University of Pécs
Faculty of Business and Economy

Criticality in Resource Constrained Projects A New Flexibility Oriented Approach

Roni Levi

Tutor:
Prof. Habil György Csébfalvi CSc

2005

Contents

Abstract

The concept of float and criticality plays a central role in project management. However, the recent literature does not offer a general and useful measure for criticality (flexibility) in resource constrained projects. This study presents a new ***Resource Constrained Total Project Float*** measure (**RCTPF**) to solve this problem. The presented resource constrained project total float measure is defined as the sum of the total floats of activities. In the proposed approach, a resource-constrained project is characterized by its "best" schedule, where best means a schedule in which the RCTPF measure is maximal. Theoretically the optimal schedule searching process is formulated as a mixed integer linear programming (MILP) problem, which can be solved directly for small-scale projects in reasonable time. The presented resource conflict oriented MILP model, which is a reformulation of the traditional time oriented resource constrained project scheduling MILP model, can be used as a new resource constrained project scheduling model in its own right. In the proposed resource usage conflict oriented MILP formulation the number of zero-one variables is usually less than in the traditional time oriented MILP formulation. The presented approach, in according to the NP-hard nature of the problem, introduces both explicit enumeration algorithm which can be solved directly for small-scale projects and implicit enumeration algorithm which provides exact solutions for small to medium size problems in reasonable time. Large-scale problems can be managed by introducing an optimality tolerance which drastically decreases the size of the searching tree. In order to illustrate the essence and viability of the proposed new approach, the subset (PAT16-PAT57) from the well-known and popular

Patterson's set was investigated. To solve the problems a "state-of-the-art" MILP solver (CPLEX 8.1) was used. The presented benchmark problems can be used for testing the quality of exact and heuristic solution procedures to be developed in the future for this problem type.

Key words: project management, resource allocation, resource constrained float, explicit enumeration, implicit enumeration

1. Introduction

Float and the critical path have long been central to the analyses of non-resource constrained projects. However, when resource constraints are introduced a new measure of float is needed. The desirability of such a measure has been noted in reviews project scheduling by Willis (1985) and Ragsdale (1989). Weist (1967) proposed the "critical sequence" as an extension of the critical path. The concept was employed by Bowers (1995) in the development of a set of heuristics for determining resource constrained float. Note that even in a simple resource constrained project, alternative resource allocations are often possible resulting in a choice of schedules with identical project durations but different critical sequences. An activity may be critical in one schedule but have considerable float (flexibility) in another. Raz and Marshall (1996) explored a definition of resource constrained float involving the generation of two different schedules. Bowers (2000) proposed a float definition for multiple alternative resource constrained schedules. However, the recent literature does not offer a general and useful measure for criticality (flexibility) in resource constrained projects. This study presents a new Resource Constrained Total Project Float measure (RCTPF) (as the sum of the total floats of activities, where the total float for an activity is defined as the difference of its latest and earliest start times) to solve the problem. The interested reader is referred to a "state-of-the-art" handbook of scheduling models of Demeulemeester and Herroelen (2002).

The following booklet is organized as follows. In Section 2 the new resource constrained total project float (RCTPF) measure for resource constrained projects is presented. It is shown that theoretically the optimal schedule searching process can be formulated as a mixed integer linear programming (MILP) problem, which may be solved directly for small-scale projects in reasonable time. In Section 3, applied upper bounding technique for the RCTPF measure is described, and a new implicit enumeration algorithm for the proposed RCTPF measure is presented. Section 4 describes the investigation method of the algorithm. The last part, Section 5, lists the most important new results and some issues that call for further investigation.

2. A New Resource Constrained Total Float Measure for Projects - An Explicit Enumeration Algorithm

2.1 The resource constrained total float measure

In order to model our new resource constrained total project float measure (RCTPF) for projects, we consider the following resource constrained project-scheduling problem (RCPSP): A single project consists of N real activities $i \in \{1, 2, \ldots, N\}$ with a nonpreemptable duration of D_i periods. The activities are interrelated by precedence and resource constraints: Precedence constraints - as known from traditional CPM-analysis - force an activity not to be started before all its predecessors are finished. These are given by relations $i \rightarrow j$, where $i \rightarrow j$ means that activity j cannot start before activity i is completed. Furthermore, activity $i = 0 (i = N+1)$ is defined to be the unique dummy source (sink). Naturally, we must have $0 \rightarrow i$ and $i \rightarrow N+1$ for all other activities $i \in \{1, 2, ..., N\}$.

