A Time Efficient Non-recursive Heuristic for Resource Constrained Project Scheduling Problems

Manabendra N Pal*, Tarun Bhaskar†, Ranjan Pal‡

Abstract no.- 002-0045

Second World Conference on POM and 15th Annual POM Conference, Cancun, Mexico, April 30- May 3,2004

Abstract

This paper considers Resource Constrained Project Scheduling Problems (RCPSP) to obtain the minimum make span schedule. A non-recursive heuristic called *Resource Time Ratio Exponent Technique (RETIREXT)* has been developed that evaluates *Schedule Performance Index (SPI)* at every decision point of a project on all possible subsets with maximum number of activities in each under available resource constraints and then selects the one having the maximum value of SPI. The SPI is expressed as a power function with the variables on each subset as the ratio of the required to the available resources of each type respectively and the Maximum Remaining Path Length (MRPL) of each activity on the subset to the duration of the project with no resource constraint. Kolisch and Sprecher

*Professor, Operations Management Group, Indian Institute of Management Calcutta, India. email: mnp@iimcal.ac.in

†Doctoral Student, Indian Institute of Management Calcutta, India. email: tarun@email.iimcal.ac.in

‡Birla Institute of Technology Meshra, Ranchi, India

benchmark problems solutions obtained by RETIREXT are compared with that obtained by other methods and RETIREXT has been found to be the most efficient one for comparable solutions.

The heuristics are used to solve the benchmark problems of Kolisch and Sprecher[17], and the results are compared with the results obtained by other methods. This method gives time effective comparable solutions.

1 Introduction

The classical resource constrained project scheduling problem is one of the most important integer programming problem. Within the classical resource-constrained project scheduling problem, the activities of a project have to be scheduled such that make span of the project is minimized. The planning and control of projects is an important problem that many network planning techniques have tried to handle. Common network planning techniques,such as programme evaluation and review technique (PERT) and critical path method (CPM), concern themselves with the time aspect only. These methods aim to minimize project duration, assuming unlimited availability of resource. When the resources are limited, the objective changes to minimize the project time taking into account the resource available at times.

The traditional term was first coined by [21] and [16] in their books. The solution methodologies for this problem can be sub-divided into three broad categories. The exact methods where we formulate and solve the problem mathematically. The second is some non-exact algorithms and the last is the use of heuristics and meta-heuristics for solving the problem.

The initial phase of the research done in this field was mainly aimed at formulating the problem as 0-1 mathematical programming problem. [5], [25], [24] and

[26] formulated this as 0-1 mathematical programming problem and used some exact methods to solve it. [8] solved the problem using the dynamic programming concept. There were some other methods used to solve this problem. [1] and [3] used complete enumeration technique to solve this problem.

The results were not very encouraging when exact methods were used. The method was not able to give solutions for large problems. So several new algorithms came up to improve the results. [27] used the breadth first technique with branch-and-bound to solve the problem. [22] also used a breadth first technique to solve the problem. This method was very successful in solving the smaller size problem but they were not able to solve some complex problems. Depth first search (DFS) was also used. [10] and [11] used an extension of depth-first branch-and-bound method for the problem. [6] used a novel method to solve the problem. They defined three lower bounds based on which they solved the problem. [18] introduced some new lower bounds and used enumeration technique.

The heuristics and meta-heuristics were also developed to solve this problem. The survey paper [15] gives the overall work in this field by 1972. This paper states that the first heuristic technique was used by [16]. In 1967 [31] developed the first software SPAR-1 (Scheduling Programme for Allocation of Resources) for the scheduling of activities, which was based on his heuristic based on total slack and preferential treatment to critical activities. In the book by [19], the authors introduced the concept of *Maximum Remaining Path Length (MRPL)* for the competing activities. They suggested that if two activities are competing for the same resource, one with the longest remaining series of activities should be given priority. This method tries to complete the longest activity at the earliest so as to minimize the time-overrun.The concept of MRPL gave rise to a new technique which was extended in different ways. [30] proposed an extended version of MRPL concept. he proposed two new algorithms, one concentrated on the time and the other on resource and finally he compiled it to get the final algorithm called "Generes". [12] used the term "resource over time" which is the ratio of resource and activity duration for each activity. The activities are selected on the basis of ROT. [23] proposed a two-level heuristic for activity scheduling. The first phase was independent of resource constraints which was introduced in the second phase. [4] used three different heuristics based on different combinations of the activities that can be scheduled together. He also used different combinations of these heuristics for the solutions. [14] discusses the relative performance of several heuristics suggested for solving RCPSP.

