Indian Management Studies Journal 17 (2013) 77-94

Indian
Management
Studies Journal

Newly Proposed Mathematical Model for Solving RCPSP

Vibha Saihjpal*, S. B. Singh and Rajendra M. Belokar*****

** Department of Mathematics, Punjabi University, Patiala.*

*** Department of Mathematics, Punjabi university, Patiala.*

**** Department of Production Engineering,*

PEC University of Technology, Chandigarh, 160012.

Abstract

Project scheduling has been a subject of wide interest since last few decades. Several algorithms have been proposed to schedule the activities of a project. Scheduling of activities considering the resource constraints along with precedence relations gives rise to a class of problems called the RCPSP. In this paper the authors propose an algorithm to solve RCPSP considering various parameters such as maximum remaining path length, delay in projects due to the activities waiting for resources, resources available versus resources required, priority of the project etc. There occurs a decision point every time when the resources required by the starting activities are more than the existing resources. At every decision point a set of competing activities is formed and the winning activities are allocated the resource. In the proposed algorithm, at each decision point, those activities are also considered whose preceding activities are near completion along with the activities whose predecessors have finished as is done in most of the existing algorithms. This might give a preference to a critical activity whose preceding activity is near completion which otherwise would be ignored at this decision point and hence the project could be delayed. The algorithm takes care of the shifting criticality as the critical path changes with the allocation of conflicting resources.

Key Words

Project scheduling, RCPSP, schedule performance index (SPI), maximum remaining path length (MRPL).

* E-mail : saihjpal.vibha@gmail.com   ** E-mail : sbsingh69@yahoo.com

*** E-mail : rajbelokar@gmail.com

INTRODUCTION

Scheduling forms the basis of any project and hence plays an important role in industries like production, software, construction etc. Scheduling of any project involves decision making at two levels: once to schedule the activities according to precedence relations and then according to the availability of resources. Using CPM a schedule is developed which does not consider any resource constraints, hence needs to be reframed giving rise to resource constraint project scheduling problems. This research aims at developing one such algorithm for allocation of limited resources to competing activities.

RCPSP has become a standard problem in project scheduling and a large research has been done on it. Heuristic methods have been applied to allocate constrained resources. Tarun et al. (2004, 2011) have proposed various methods for the calculation of critical path by allocating constrained resources to competing activities. They have used fuzzy numbers and heuristics, considering a few assumptions as the activities being non pre-emptive, fixed resource requirement and availability, fixed precedence relationship with an objective of minimization of expected project completion time. Considering the uncertainties in a project, they have developed an algorithm that takes care of the uncertainties of the activities which have not been scheduled at a particular point of time. At every decision point, out of the set of competing activities, the subset satisfying resource constraints and having maximum criticality is given the preference. Tarun et al. (2011) also developed a non-recursive heuristic called resource time ratio exponent technique to evaluate SPI for RCPSP to obtain minimum make span schedule. They used it to solve Kolisch and Sprecher benchmark problems and found the results to be better than other methods.

A fundamental assumption in RCPSP is that the activities are non pre-emptive. Francisco Ballest?´n et al. (2008) conclude by the survey on papers dealing with scheduling of the pre-emptive activities that it has little effect on project length when constant resource availability levels are defined. They have developed a generalization of RCPSP where a maximum of one interruption per activity is allowed and can further be applied to more general pre-emptive problems.

Sonke Hartmann et al. (2010) define RCPSP as a basic model which has assumptions that are too restrictive for many practical applications and hence have presented various extensions of the basic RCPSP based on the structure of the problems. They have generalised the concepts of the activity, the precedence relations and the resource constraints and have discussed alternate approaches for scheduling multiple projects.

Jan We.glarz et al. (2011) presented a survey on the single-project problems which are deterministic, single-objective, and multi-mode.

He´di Chtourou et al. (2008) have presented an algorithm for robust resource-constrained project scheduling in two in two phases. In the first phase they have presented an algorithm that solves RCPSP for minimising the make span only using a priority-rule-based heuristic, and the second phase consists of an algorithm solved for maximizing the schedule robustness while considering the make span obtained in the first stage.

