THE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM FORMULATION CONSIDERING PROCESSING TIME AND AMOUNT OF RESOURCE CONTINUOS

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) with processing time and amount of resource as continuous variables. The problem is modeled as a resources allocation problem combined with a scheduling activities problem, considering disjunctive graphs for the Job Shop. Moreover, precedence relations between the activities must be respected. The simultaneous allocation of renewable resources is modeled with the aid of binary variables. In this formulation, the processing time is a nonlinear function of the amount of resource allocated. Linear approximations using arbitrary set of line segments are used to reformulate the problem as a mixed linear programming problem. Although the treatment is initially aimed to continuous resources, it presents itself as an effective treatment for problems with discrete resources. The implementation of the proposed procedure is illustrated by a discrete example whose resources are treated in a relaxed way.*

Keywords: *Nonlinear Program, Linear Mixed Program, Resource Constrained, Scheduling.*

1. INTRODUCTION

The Resource Constrained Project Scheduling Problems are classified as literature relevant problem that motivates many authors, such as Artigues *et al.* (2003), Balas (1967), Brucker and Knust (2000), Carlier and Nerón (2003), Christofides *et al.* (1987), Deblaere *et al.* (2007), Klein and Scholl (1999) and Mingozzi *et al.* (1998). However, due to the complexity and difficulty of resolving such problems, many variations of the original formulation can be observed.

In agreement with Brucker *et al.* (1998), scheduling problems are formulated for a set of activities to be executing sharing renewable resources. Precedence relations must be respected, that is to say, some activities necessarily must be performed before others. Moreover, when the project is executed, if a resources are transferred from one activity to another, the first must precede the second, even if not a priori established, an approach introduced by Balas (1967). This concept turns the problem as a flow problem, as considered by other authors as Artigues *et al.*, (2003) and Deblaere *et al.* (2007).

However, despite the particular features presented for each formulation of RCPSP, all these authors consider that the value of the processing time of each activity of the project and the amount of resource used in implementation, are known constants. In that context, only the sequence of realization can be optimized. As noticed by Konstantinidis (1998), that is usual in many other studies and publications.

On the other hand, according to Power *et al.* (2004), in many practical applications, the amount of resources consumed and the processing time of the activities are rarely constants, an approach followed by Vieira *et al.* (2007), where the processing time of each activity is assumed be a nonlinear function of the amount of resource.

In Vieira *et al.* (2007), the nonlinear relation between the processing time and amount of resource is approximated by a single line. However, this procedure generates relevant imprecision often providing solutions quite different from the exact optimal solution.

In this work, it's proposed a linear approximation involving an arbitrary number of line segments, such that it's possible to obtain solutions arbitrarily near to the exact one. A relevant aspect of this approach study is its applicability to problems with discrete resources. In this case, relaxed solutions are obtained. These solutions are computationally easier to be obtained and may be easily adjusted later to meet the question of discrete resource, if necessary.

In Section 2 of this work, the nonlinear formulation of RCPSP is presented. Section 3 contains the developments and all arguments supporting the approach proposed. The usefulness of the procedure in problems with discrete resources is briefly discussed in Section 4. An example of optimal allocation and sequencing problem is computationally modeled and solved in Section 5. Finally, Section 6 presents the main conclusions of the paper.

2. NONLINEAR FORMULATION OF RCPSP

Consider a problem of minimization of the time needed to execute a sequence of *n* activities that need to share the same resource *R*, where the activity processing time varies nonlinearly with the amount of resource applied. In the context of the optimization theory, that is a nonlinear problem of optimal activities sequencing (Job Shop) with optimal allocation of resources.

Mathematically, the problem can be formulated as:

$$\textit{Minimize}\ \ t_{n+1} \tag{1}$$

such that:

$$p_i = \beta_i(q_i) \qquad i = 1,...,n; \tag{2}$$

$$(t_j - t_i - p_i)y_{ij} \geq 0 \qquad i = 0,1,...,n;\ j = 1,..,n+1; \tag{3}$$

$$0 \leq f_{ij} \leq l_{ij} y_{ij} \qquad i = 0,1,...,n\ j = 1,...,n+1; \tag{4}$$

$$\sum_{j=1}^{n+1} f_{0j} = R \tag{5}$$

$$\sum_{j=1}^{n+1} f_{ij} = q_i \qquad i = 1,..,n; \tag{6}$$

$$\sum_{i=0}^{n} f_{ij} = q_j \qquad j = 1,..,n; \tag{7}$$

$$\sum_{i=0}^{n} f_{in+1} = R \tag{8}$$

$$y_{ij} + y_{ji} \leq 1,\ y_{ij} \in \{0,1\}, \quad i = 0,1,...,n+1;\ j = 0,1,...,n+1; \tag{9}$$