The paper introduces a non-recursive heuristic for solving project scheduling problem under multiple renewable resource constraints (RCPSP). A decision point is defined along the time axis of the progress of the project when any decision may be taken to schedule activities from the remaining ones, at the earliest instance of completion of any activity (s) already scheduled earlier. The decision points are counted by cycle number. The appropriate selection of activities for scheduling at any decision point in order to minimize the makespan is carried out by maximizing the schedule performance index (SPI) over all possible subsets of activities with maximum number in each with the resources available at the decision point. This performance indicator reflects how efficiently for a subset at a decision point, the available resources are utilized and the time overrun of the remaining part of project is reduced. The SPI is expressed as a power function with the variables on each subset as the ratio of the required to the available resources of each type respectively and the Maximum Remaining Path Length (MRPL)[19] of each activity on the subset to the duration of the project with no resource constraint. The exponent of the resource quotient variables have been obtained by taking the respective ratio of the minimum amount of resource required to complete the project without any time overrun to the available resource for the project with no other resource constraint. These exponents are defined as the relative scarcity factors of the respective resources for the overall project. In order to obtain the common exponent of time quotient, first, the minimum project duration time for each type of resource assuming no constraint on other types of resources are calculated respectively, and then dividing each of these calculated minimum project duration time to the project duration time without any resource constraint. These exponents are defined as the relative time overrun factors of the overall project due to scarcity of the respective type of resource. The maximum of all the relative time overrun factors is taken to be the common exponent of time quotient.

The rest of the paper has been structured as follows. In section 2 we explain the proposed heuristic. An example has been solved in section 3 and the advantage of using this method has been shown. In section 4, the computational results and the complexity of the method has been discussed and then we have concluded in section 5.

2 The Proposed Heuristic

The method establishes the relative importance of each type of given resources in terms of bottleneck posed by the same independently in relation to the maximum amount required to complete the project without time overrun. It also inbeds the importance of relatively critical activities at any decision point. The model formulation with the notation used are given as follows:

Notations

j Activity number of the project. j = 1,2 . . . , N.

t_j Duration of activity j of the project

M_j Maximum remaining path length of activity j

r_{jk} Renewable resource of type k required per unit time over the duration of the activity j. k = 1,2 . . . , P.

R_k Total available renewable resource of type k, k=1,2...P.

T Duration of project without any resource constraint

i Cycle number, i.e. the decision point corresponding to time T_i along the time axis of the progress of the project when any decision may be taken to schedule activities from the remaining ones, at the earliest instance of completion of any activity (s) already scheduled earlier.

$\bar{R}_k$ Minimum renewable resources resources of type k required to complete the project in time T with no other (P-1) resource constraints

T_k Duration of the project with minimum time overrun given only the resource constraint of type k with no other (P-1) resource constraint

R_{ik} Renewable resource of type k available at cycle i, k=1,2...P.

C_i Decision set of activities at cycle i, i.e. largest set of activities which can be scheduled with no resource constraint

l_i Number of activities in C_i

c_{im} The subset m of C_i that could be formed taking as many activities as possible without violating any resource constraint at cycle i, m=1,2....K, where, $C_i = \bigcup_{m=1}^{K} c_{im}$, K being all possible subsets formed from C_i

l_{im} Number of activities in c_{im} where $\sum_{m=1}^{K} l_{im} \geq l_i$.

$\frac{\bar{R}_k}{R_k}$ Relative scarcity factor of resource of type k for the overall project.

$\frac{\sum_{j \in c_{im}} r_{jk}}{R_{ik}}$ Relative utilization factor of resource of type k at cycle i for the subset of activities, c_{im}.

$\frac{T_k}{T}$ Relative time overrun factor of the overall project due to scarcity of resource of type k.

Z_{im} Schedule Performance Index (SPI) at cycle i for subset $c_{im}$is defined as.