There always exists a mismatch between the proposed schedule and actual implementation of the project. Oya I. Tukel et al. (2006) proposed to tackle this by inserting buffers at various stages of the project schedule. Luong Duc Long et al. (2008) developed a fuzzy critical chain method for RCPSP. They added a buffer only at the end of the schedule and updated the schedule according to the extent to which the project buffer is consumed. They calculated the size of project buffer by computations with fuzzy numbers.

Mahdi Mobini et al. (2011) adopted a different approach, they used a meta heuristic algorithm called Artificial Immune Algorithm (AIA), which is inspired by the vertebrate immune system to solve the RCPS problem considering make span minimisation as the objective. Their computational results show that the proposed algorithm has competitive results in comparison with the existing algorithms, hence inspire for solving real-world problems. Another approach using a meta heuristic algorithm is presented by Siamak Baradaran et al. (2010). Considering the resource constrained project scheduling problem as an NP hard model, they used a meta heuristic algorithm namely the hybrid scatter search. They applied the algorithm at each decision point where more than one activity awaits to be operated but resource required is scarce. They used numerical example to explain it.

Pedro Godinho et al. (2012) proposed an adaptive model for multi-mode project scheduling under uncertainty. They assumed that there is a due date for concluding the project and a tardiness penalty for failing to meet this due date and that several distinct modes may be used to undertake each activity. They defined scheduling policies based on a set of thresholds and proposed a procedure based on the electromagnetism heuristic for choosing scheduling policy. They concluded that when the different modes have different characteristics and there is a reasonable difference between the average duration of the project and the due date, the cost advantage of the adaptive policy becomes very significant.

This paper proposes an algorithm for project scheduling under constrained resources considering uncertainty in the estimated activity durations which are assumed to be normally distributed. The proposed algorithm produces an effective schedule by allocating the limited resources at each decision point so as to ensure optimal allocation of resources leading to minimal project duration. It aims at developing a schedule for multiple projects running simultaneously. An algorithm has been developed by Tarun et al.to solve RCPSP based on SPI (Schedule Performance Index). In his algorithm, the activities competing at any decision point includes all those activities whose predecessors have finished. But in the proposed algorithm, even those activities are considered among the competing whose predecessors are about to finish. The 'about to finish' activities include those activities whose LST is less than or equal to the finish time of the activities ready to start. Along with the concept of maximum remaining path length and ratio of resource required and resource available as given by Tarun et al. the proposed algorithm also considers delay in the project due to an activity if it does not get the resource. Another factor added in the expression of SPI in the proposed algorithm is considering the profits fetched by a project. Moreover, the competing subsets are formed based on the resource they require rather than the completion of their predecessors.

Section 2 of the paper presents the notations used in the algorithm. The problem is defined in section 3 and section 4 explains the proposed algorithm. Comparison between the schedule obtained through the algorithm proposed by Tarun et al.(2007) and through the algorithm proposed in this paper, with the help of a hypothetical case study is presented in Section 5. Section 6 gives the discussion of results and future scope.

NOTATIONS

- u_j : Optimistic time required for activity *j*.
- b_j : Pessimistic time required for activity *j*.
- σ_j : The standard deviation of time for activity *j*.
- *j* : Activity number of the project : 0, 1, 2, 3, ..., n + 1 where 0 and n + 1 are dummy start and dummy end nodes respectively.
- t_j : Duration of activity, a random variable.
- *Mj* : Maximum remaining path length of activity *j*.
- *Aj* : Set of activities on the path of maximum length starting from activity *j*.
- A_0 : Set of activities on the critical path.
- S_j : Net standard deviation of the path associated to M_j given by $\left[\sum j \in Aj \sigma j^2\right]^{\frac{1}{2}}$.