$$y_{ij} = 1 \qquad \forall (i,j) \in E \tag{10}$$

$$t_i \geq 0 \qquad i = 0,1,...,n+1; \tag{11}$$

$$\underline{q}_i \leq q_i \leq \overline{q}_i \qquad i = 1,...,n \tag{12}$$

where:

t_i (i=1,…,n) indicates the starting time of the activity with index i;
t_0 denotes the project starting time, coinciding with the end of artificial activity index 0 which precedes all others;
t_{n+1} denotes the project end time representing the starting time of artificial activity with index n+1 which is preceded by all other activities;
q_i (i=1,…,n) indicates the amount of resource used by the activity with index i;
p_i (i=1,…,n) indicates the processing time of the activity with index i;

$$y_{ij} = \begin{cases} 1, & \text{if the } i\text{-th activity precede the } j\text{-th activity;} \\ 0, & \text{otherwise;} \end{cases}$$

f_{ij} (i=1,…,n; j=1,…,n) denotes the amount of resource that will be transferred from the *i-th* activity to the *j-th* activity;
l_{ij} (i=1,…,n; j=1,…,n) denotes the maximum amount of resource the can be transferred from the the *i-th* activity to the *j-th* activity;
n indicates the number of project activities;
R indicates the total available amount of resource;

$E = \{(i, j)$: i=0,1,…,n; j=1,…,n+1, $i \neq j\}$ represents the set of indexes of activity pairs which precedence is previously imposed.

Equation (1) indicates the minimization of the project conclusion date (t_{n+1}). That is to say, the artificial activity (n+1) must be precede all other activities (y_{in+1}=1, i = 0, 1,..., n). Constraints (2) define the processing time of each activity of the project as a nonlinear function of the amount resource allocated. Constraints (3) give the precedence relations between activity i and activity j. They will be active when y_{ij} = 1 and inactive when y_{ij} = 0. Constraints (4) establish limits to the resource between the activity i end the activity j. Notice that y_{ij} = 0 implies f_{ij} = 0. Constraints (5) guarantee the total amount of resource that leaves the starting activity must be the total amount available. Constraints (6) and (7) establish that the amount allocated at an activity must be equal to the input and output flowing through. Constraints (8) guarantee that the total amount of resource that arrives on the final activity must be the total amount of available resource. Constraints (9) prohibit variables y_{ij} and y_{ji} are both equal to one. Constraints (10) give the precedence relations established a priori. Finally, constraints (11) and (12) provide, respectively, the positivity of the start dates of each activity and minimum and maximum limits for the amount of resource.

Notice that problem has nonlinearities in constraints (2), (3) and (4). Constraints (3) and (4) will be handled according to procedures already established in the literature (Artigues *et al*., 2003). Specially, constraints (2) are the focus of this work and will receive an original treatment that will be presented below.

3. MANIPULATING PROCESSING TIME CURVES OF ACTIVITIES

In this article, processing time of an activity (p_i) is supposed inversely proportional to the amount of resource allocated (q_i), as illustrated in Fig. 1, according to the equation:

$$p_i(q_i) = \frac{\alpha_i}{q_i} \qquad q_i \in \left[\underline{q}_i, \overline{q}_i\right], \tag{13}$$

where α_i represents a proportionality factor corresponding to activity i and $\underline{q}_i$ and $\overline{q}_i$ represents the minimum and maximum allowed amount of resource to execute that activity.

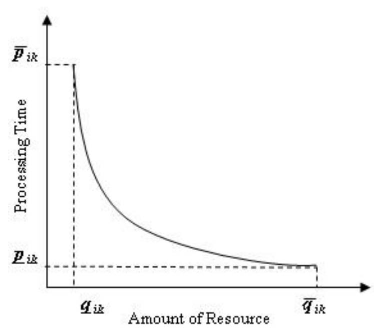

Figure 1. Graph of the a nonlinear function $p_i(q_i)$

3.1. Linear approximation by a line segment

A linear approximation for the function $p_i(q_i)$, adopted in Vieira *et al*., (2007), is to replace the curve by the line segment shown in Fig. 2.

