the optimal solution are shown in Tab. 2.

Table 2. Solution of example problem.

Activity	t_i	x_{ik}	P_i
0	0	0	0
1	0	1	3
2	0	0	6
3	3	0	2
4	3	2	2
5	6	1	1
6	7	2	2
7	7	1	4
8	7	0	6
9	11	2	2
10	11	1	2
11	13	0	0

When the maximum amounts of resources are allocated, as happen with the activities 1, 4, 5, 6 and 10, the values of the processing times are the same to the minimum values, of the respective activities.

Already with the activities 7 and 9, in spite of additional resources be allocated, those amounts of resources don't represent the maximum amounts. Therefore, the processing times, for both activities, don't represent the minimum times.

The solution can also be visualized in the Gantt's diagram shown by Fig. 2.

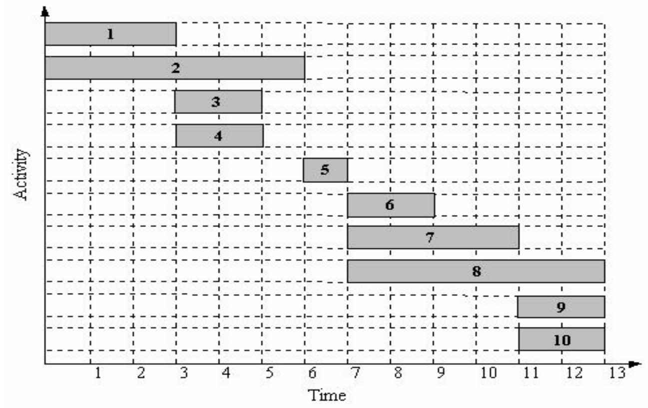

Figure 2. Gantt's diagram of example problem

Notice that the activities that use the same resource type, activities 1, 2 and 3, 4 and 5, 7, 8 and 9, can be execute simultaneously. That is possible since a certain unit of resource given is not allocated, at the same time, for more than an activity and the total amount of the used resource doesn't overtake the value of the available amount of that resource.

Also notice that the activities 3 and 6 are not critical activities in the execution of the project. In that way, the processing times of those activities can be increased, inside of certain limits, without the project duration is altered. In case it happens, for example, a delay in the execution of the project or a reduction of the amounts of available resources, those activities can be affected without modifying the finalization date of the project.

5. CONCLUSIONS

The proposed model was computational implemented and used MPL (Mathematical Programming Language) for the modeling and GLPK (Linear GNU Programming Kit) for the resolution of this problem. In set of problems of small and medium size (among 25 activities and without great difficulties in the allocation of resources), they obtained optimal solutions, besides of the example problem presented here and studied by Artigues *et al.* (2003).

The presented model can be applied in a great number of situations and assists several problems inside of the studied context because it is more generic. That generality is enlarged, mainly, for considering the processing times and the amounts of resources as continuous variables of the problem.

Furthermore, the model treats the problem of allocation of resources as a flow problem. The renewable resource is transferred of an activity for other, so that all of the activities are execute.

Due to the degree of difficulty of the scheduling problems and of the presented model, the research for the realization of future works is necessary. In that way, it is of great interest the implementation of mathematical techniques for the resolution of the others big problems. These techniques can be, for example, the Benders Decomposition method or the inclusion of new constraints in the problem to become it next to the administrative reality.

6. REFERENCES

Artigues, C., Michelon, P. and Reusser, S., 2003, "Insertion Techniques for Static and Dynamic Resource Constrained Project Scheduling", European Journal of Operational Research, , Vol. 149, pp. 249-267.

Blazewicz, J., Lenstra, J. K. and Kan, A. H. G. R., 1983, "Scheduling Projects Subject to Resource Constraints: Classification and Complexity", Discrete Applied Math, Vol. 5, pp. 11-24.

Boctor, F. F., 1990, "Some Efficient Multi-Heuristic Procedures for Resource-Constrained Project Scheduling", European Journal of Operational Research, Vol. 49, pp. 3-13.

Brucker, P. and Knust, S., 2000, "A Linear Programming and Constraint Propagation-Based Lower Bound for the RCPSP", European Journal of Operational Research, Vol. 127, pp. 355-362.

Brucker, P., Knust, S., Schoo, A. and Thiele, O., 1998, "A Branch and Bound Algorithm for the Resource-Constrained Project Scheduling Problem", European Journal of Operational Research, Vol. 107, pp. 272-288.

Carlier, J. and Néron, E. (2003), "On Linear Lower Bounds for the Resource Constrained Project Scheduling Problem", European Journal of Operational Research, Vol. 149, pp. 314-324.

Carvalho, C. R. V., 1998, Une Proposition D'Intégration de la Planification et L'Ordonnancement de Production: Application de la Méthode de Benders, Tese, Universidade Blaise Pascal.

Christofides, N., Alvarez-Valdes, R. and Tamarit, J. M., 1987, "Project Scheduling with Resource Constraints: A Branch and Bound Approach", European Journal of Operational Research, Vol. 29, pp. 262-273.

Dauzère-Pèrés, S., Roux, W. and Lasserre, J. B., 1998, "Multi-Resource Shop Scheduling with Resource Flexibility", European Journal of Operational Research, Vol. 107, pp. 289-305.

Demeulemeester, E. L. and Herroelen, W. S., 1996, "An Efficient Optimal Solution Procedure for the Preemptive Resource-Constrained Project Scheduling Problem", European Journal of Operational Research, Vol. 90, pp. 334-348.

Fendly, L. G., 1968, "Toward the Development of a Complete Multiproject Scheduling System", Journal of Ind. Engineering, Vol. 19, pp. 505-515.

Garey, M. R. and Johnson, D. S., 1979, "Computers and Intractability: A guide to the Theory of NP-Completeness", Ed. W. H. Freeman and Company, New York.

Klein, R. and Scholl, A., 1999, "Computing Lower Bounds by Destructive Improvement: An Application to Resource-Constrained Project Scheduling", European Journal of Operational Research, Vol. 112, pp. 322-346.

Kolisch, R., Sprecher, A. and Drexl, A., 1995, "Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems", Management Science, Vol. 41, pp. 1693-1703.

Konstantinidis, P. D., 1998, "A Model to Optimizad Project Resource Allocation by Construction of a Balanced Histogram", European Journal of Operational Research, Vol. 104, pp. 559-571.

Mingozzi, A., Maniezzo, V., Ricciardelli, S. and Bianco, L., 1998, "An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation", Management Science, Vol. 44, pp. 714-729.

Oguz, O. and Bala, H., 1994, "A Comparative Study of Computational Procedures for the Resource Constrained Project Scheduling Problem", European Journal of Operational Research, Vol. 72, pp. 406-416.

Pritsker, A. B., Watters, L. J. and Wolfe, P. M., 1969, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", Management Science, Vol. 16, pp. 93-108.

Schrage, L., 1971, "Solving Resource-Constrained Network Problems by Implicit Enumeration-Nonpreemptive Case", Operational Research, Vol. 18, pp. 225-235.

Valls, V., Quintanilha, S. and Ballestín, F., 2003, "Resource-Constrained Project Scheduling: A Critical Activity Reordering Heuristic", European Journal of Operational Research, Vol. 149, pp. 282-301.

Wiest, J. D., 1967, "A Heuristic Model for Scheduling Large Projects with Limited Resources", Management Science, Vol. 13, pp. 359-377.

7. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.