r_{jk}	:	Renewable resource of type k required to perform activity j.
R_k	:	Total available resource of type k.
T	:	Project completion time without resource constraints. This is the critical path length of the project based on the expected value of the time taken by each activity.
S	:	Standard deviation of the critical path.
$\overline{R_K}$	:	Minimum resource of type k required to complete the project in time T assuming no other $K - 1$ resource constraints.
T_k	:	Time taken (based on the expected value of the activity time) to complete the project considering only the k^{th} resource constraint and no other resource constraints.
i	:	Number of a decision point. A decision point occurs either at the beginning of the project or when at least one of the running activities is completed.
R_{ik}	:	Resource of type k available at decision point i.
C_i	:	The set of activities which are ready to be scheduled in decision point i. This is the set of activities, which satisfies the precedence relationship.
d	:	Delay in project if the activity is made to wait.
h_g	:	Preference value of the project g.
P_g	:	Profit through the project g.
Z_{im}	:	The value of SPI of activity m in cycle i.
Z_j	:	The value of SPI of activity j.

PROBLEM DESCRIPTION

The algorithm proposed in this paper is for the class of problems with uncertain activity durations and limited resources. The activity durations are estimated based on previous experience and also, at times, on nature of the manager. It depends on the individual how much safety time he adds to the duration of each activity. Since the proposed activity duration may not exactly match the actual time the activity finishes in, lower and upper bound is considered for the duration of each activity. The activity durations are assumed to be normally distributed. The problem under consideration is of time minimization with optimal resource allocation having the following properties :

1. A well-defined set of activities.
2. Random activity duration which is normally distributed and is based

on lower and upper bounds.

3. Fixed precedence relationships.
4. Fixed requirement and availability of resources considering only renewable resources.
5. Pre-emption is not allowed.
6. Expected outcome of projects in terms of profit or attaining more projects is known.

THE PROPOSED ALGORITHM

The problem deals with the optimal allocation of resources to activities at every decision point i.e. wherever a resource constraint occurs. The proposed algorithm allocates the resources according to a priority rule which is followed at every decision point. A decision point occurs at the beginning of the project or when any activity finishes. One or more decision points occur for each limited resource. At every decision point, a set of competing activities is formed which contains the activities requiring a particular resource and whose preceding activities are finished or are about to finish. Among the activities whose preceding activities are about to finish, those activities are considered whose LST is less than the finishing time of the activities ready to start. The priority rule to decide the winning activity for allocation is based on the following parameters :

1. Maximum remaining path length (Moder et al., 1983).
2. Delay in the project due to an activity forced to wait.
3. Ratio of resource requirement to resource availability.
4. Preference given to any project based on its expected profit.

Maximum Remaining Path Length and Delay Factor

The maximum remaining path length, M_j of a particular activity is the length of the longest remaining path starting from that activity. The set of activities on this path is denoted by A_j. To take care of the uncertainty in activity durations, the standard deviation multiplied to n (the weight age given to the standard deviation) of each activity is added to the length of the remaining path of the activity. It is given by $M_j = \Sigma_j \quad A_j t_j + nS_j$. A delay factor d is added to M_j. To give a preference to the activities on the critical path at a decision point, $M_j + d$ is divided by $T + nS$. When d is zero, this ratio is always less than one and is higher for the activities on the critical chain, hence giving them a preference. The parameter n indicates the weightage given to the standard deviation i.e. the uncertainty in activity durations. The ratio is raised to the power q which takes care of the overall criticality of the project. q is measured as the ratio of max T_k to that of T. T_k represents the time taken by the project to finish considering only k^{th} type of resource constraint and no other $(k - 1)^{th}$ resource constraints. It can be calculated using the normal procedure for calculating CPM considering this resource constraint as another precedence relation.

A factor d called the delay factor, added to M_j introduces the delay in the project if the activity in question is forced to wait for the resource. At times, the MRPL for an activity may be longer and hence a higher SPI, but there may exist a large slack in the remaining path and the other competing activity might have a smaller MRPL along with a small slack. Here if the resource is allocated to the activity with smaller slack project may not be delayed at all but if the allocation is done the other way, the project may get delayed because of smaller slack of the waiting activity. This would be applicable for the non-critical paths but it has to be considered because non-critical paths may become critical after resource allocation.