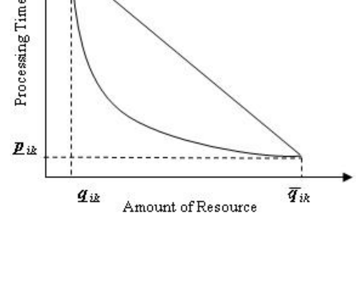

Figure 2. Graph of one segment linear approximation

The line shown in Fig. 2 has the equation:

$$p_i = \bar{p}_i - a_i x_i \tag{14}$$

with:

$$a_i = \frac{\bar{p}_i - \underline{p}_i}{\bar{q}_i - \underline{q}_i}, \tag{15}$$

$$x_i = q_i - \underline{q}_i, \tag{16}$$

where:

$\bar{p}_i$ denotes the processing time for activity i when the amount of resource $\underline{q}_i$ is allocated; and

$\underline{p}_i$ denotes the processing time for activity i when the amount of resource $\bar{q}_i$ is allocated.

Notice that x_i represents the amount of resource allocated for the execution of activity i, beyond the minimum quantity $\underline{q}_i$. In turn, $\bar{p}_i$ e $\underline{p}_i$ represent, respectively, the maximum and the minimum processing time allowed to execute activity i.

As already commented, this approach creates some difficulties, such as a very large discrepancy between the real and the approximate processing time, which can substantially changes the solution obtained.

3.2. Linear approximation by an arbitrary number of line segments

Aiming to improve the results accuracy and to control the approximation errors, this paper proposes an approximation of the nonlinear curves by arbitrary number of line segments, turn possible approximations sufficiently close to exact values. To understand the procedure, consider in Fig. 3 an approach through three line segments: the first one over the interval [$\underline{q}_i$, q_{i1}], the second one over the interval [q_{i1}, q_{i2}] and the third one over the interval [q_{i2}, $\bar{q}_i$].

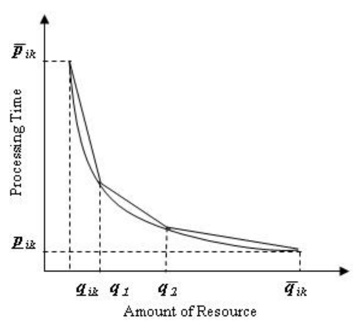

Figure 3. Approximating by three line segments

Notice that the time calculated by linear approximations will be always equal to or greater than the exact value. Thus, the approximate optimal solution is always feasible for the original problem. Notice also that with the approach of Fig. 3, the difference between the real and the approximate processing time is very small if compared with to Fig. 2. Notice also that approximation by three segments is only a didactic example.

To understand the arbitrary segmentation of a nonlinear processing time curve, consider Fig. 4.

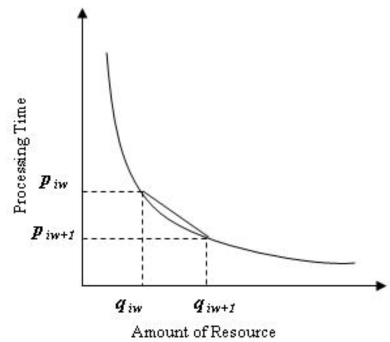

Figure 4. Arbitrary segmentation of a processing time curve

Fig. 4 presents an arbitrary segment approximating the curve on the arbitrary interval $q_i \in [q_{iw}, q_{iw+1}]$.
In this case, the straight segment shown in Fig. 4 has the equation:

$$p_i = p_{iw} - a_{iw} x_i \ , \tag{17}$$

where,

$$a_{iw} = \frac{p_{iw} - p_{iw+1}}{q_{iw} - q_{iw+1}}, \tag{18}$$

$$x_i = q_i - q_{iw}, \quad q_{iw} \leq q_i \leq q_{iw+1} \ . \tag{19}$$

Notice that q_{iw} and q_{iw+1}, respectively, correspond to the lower and higher amount of resource for the considered interval. In turn, p_{iw} and p_{iw+1} correspond to the higher and lower processing time, respectively.

3.3. Linear approximation by multiple segments applied to the RCPSP

To apply the multiple segments approximation to the Resource Constrained Project Scheduling Problem (RCPSP) it is necessary to extend each segment to the entire domain of each activity $q_i \in \left[\underline{q}_i, \overline{q}_i\right]$. To understand the strategy, firstly consider an approximation by only two segments as shown in Fig. 5.