RETIREXT

$$Z_{im} = \prod_{k=1}^{P} (\frac{\sum_{j \in c_{im}} r_{jk}}{R_{ik}})^{p_k} \cdot [\frac{1}{l_{im}} \cdot \sum_{j \in c_{im}} (\frac{M_j}{T})^q] \quad p_k, q \geq 1. \tag{1}$$

where,

$$p_k = \frac{\bar{R}_k}{R_k}, \quad \text{for} \bar{R}_k > R_k \quad \text{else} \quad p_k = 1$$

$$\text{and} \quad q = \frac{max_k(T_k)}{T}.$$

$\bar{R}_k$ and T_k can be calculated by Brook's [19] algorithm and Burgess and Killebrew algorithm [7] respectively.

For any subset of activities c_{im} at a decision point i, if one or more resource types (but not all) are not required then put the corresponding value of $\frac{\sum_{c_{im}} r_{jk}}{R_{ik}} = \delta$, where $\delta$is a small positive real number say 0.01.

At any decision point i if the decision set activities do not require resources of any type, then all the decision set activities must be scheduled at i.

At cycle i, the subset of activities c_{im} (m=1,2...K), that has maximum value of Z_{im} is selected for scheduling .

3 An Example

We take the following network to check our heuristic. The problem has been shown in the following figure. The mode of description is activity on arc (AOA).

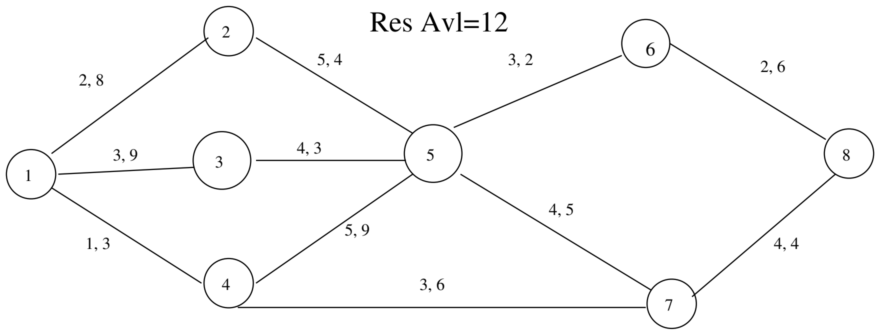

Figure 1: Problem 1

The numbers on the arc shows the duration and the resource requirement respectively. There is only one type of resource. The net availability of resource is 12 units. The critical path length without any resource constraint is 15 units.

Considering the precedence relationship, the activities that can be scheduled in the first cycle are: 12, 13 and 14. But the resource constraints allows either 12 and 14 or 13 and 14 in the first cycle.

Combination 1: 13 and 14

Combination 2: 12 and 14

Combination 3: 12, 13 and 14 (infeasible because of resource constraint)

We now calculate p and q from the given data. Since resource required to complete the project in the time calculated by critical path method is 20 and the time taken by using the MRPL concept is 25 units, therefore the values of p and q are as follows.

$$p = \frac{20}{12} = 1.667$$
$$q = \frac{25}{15} = 1.667$$
$$Z(1) = (12/12)^{1.667}[(1/2)(15/15)^{1}.667 + (14/15)^{1}.667] = 0.945678$$
$$Z(2) = (11/12)^{1.667}[(1/2)(15/15)^{1}.667 + (14/15)^{1}.667] = 0.8668$$

So the first combination is selected and activities 13 and 14 are scheduled at time $= 0$ in cycle 0.

The activity 14 gets over in 1 unit time. So at time = 1, we have 3 unit resource in hand and the activities that can be scheduled are 12, 45 and 47 which cannot be scheduled because of the resource constraint. The activity 13 gets over at time = 3 units and we get 12 unit resource in hand. So the scheduling of next cycle can be done. The options are 12, 45, 35 and 47. The Z values are calculated and the combination of activities 45 and 35 is selected.

Activity 35 gets over at time = 7 and we get 3 units of resource. The activities which can be scheduled now are 12 and 47. But the resource constraint does not allow any activity to be scheduled. The next cycle starts at time = 8. We have 12 units of resource and the activities that can be scheduled are 12 and 47 from which only one can be scheduled and based on Z value we select 12.

In the next cycle (time = 10), the options are 25 and 47 and both of them can be scheduled. Similarly we schedule 56 with 57, only 68 and only 78 in the respective cycles. The net time span is 23 units and time overrun is 8 (23 - 15) units.