Ratio of Resource Requirement to Resource Availability

Preference is given to the activity requiring larger units of resources. It is measured by the ratio $\frac{r_{jk}}{R_{jk}}$ i.e. ratio of the resource available to the resource required to complete the project in the minimum possible time. This factor is raised to the power P_k which represents the probability of the resource crunch of that type of resource during the whole time span of the project and is given by $P_k = \frac{\overline{R_T}}{R_k}$.

Preference Given to Any Project Based on its Expected Profit

Making money out of the project is the ultimate aim of the project manager hence preference is given to the project which fetches larger profit or which can fetch more projects. For this a factor given by $\frac{h_g p_g}{\sum_1^m h_g p_g}$ is introduced in the expression for SPI. Here m denotes the number of projects running parallel and sharing resources. This ratio gives the weight to the project. This factor would be higher for projects bearing higher weightage since the denominator remains constant.

Hence the final expression for SPI is given by

$$Zj = \left[\left(\frac{d + m_j + nS_j}{T + nS} \right)^q \frac{h_g p_g}{\sum_1^m h_g p_g} \left(\frac{r_{jk}}{R_{ik}} \right)^{P_k} \right] \tag{4.1}$$

The activity with maximum value of is allocated the resource first of all, followed by the one with lesser value of and so on till the resource is completely exhausted.

Remark

The proposed algorithm will be known as Pari RCPSP. The name has been coined in the memory of beloved daughter Late Ms. Pari of second author.

COMPUTATIONAL PROCEDURE

1. Calculate the critical path using CPM and the standard deviation of the critical path using most likely time of the activities.
2. Calculate the values of T_k, R_{ik}, P_k and q.
3. At every decision point, calculate the value of Z_j for all the competing activities.
4. Allocate the resources to the activities in the decreasing order of Z_j.

CASE STUDY

Mr. X owns a production company. He is working on two projects P_1 and P_2 simultaneously. If P_1 finishes in time his company would get multiple projects, although P_2 is a project of another regular client. Hence P_1 is more important for Mr. X. Activity durations, precedence relations and resources to be used in projects P_1 and P_2 are given in tables below.

Table 1

Project 1					Project 2				
Acti-vity	**Duration** a_j	**Duration** b_j	**Succe-ssor**	**Resource required**	**Acti-vity**	**Duration** a_j	**Duration** b_j	**Succe-ssor**	**Resource required**
0	–	–	1,2,3,4,5	–	0	–	–	1,2,3,4	–
1	37	43	6,7	R1	1	33	37	6,7,8	R1
2	32	38	10,11	R2	2	24	26	9	R2
3	8	16	9	R3	3	30	40	5	R1
4	32	44	8	R4	4	27	33	6,7,8	R3
5	8	12	10,11	R5	5	31	39	15	R3
6	14	18	10,11	R5	6	14	26	13,10	R4
7	9	15	9	R4	7	12	16	11,14	R4
8	14	22	10,11	R3	8	12	18	15	R3
9	7	13	13	R5	9	22	38	13	R3
10	13	17	13	R5	10	4	16	12	R5
11	7	17	13	R5	11	9	15	17	R6
12	26	34	13	R4	12	6	10	17	R6
13	–	–	–	–	13	9	11	17	R6
					14	7	13	17	R5
					15	7	11	17	R5
					16	23	27	17	R2
					17	–	–	–	–

Since project 1 would fetch multiple projects to Mr. X, he has fixed the value of h=2 for project P1 and h=1 for project p2 which is any routine project. P1 would gain Mr. X a profit of 10 lakhs and P2 would fetch him 8 lakhs as estimated by his accounts personnel. Mr. X owns one unit each of resources R2, R3, R5 and R6 and two units each of R1 and R4. Assuming the activity durations to be normally distributed, the most likely activity durations and the standard deviations is calculated as given in Table 2.