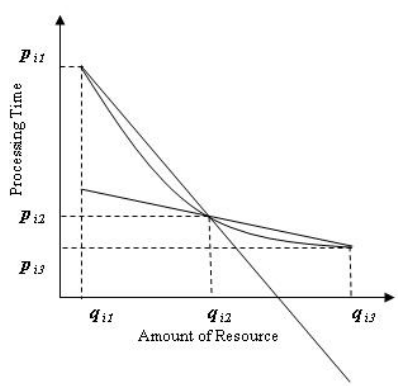

Figure 5. Two segments approximation for a processing time curve

Notice that the support line of each segment is extended over all relevance interval $q_i \in \left[\underline{q}_i, \overline{q}_i\right]$. Thus, the first extended segment has the following mathematical representation:

$$p_i = p_{i1} - a_{i1} x_i \ , \tag{20}$$

where,

$$a_{i1} = \frac{p_{i1} - p_{i2}}{q_{i1} - q_{i2}}, \tag{21}$$

$$x_i = q_i - q_{i1}, \quad \underline{q}_i = q_{i1} \le q_i \le q_{i3} = \overline{q}_i . \tag{22}$$

The second extended segment has the equation:

$$p_i = p_{i2} - a_{i2} x_i \ , \tag{23}$$

where,

$$a_{i1} = \frac{p_{i2} - p_{i3}}{q_{i2} - q_{i3}}, \tag{24}$$

$$x_i = q_i - q_{i2}, \quad \underline{q}_i = q_{i1} \le q_i \le q_{i3} = \overline{q}_i . \tag{25}$$

The idea is to incorporate the two segments approximation in the optimization problem by the inequalities:

$$p_i = p_{i1} - a_{i1} x_i, \qquad 0 \le x_i \le \overline{q}_i - \underline{q}_i , \tag{26}$$

$$p_i = p_{i2} - a_{i2} x_i, \qquad 0 \le x_i \le \overline{q}_i - \underline{q}_i \tag{27}$$

Notice that the extended interval established for both approximation is $\underline{q}_i \le q_i \le \overline{q}_i$. The lower value of p_i satisfying both inequalities will be over the first segment, if $q_i \in [q_{i1}, q_{i2}]$, and will be over the second segment, if $q_i \in [q_{i2}, q_{i3}]$.

It is not difficult to see that, to minimize the time needed to execute the activities, the value of p_i will be the smallest value satisfying both inequalities ((26) and (27)). In other words, in the optimal solution, the value of p_i will be lower bounded by inequality (26), if the optimal q_i belongs to the first interval or by inequality (27), if the optimal q_i belongs to the second interval.

It's not difficult to realize that the approach can be generalized to an arbitrary number of segments by considering the inequalities:

$$\begin{cases} p_i \ge p_{i1} - a_{i1} x_i \\ p_i \ge p_{i2} - a_{i2} x_i \\ \vdots & \qquad 0 \le x_i \le \overline{q}_i - \underline{q}_i \\ p_i \ge p_{iw_i} - a_{iw_i} x_i \end{cases} \tag{28}$$

where w_i denotes the number of segments to approach the processing time of activity i.

4. REWRITING THE RCPSP AS A MIXED LINEAR INTEGER PROGRAM

Applying the multiple segmentation to each activity of the project, we can rewrite the RCPSP as the following mixed linear integer program:

$$\textit{Minimize } \ t_{n+1} \tag{29}$$

such that:

$$\begin{cases} p_i \geq p_{i1} - a_{i1}x_i \\ p_i \geq p_{i2} - a_{i2}x_i \\ \vdots \\ p_i \geq p_{iw_i} - a_{iw_i}x_i \end{cases} \qquad 0 \leq x_i \leq \bar{q}_i - \underline{q}_i \qquad i = 1,\ldots,n \tag{30}$$

$$t_j - t_i - p_i - M_1 y_{ij} \geq -M_1 \qquad i = 0,1,\ldots,n;\ j = 1,..,n+1; \tag{31}$$

$$f_{ij} - l_{ij} y_{ij} \leq 0 \qquad i = 0,1,\ldots,n\ j = 1,\ldots,n+1; \tag{32}$$

$$\sum_{j=1}^{n+1} f_{0j} = R \tag{33}$$

$$\sum_{j=1}^{n+1} f_{ij} = q_i \qquad i = 1,..,n; \tag{34}$$

$$\sum_{i=0}^{n} f_{ij} = q_j \qquad j = 1,..,n; \tag{35}$$

$$\sum_{i=0}^{n} f_{in+1} = R \tag{36}$$

$$y_{ij} + y_{ji} \leq 1,\ y_{ij} \in \{0,1\}, \qquad i = 0,1,\ldots,n+1;\ j = 0,1,\ldots,n+1; \tag{37}$$

$$y_{ij} = 1 \qquad \forall (i,j) \in E \tag{38}$$

$$t_i \geq 0 \qquad i = 0,1,\ldots,n+1; \tag{39}$$

$$f_{ij} \leq 0 \qquad i = 0,1,\ldots,n\ j = 1,\ldots,n+1; \tag{40}$$

where:

x_i (i=1,…,n) denotes the amount of resource beyond the minimum quantity $\underline{q}_i$ to be allocated in the activity i;

w_i (i=1,…,n) denotes the number of segments to approach the processing time of activity i;