If we use the MRPL concept given in [19], the net time span is 25 days and time overrun is 10 units.

We have used our method to solve the problems given in [2], [24], [9] and [30]. We have achieved either the optimal or the best reported solution for the problem.

4 Computational Results and Performance Evaluation

The heuristic was evaluated for the sets of instances generated with PROGEN [17]. The algorithm was coded in C++ language and run on a Linux (Mandrake 9.0) based machine with Pentium III processor and 128 MB RAM. The results are summarized in the table 1 and the comparison with the other methods given in table 2. Wherever the best reported solution is not achieved, the deviation from the best reported solution is represented by $\in$ defined as:

$$\in = \frac{\text{solution using RETIREXT} - \text{best reported solution}}{\text{best reported solution}} \tag{2}$$

The results obtained by using RETIREXT is given in table 1.

Problem Set	# of Problems	=Best Reported Results	average $\in$	time(secs)
30 nodes	480	364	7.8%	0.45
60 nodes	480	291	9.0%	0.60
90 nodes	480	281	12.6%	1.8
120 nodes	600	96	38.2%	5.2

Table 1: Solution of benchmark problems using RETIREXT

Some of the methods suggested in the literature could only solve small problems. [22] solved only the 30 node problems. The number of optimal solutions obtained by using this method is better than some method and worse than some other methods, but the deviation from the best reported solution is not very high and it takes very less amount of time to give this solution. The time taken by other methods compared to ours is shown in table 4.

Problem Set	Average time taken in secs				
	[13]	[20]	[29]	[28]	RETIREXT
30	0.64	na	1.61	0.48	0.45
60	8.89	2.4	2.76	1.49	0.60
90	32.43	7.2	4.63	na	1.8
120	219.86	41	17	2.92	5.2

Table 2: Comparison of time taken by different methods

The results and comparisons show that the solutions do not deviate much from the best reported solution and it takes substantially less amount of time.

This method is very similar to the parallel schedule generation scheme (SGS). The time complexity of the parallel SGS has been proved to be of O(n^2k). In this section we check whether the time complexity of our method is same or different. We calculate it for each step involved in the method.

Now we calculate the time complexity of the method. The different sequential steps involved in this method and the associated time complexities are given below.

- The first step is to find the subset of all the feasible activities. We find all the activities which can be put in the same subset without violating the resource constraint. This is equivalent to finding the critical element in a bin-packing problem, whose complexity has been proved to be of $O(n)$.

- The second step is to calculate the critical time. The complexity of calculating the critical time period without any resource constraint and calculation of MRPL of each activity is $O(n^2)$.

- The time complexity for finding the $max_k(T_k)$, which is of $O(n^2k)$.

- The complexity of calculating $\bar{R}_k$ is also $O(n^2k)$.

The calculation of the value of Z_{im} involves the calculation of two parts and these parts are calculated sequentially. So the net complexity of the method is the maximum of the complexities of all the sequential steps. Therefore, we can say that the time complexity of the method is of the order $O(n^2k)$. Therefore we can say that the method is of polynomial time complexity and thus is useful for solving big problems also.

5 Conclusions

In this paper we have proposed a non-recursive heuristic to solve the multi-resource constrained project scheduling problems. This heuristic is based on a priority rule which selects a group of activities in spite of one particular activity as done by other priority rules. We form subsets of feasible activities taking care of the precedence relationships and resource constraints. This method is based on the concept that if we have several subset of activities to be scheduled at some decision point with resource constraints, the method gives preference to the subset that requires relatively more resource than other subsets, and on the activities having more MRPL. We have also shown that the method proposed by us has polynomial time complexity. For comparable solutions, the heuristic is observed to take less amount of computation time.

References

[1] E. Balas. Project scheduling with resource constraints. In *Applications of mathematical programming techniques*. English University Press, 1971.

[2] Bell and Han. New heuristic solution for solving resource constrained project scheduling problem. *Naval Research Logistics*, 38:315, 1991.

[3] C. E. Bell and K. Park. Solving resource constrained project scheduling problem using a search. *Naval Research Logistics*, 37:61–84, 1990.

[4] F. F. Boctor. Some efficient multi-heuristic procedure for rcpsp. *European Journal of Operational Research*, 49:3–13, 1990.