Table 2

Project 1			Project 2		
Activity	**Duration (t_j) (most likely)**	**Standard Deviation**	**Activity**	**Duration (t_j) (most likely)**	**Standard Deviation**
0	–	–	0	–	–
1	40	1	1	35	0.667
2	35	1	2	25	0.33
3	12	1.33	3	35	1.667
4	38	2	4	30	1
5	10	0.667	5	35	1.33
6	16	0.667	6	20	2
7	12	1	7	14	0.667
8	18	1.33	8	15	1
9	10	1	9	30	2.667
10	15	0.667	10	10	2
11	12	1.667	11	12	1
12	20	1.33	12	8	0.667
13	–		13	10	0.33
			14	15	1
			15	9	0.667
			16	25	2
			17	–	–

Using the most likely activity duration, the EST, LST, EFT, LFT is calculated as given in table 3 and table 4 and the critical paths are found as shown in fig 1 and fig 2 respectively for Project 1 and Project 2.

Table 3

Activity	1	2	3	4	5	6	7	8	9	10	11	12	13
EST	0	0	0	0	0	40	40	38	52	56	56	71	91
EFT	40	35	12	38	10	56	52	56	62	71	68	91	91
LST	0	21	69	0	46	40	40	38	81	56	79	71	91
LFT	40	56	81	38	56	56	52	56	91	71	91	91	91

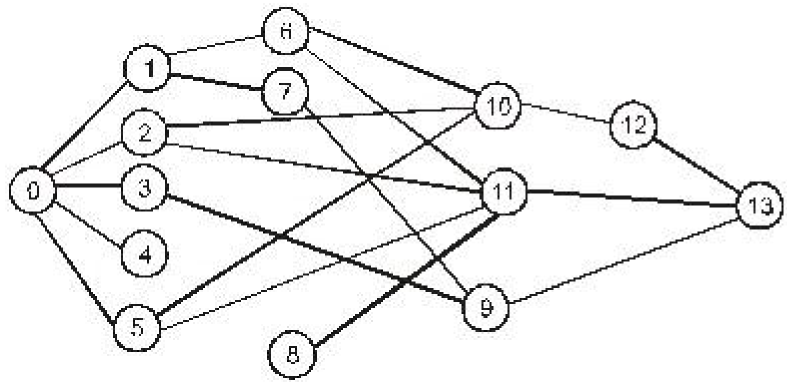

Figure 1 : Network Diagram for Project 1

Table 3

Activity	EST	EFT	LST	LFT
1	0	35	20	55
2	0	25	39	64
3	0	35	0	25
4	0	30	25	55
5	35	70	35	70
6	35	55	66	86
7	35	49	78	92
8	35	50	55	70
9	25	55	64	94
10	55	65	86	96
11	49	61	92	104
12	65	73	96	104
13	55	73	86	104
14	49	59	94	104
15	70	79	70	79
16	79	104	79	104
17	–	–	–	–

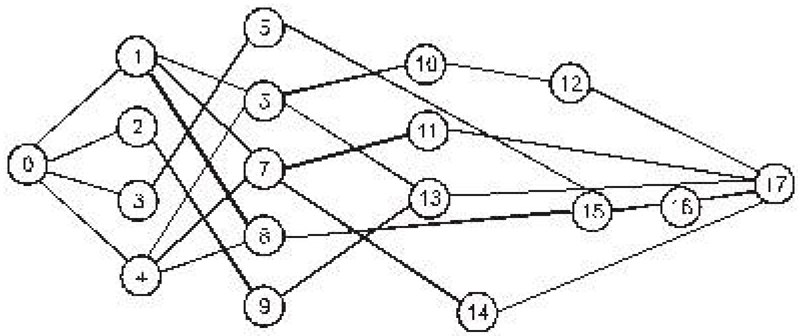

Figure 2 : Network Diagram for Project 2

The schedules obtained as shown in Fig. 1 and Fig. 2 give the basic schedule where the resource availability is not considered. Whereever the availability of resources is constrained, a decision needs to be taken for the allocation of resource. This paper uses two criterions for allocation: one using the proposed algorithm and the other using the algorithm proposed by Tarun et al.which has been modified to develop the proposed algorithm. First we solve the problem of resource allocation using the proposed algorithm.