M_1 denotes a posit constant with sufficient large value (Artigues *et al.*, 2003);

l_{ij} (i=1,…,n; j=1,…,n) denotes the upper bound for the flow f_{ij}, calculated as $l_{ij} = l_{ji} = \min\{\bar{q}_i, \bar{q}_j\}$;

and t_i, t_0, t_{n+1}, p_i, y_{ij}, f_{ij}, n, R e E are defined as in original formulation (Section 2).

Notice that, according constraints (30), the processing time of all activities is now defined as linear functions of the amount of resource allocated, and the value of optimal p_i will be furnished by the appropriate line segment. Also notice that nonlinear constraints (3) and (4) presented in the original formulation was replaced by linear constraints (31) and (32), according usual manipulations (Artigues *et al.*, 2003).

5. SOLVING PRACTICAL PROBLEMS WITH DISCRET RESOURCE BY RELAXATION

Although the formulation showed above was developed for problems with continuous resources (which may take any real value within the domain interval), the approach can also be used to obtain relaxed solutions for problems with discrete resources (which may take only integer values).

To do this the problem must be modeled considering the processing time of each activity as a continuous function of resources, as made previously for continuous resources. Apparently, it seems interesting to take a line segment for each unit of resource. Thus, it is expected to obtain relaxed solutions that either are naturally integer despite the relaxation, or can migrate to solutions with integer resources through small adjustments made afterwards.

The idea of treating discrete resources as continuous functions can be very interesting in the mixed integer optimization problems, since the relaxed problem will have the number of integer variables substantially smaller. This make easier to obtain computational solutions.

In fact, in the case of scheduling problem with discrete resources, a relaxed version will have only the binary precedence variables (y_{ij}) as integer variables. All others are continuous.

As will be seen in the sequence, an example showed in this paper is a problem with discrete resources for which the relaxed approach applies with success.

6. AN ASSEMBLY EXAMPLE PROBLEM

Consider a project of to assembly a set of five mechanical equipments. Each equipment assembly corresponds to an activity which needs a number of man-days to be performed, as shown in Tab. 1.

Table 1. Management data of example problem.

Activity	Man-Days	Worker Limits
1	20	$1 \leq$ workers ≤ 7
2	25	$2 \leq$ workers ≤ 8
3	25	$3 \leq$ workers ≤ 9
4	30	$4 \leq$ workers ≤ 10
5	15	$5 \leq$ workers ≤ 11

According the specificity of each device, there is a minimum and a maximum limit for the number of men that can work simultaneously over the equipment, as shown in Tab. 1. The assembly must be executed by a company that has a team with ten employees.

Notice that activity “5” needs at least 5 and at most 11 employees to be executed. However, technically, only a maximum of 10 workers could be used, because that is the available number of workers.

Activities "0" and "6" are artificially introduced as shown in the graph of Fig. 4. In this example, the equipments can be assembled in any order, without any precedent relation established a priori. Only for illustration, Fig. 4 shows the five activities being carried out in parallel.

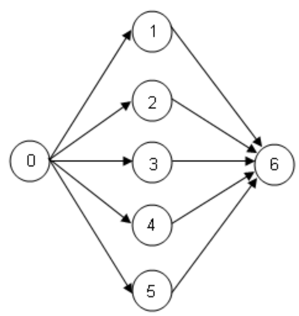

Figure 6. Example of a graph for the five equipment assembly

Maximum ($\overline{p}_i$) and minimum ($\underline{p}_i$) processing times for each activity can be obtained from minimum ($\underline{q}_i$) and maximum ($\overline{q}_i$) amount of resource, as shown in Tab. 2.

Table 2. Technical data for the example problem.

Activity	R	$\underline{q}_i$	$\overline{p}_i$	$\overline{q}_i$	$\underline{p}_i$
1	10	1	20.0	7	2.86
2	10	2	12.5	8	3.13
3	10	3	8.3	9	2.78
4	10	4	7.5	10	3.00
5	10	5	3.0	10	1.50

Notice that, as explained before, for activity “5” the maximum number of employees was established as 10, the available number of workers.