[5] E. H. Bowman. The schedule sequencing problem. *Operations Research*, 7:621–624, 1959.

[6] P. Brucker, S. Knust, A. Schoo, and O. Thiele. A branch and bound algorithm for resource-constrained project scheduling problem. *European Journal of Operations Research*, 107:272–288, 1998.

[7] A. R. Burgess and J. B. Killebrew. Variation in activity level on a cylindrical arrow diagram. *Journal of Industrial Engineering*, 13(2), 1962.

[8] J. A. Carruthers and A. Battersby. Advances in critical path methods. *Operations Research Quarterly*, 17(4):359–380, 1966.

[9] Davis and Patterson. A new heuristic for resource constraint project scheduling problem. *Management Science*, 21(8):946, 1975.

[10] E. Demeulemeester and W. Herroelen. A branch-and-bound procedure for multiple resource-constrained project scheduling problem. *Management Science*, 38(12):1803–1818, 1992.

[11] E. Demeulemeester, W. Herroelen, and S. Willy. New benchmark results for resource-constrained project scheduling problem. *Management Science*, 43(11):1485–1463, 1997.

[12] E. A. Elsayed. Algorithms for project scheduling with resource constraints. *International Journal of Production Research*, 20(1):95–103, 1982.

[13] K. Fleszar and K. S. Hindi. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, 2003.

[14] S. Hartmann and R. Kolisch. Experimental evaluation of state-of-the art heuristics for the resource constrained project scheduling problem. *European Journal of Operational Research*, 127:394–407, 2000.

[15] W. S. Herrolen. Resource constrained project scheduling- the state of the art. *Operations Research Quarterly*, 23(3):261–275, 1972.

[16] J. Kelley. *Industrial Scheduling.* Prentice Hall, 1963.

[17] R. Kolisch and A. Sprecher. Psplib: A project scheduling problem library. *European Journal of Operations Research*, 96:205–216, 1996.

[18] A. Mingozzi, V. M. S. Riciardelli, and L. Bianco. An exact algorithm for project scheduling and resource constraints based on a new mathematical formulation. *Management Science*, 44(5):714–728, 1998.

[19] J. J. Moder, C. R. Phillips, and E. W. Davis. *Project Management with PERT, CPM and precedence diagramming.* Van Nostrand Reinhold Company, 1983.

[20] R. H. Mohring, A. S. Schulz, F. Stork, and M. Uetz. Solving project scheduling problem by minimum cut computations. *Management Science*, 49(3), 2003.

[21] J. L. Muth and G. L. Thompson. *Industrial Scheduling.* Prentice Hall Englewood Cliffs, 1963.

[22] T. Nazereth, S. Verma, S. Bhattacharya, and A. Bagchi. Multi-resource constrained project scheduling problem: A breadth first approach. *European Journal of Operations Research*, 112:347–366, 1999.

[23] M. I. Norbis and M. Smith. Two-level heuristic for resource constrained scheduling problem. *International Journal of Production Research*, 24(5):1203–1219, 1986.

[24] J. Patterson. Project scheduling. *Naval Research Logistics*, 23(1), 1976.

[25] J. H. Patterson and W. D. Huber. A horizon-varying, zero-one approach to project scheduling. *Management Science*, 20:990–998, 1974.

[26] A. A. B. Pritsker, L. J. Watters, and P. M. Wolfe. Multi-project scheduling with limited resources: A zero-one programming approach. *Management Science*, 16:93–107, 1969.

[27] J. P. Stinson. A branch and bound algorithm for multiple-resource constrained project scheduling problem. *AIIE Transactions*, 80:94–102, 1978.

[28] P. Thomas and A. Lova. A competitive heuristic solution technique for resource constrained project scheduling. *Annals of Operations Research*, 102:65–81, 2001.

[29] V. Valls, S. Quintanilla, and F. Ballastin. Resource constrained project scheduling: A critical activity reordering heuristic. *European Journal of Operational Research*, 149, 2003.

[30] G. E. Whitehouse and J. R. Brown. Generes: An extension of brook's algorithm for project scheduling with resource constraints. *Computers and Industrial Engineering*, 3:261–268, 1979.

[31] J. D. Wiest. A heuristic model for scheduling large projects with limited resource. *Management Science*, 13(6):359–377, 1967.