At the starting point of the project resource R1 is required by A1 of P1, A1 of P2 and A3 of P2. So this is a decision point. To decide the activities for allocation, SPI is calculated for each activity and the one with highest value of SPI gets the resource. The values of SPI as calculated according to equation (4.1), for A1 of P1, A1 of P2 and A3 of P3 are 0.499, 0.0994 and 0.1419 respectively. Hence the winning activities would be A1 of P1 and A3 of P3, as two units of R1 are available. Since A1 of P2 will have to wait till R1 gets free, all the activities succeeding A1 of P2 will also get postponed. Hence the EST, LST, EFT and LFT need to be rescheduled for all those activities. Simultaneously, a decision would be taken for the allotment of R2 to A2 of P1 or A2 of P2, and R3 for A3 of P1 or A4 of P2. Since at the starting time, only A4 of P1 requires R4, it gets allocated to A4. Similarly R5 gets allocated to A5 of P1. As all these allocations are done, the EST, LST, EFT and LFT get rescheduled for the activities which have to wait and all their succeeding activities. The next decision points are identified according to revised values of EST, LST, EFT and LFT. The next decision points and the winning activities are summarised in Table 4. In the column of competing activities, the highlighted activities are the winning activities.

Table 5

Activity Along With Finishing Times

R1 (2 units)	R2 (1 unit)	R3 (1 unit)	R4 (2 units)	R5 (1 unit)	R6 (1 unit)	Activity finished	time	Competing activities
A1P1(40) A3P2(35)	A2P1(35)	A4P2 (30)	A4P1(38)	A5P1(10)	_	_	0	R1 : A1P1,A3P2, A1P2 R2 : A2P1, A2P2 R3 : A4P2, A3P1
A1P1(40) A3P2(35)	A2P1(35)	A4P2 (30)	A4P1(38)			A5P1	10	_
A1P1(40) A3P2(35)	A2P1(35)		A4P1(38)			A4P2	30	R3 : A3P1, A8P1
A1P1(40) A1P2(70)	A2P2(60)		A4P1(38)			A2P1, A3P2	35	R1 : A1P2
A1P1(40) A1P2(70)	A2P2(60)	A8P1 (56)	A7P1(52)			A4P1	38	R4 : A7P1
A1P2(70)	A2P2(60)	A8P1 (56)	A7P1(52)	A6P1(56)		A1P1	40	_
A1P2(70)	A2P2(60)	A8P1 (56)		A6P1(56)		A7P1	52	R4 : A6P2, A7P2, A12P
A1P2(70)	A2P2(60)	A3P1 (68)		A10P1 (71)		A6P1 A8P1	56	R3 : A3P1, A9P2 R5 : A9P1, A10P1, A11P1
A1P2(70)	A5P2(95)	A3P1 (68)		A10P1 (71)		A2P2	60	R2 : A5P2
A1P2(70)	A5P2(95)	A8P2 (85)		A10P1 (71)		A3P1	68	R3 : A8P2, A9P2
	A5P2(95)	A8P2 (85)	A6P2(90)	A10P1 (71)		A1P2	70	_

Contd. Table 5

	A5P2(95)	A8P2 (85)	A6P2(90), A12P1(91)	A11P1 (83)		A10P1	71	R5 : A9P1, A11P1
	A5P2(95)	A8P2 (85)	A6P2(90), A12P1(91)	A9P1(93)		A11P1	83	R5 : A9P1, A14P2
	A5P2(95)	A9P2 (115)	A6P2(90), A12P1(91)	A9P1(93)		A8P2	85	R3 : A9P2
	A5P2(95)	A9P2 (115)	A7P2(104), A12P1(91)	A9P1(93)			90	R4 : A7P2
	A5P2(95)	A9P2 (115)	A7P2(104)	A9P1(93)		A12P1	91	_
	A5P2(95)	A9P2 (115)	A7P2(104)	A10P2 (103)		A9P1	93	R5 : A14P2, A10P2
		A9P2 (115)	A7P2(104)	A10P2 (103)		A5P2	95	R2 : A16P2
		A9P2 (115)	A7P2(104)	A15P2 (114)	A12P2 (111)	A10P2	103	_
		A9P2 (115)		A15P2 (114)	A12P2 (111)	A7P2	104	_
		A9P2 (115)		A15P2 (114)	A11P2 (123)	A12P2	111	_
	A16P2(139)	A9P2 (115)		A14P2 (124)			114	_
	A16P2(139)			A14P2 (124)			115	_
	A16P2(139)						124	_
							139	