This problem was implemented and solved computationally by using the MPL (Mathematical Programming Language) and GLPK (GNU Linear Programming Kit).

Initially, a first solution was obtained approaching the processing time curves by just one line segment, whose results are presented in Tab. 3.

Table 3. Solution by just one line segment approximation.

Activity	x_i	t_i	p_i	q_i
0	-	0	0	-
1	6	0	2.9	7
2	0	0	12.5	2
3	1	5.0	7.5	4
4	0	5.0	7.5	4
5	3	2.9	2.1	8
6	-	12.5	0	-

A second solution was obtained approach the processing times curves by multiple segments (one segment for each unit interval), as shown in Tab. 4.

Table 4. Solution by multiple segments approximation.

Activity	Segments Number	x_i	t_i	p_i	q_i
0	-	-	0	0	-
1	6	3	0	5	4
2	6	3	6.5	5	5
3	6	2	6.5	5	5
4	6	2	0	5	6
5	5	5	5	1.5	10
6	-	-	11.5	0	-

Both optimal solutions are showed by Gantt's diagram as showed on Fig. 7 and Fig. 8, respectively.

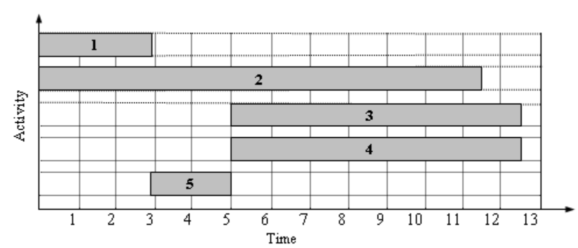

Figure 7. Gantt's diagram for just one line segment approximation

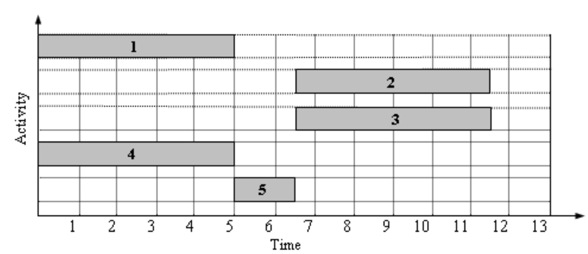

Figure 8. Gantt's diagram for multiple segments approximation

Notice that the sequence of activities for each solution is quite different. In turn, the time to execute them was lower for the second solution (12.5 days for the first one and 11.5 days for the second one).

7. CONCLUSIONS

This paper discussed the solution of a nonlinear Resource Constrained Project Scheduling Problems that considers both the processing time of activities and the amount of resources as continuous variables.

Multiple linear segments approximations were used to express each activity processing time as linear functions of the amount of resource. By using linear inequalities extended over the domain these approaches turn possible to reformulate the original problem as a mixed linear integer program.

Although the procedure has been devised for continuous resources, it can be also applied to obtain relaxed solutions of problems with discrete resources. In this case, the solutions can be easily adjusted afterwards, if necessary, to redeem the discrete character of resources. Relaxation of the problem substantially reduces the number of integer variables, a great advantage from computational point of view.

One example of assembly was solved showing the effectiveness of the multiple segments approach.

The authors hope to present later a generalization of the approach solving problems whose activities make use of multiple resources simultaneously.

8. ACKNOWLEDGEMENTS

The research was supported by CAPES.

9. 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.

Balas, E., 1967, “Project Scheduling with Resource Constrained”, Operational Research, Vol. 15, pp. 915-957.

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.

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.

Deblaere, F., Demeulemeester, E. L., Herroelen, W. S. and Vonder, S. V., 1996, “Robust Resource Allocation Decisions in Resource Constrained Project”, Decision Science, Vol. 38, pp. 5-37.

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.

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”, European Journal of Operational Research, Vol. 153, pp. 239-254.

Poder, E., Beldiceanu, N. and Sanlaville, E., 2004, “Computing a Lower Approximation of the Compulsory Part of a Task with Varying Duration and Varying Resource Consumption”, Management Science, Vol. 16, pp. 93-108.

Vieira, C. S., Pinto, R. L. U. F. and Carvalho, C. R., 2007, “A Model for Scheduling of Activities and Allocation of Constrained Resource in Projects”, Proceedings of the 19th International Congress of Mechanical Engineering, Brasília, Brazil.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.