Table 6
Activity Along With Finishing Times

R1 (2 units)	R2 (1 unit)	R3 (1 unit)	R4 (2 units)	R5 (1 unit)	R6 (1 unit)	Activity finished	time	Competing activities
A1P1(40) A3P2(35)	A2P1(35)	A4P2 (30)	A4P1(38)	A5P1(10)	-	-	0	R1 : A1P1,A3P2, A1P2 R2 : A2P1, A2P2 R3 : A4P2, A3P1
A1P1(40) A3P2(35)	A2P1(35)	A4P2 (30)	A4P1(38)			A5P1	10	_
A1P1(40) A3P2(35)	A2P1(35)	A3P1 (42)	A4P1(38)			A4P2	30	R3 : A3P1
A1P1(40) A1P2(70)	A5P2(70)	A3P1 (42)	A4P1(38)			A2P1, A3P2	35	R1 : A1P2 R2 : A2P2, A5P2
A1P1(40) A1P2(70)	A5P2(70)	A3P1 (42)	A7P1(52)			A4P1	38	R4 : A7P1
A1P2(70)	A5P2(70)	A3P1 (42)	A7P1(52)	A6P1(56)		A1P1	40	_
A1P2(70)	A5P2(70)	A8P1 (60)	A7P1(52)	A6P1(56)		A3P1	42	R3 : A8P1
A1P2(70)	A5P2(70)	A8P1 (60)		A6P1(56)		A7P1	52	R4 : A6P2, A7P2, A12P1
A1P2(70)	A5P2(70)	A8P1 (60)		A10P1 (75)		A6P1	56	R5 : A9P1, A10P1, A11P1
A1P2(70)	A5P2(70)	A9P2 (90)		A10P1 (75)		A8P1	60	R3 : A9P2
	A2P2(95)	A9P2 (90)	A6P2(90) A7P2(84)	A10P1 (75)		A1P2, A5P2	70	R2 : A2P2

Contd. Table 6

	A2P2(95)	A9P2 (90)	A6P2(90) A7P2(84)	A11P1(87)		A10P1	75	R5 : A11P1, A9P1
	A2P2(95)	A9P2 (90)	A6P2(90), A12P1(104)	A11P1(87)	A11P2 (96)	A7P2	84	R4 : A12P1
	A2P2(95)	A9P2 (90)	A6P2(90)A12 P1(104)	A14P2(97)	A11P2 (96)	A11P1	87	R5 : A9P1, A14P2
	A2P2(95)	A8P2 (105)	A12P1(104)	A14P2(97)	A11P2 (96)	A6P2, A9P2	90	R3 : A8P2
		A8P2 (105)	A12P1(104)	A14P2(97)	A11P2 A9P2	A2P2	95	—
		A8P2 (105)	A12P1(104)	A14P2(97)	A13P2 (104)	A11P2	96	R6 : A13P2
		A8P2 (105)	A12P1(104)	A10P2(107)	A13P2 (104)	A14P2	97	R5 : A9P1, A10P2
		A8P2 (105)		A10P2(107)		A12P1, A13P2	104	
				A10P2(107)		A8P2	105	
				A9P1 (117)	A12P2 (115)	A10P2	107	R5 : A9P1, A15P2
				A9P1 (117)		A12P2	115	
				A15P2 (126)		A9P1	117	R5 : A15P2
	A16P2(151)					A15P2	126	
						A16P2	151	

Next, the scheduling is done using the algorithm given by Tarun et al. The allocation at various decision points is summarised in Table 6. Here SPI is calculated using the expression :

$$Z_{im} = \prod_{k=1}^{k} \left(\frac{\Sigma j \in c_{im} \ r_{jk}}{} \right)^{pk} \cdot \frac{1}{lim} \cdot \Sigma j \in c_{im} \left(\frac{M_j + ns_j}{T + ns} \right)^{q} \quad (5.1)$$

Where c_{im} is the subset m of the set C_i that can be formed taking as many activities as possible without violating any resource constraint, m = 1, 2, 3.....M. l_{im} denotes the number of activities in the subset C_{im} and Z_{in} is the value of SPI of subset *m* in cycle *i*.

RESULTS AND DISCUSSIONS

The schedule obtained in Table 5 finishes the Project 1 at 93 units and project 2 at 139 units while the one in Table 6 finishes Project 1 at 117 units and project 2 at 151 units, hence extending the projects by 25.8% and 8.63% respectively.

For every activity there exists an interval in which the starting time falls, starting from EST to LST. Starting the activity at EST provides a larger slack and hence absorbs some delays but starting the activity at LST would decrease WIP and may lead to resource levelling in some cases. Here lies the future scope to this problem. An optimal starting time would give a more efficient schedule by reducing the WIP time where applicable and increasing the slack, hence reducing the delay caused due to resource constraint where possible.

References

Ballest?´n Francisco a, Vicente Valls b; and Sacramento Quintanilla (2008), Pre-emption in Resource-constrained Project Scheduling, *European Journal of Operational Research,* Vol 189, pp. 1136–1152.

Baradaran Siamak; Ghomi S. M. T. Fatemil; Mobini Mahdi; and Hashemin S. S. (2010), A Hybrid Scatter Search Approach for Resource-constrained Project Scheduling Problem in PERT-type Networks, Volume 41, Issues 7-8, July-August, pp. 966-975.

Bhaskar Tarun; Pal Manabendra; N. Pal; and Asim K. (2011), A Heuristic Method for RCPSP with Fuzzy Activity Times, *European Journal of Operational Research,* Volume 208, Issue 1, 1 January, pp. 57-66.

Bhaskar, Tarun; Manabendra, Nath Pal; Project Scheduling under Uncertainty, (2007), 18th IRMA International Conference, Vancouver, Canada, May 19-23.

Chtourou He´di; and Haouari Mohamed (2008), A Two-stage-priority-rule-based Algorithm for Robust Resource-constrained Project Scheduling, *Computers & Industrial Engineering*, Volume 55, Issue 1, August 2008, pp. 183-194.

Glarz Jan We.; Jozefowska Joanna; Mika Marek; and Waligora Grzegorz (2011), Project Scheduling with Finite or Infinite Number of Activity Processing Modes – A Survey, *European Journal of Operational Research,* Volume 208, Issue 3, 1 February, pp. 177-205.

Hartmann Sonke; and Briskorn Dirk (2010), A Survey of Variants and Extensions of the Resource-constrained Project Scheduling Problem, *European Journal of Operational Research,* Volume 207, Issue 1, 16 November, pp. 1-14.

Long Luong Duc; and Ohsato Ario (2008), Fuzzy Critical Chain Method for Project Scheduling under Resource Constraints and Uncertainty, *International Journal of Project Management*, Volume 26, Issue 6, August, pp. 688-698.

Mobinia Mahdi; Mobinib Zahra; and Rabbania Masoud (2011), An Artificial Immune Algorithm for the Project Scheduling Problem under Resource Constraints, *Applied Soft Computing,* Volume 11, Issue 2, March, pp. 1975-1982.

Moder J. J.; Phillips C. R.; and Davis E. W. (1983), Project Scheduling with PERT, CPM and Precedence Diagramming, Van Nostrand Reinhold Company.

Pal Manabendra N.; Tarun Bhaskar; and Rajan Pal (2004), A Time Efficient Non-recursive Heuristic Resource Constrained Project scheduling problems, Second World Conference on POM and 15th Annual POM Conference, Cancum, Mexico, April 30 to May 3, Abstract No. 002_0045.

Tukel Oya I. (2006), Rom Walter O., Eksioglu Sandra Duni, An investigation of buffer sizing techniques in critical chain scheduling, *European Journal of Operational Research,* Vol. 172, pp. 401–416.