Resource constraints arise as follows: In order to be processed, activity i requires R_{ir} units of resource type $r \in \{1, ..., R\}$ during every period of its duration. Since resource r, $r \in \{1, ..., R\}$, is only available with the constant period availability of R_r units for each period, activities might not be scheduled at their earliest (network-feasible) start time but later.

Let T denote the project's makespan and let $T+1$ denote the start time of the unique dummy sink. Let $\overline{T}$ denote an upper bound on the project's makespan $(T \leq \overline{T})$.

The traditional RCPSP approach minimizes the starting time of the unique sink and thus the makespan of the project. In this study, without loss of generality, we assume that makespan T is the resource constrained minimal makespan and fix the position of the unique dummy sink in period $T+1$.

Let X_i, where $\underline{X}_i \leq X_i \leq \overline{X}_i$, denote the start time of activity i, for $i \in \{1, ..., N\}$. (Note $\underline{X}_i (\overline{X}_i)$ denotes the earliest (latest) starting time of activity i). Because preemption is not allowed, the ordered set $X = \{X_1, ..., X_N\}$ defines a schedule of the project.

Let X_{is}, where $\underline{X}_i \leq s \leq \overline{X}_i$, denote a zero-one decision variable:

$$X_{is} = \begin{cases} 1 & \text{if activity i is started in period s} \\ 0 & \text{otherwise} \end{cases}, \; i \in \{1, ..., N\}. \quad (1)$$

According to the applied notation:

$$X_i = \sum_s s * X_{is}, \; \sum_s X_{is} = 1, \; i \in \{1, ..., N\}. \quad (2)$$

Let $PS = \{i \to j \mid i \neq j, i \in \{1,\dots,N\}, j \in \{1,\dots,N\}\}$ denote the set of predecessor-successor relations. A schedule is network feasible if satisfies the predecessor-successor relations:

$$X_i + D_i \leq X_j, \text{ if } i \to j \in PS. \tag{3}$$

For a network feasible schedule X, let $A_t = \{i \mid X_i \leq t < X_i + D_i\}, t \in \{1,\dots,T\}$ denote the set of active (working) activities in period t and let

$$U_{tr} = \sum_{i \in A_t} r_{ir}, \ t \in \{1,\dots,T\}, \ r \in \{1,\dots,R\} \tag{4}$$

be the amount of resource r used in period t.

A network feasible schedule X is resource feasible if satisfies the resource constraints:

$$U_{tr} \leq R_r, \ t \in \{1,\dots,T\}, \ r \in \{1,\dots,R\}. \tag{5}$$

The objective of our approach is to find a resource feasible schedule, in which the total float (the scheduling flexibility) is maximal:

$$\text{maximize } RCTPF = \sum_{i=1}^{N} \overline{X}_i - \underline{X}_i \tag{6}$$

Easy to see that the presented performance measure RCTPF is irregular, therefore we cannot apply the usual modeling practices and tricks to formulate the model. We must rethink everything from the beginnings. We will show that the result of the revision will be a new MILP formulation of RCPSP.

2.2 A mixed integer linear programming formulation

The presented MILP formulation is based on the forbidden (resource constraint violating) set concept. A forbidden set F of activities is identified such that: (1) all activities in the set may be executed concurrently, (2) the usage of some resource by these activities exceeds the resource availability, and (3) the set does not contain another forbidden set as a proper subset. See, for example, Bell and Park (1990).

A resource conflict can be repaired explicitly by inserting a network feasible precedence relation between two forbidden set members, which will guarantee that not all members of forbidden set can be executed concurrently. An inserted explicit conflict repairing relation (as its side effect) might be able to repair one or more other conflicts implicitly, at the same time.

Let $i \Rightarrow j$ denote that activity j is a direct or indirect successor of activity i.

An $i \rightarrow j$ explicit repairing relation might be replace with a $p \rightarrow q$ relation, where $i \Rightarrow p$ and $q \Rightarrow j$, $i \neq p \vee q \neq j$, and there is a forbidden set in which $p \rightarrow q$ is an explicit repairing relation.

Let $Pred(i)$ denote the set of immediate (direct) predecessors of activity i and let us denote with $ExpRel(F)(ImpRel(F))$ the set of implicit (explicit) repairing relations of forbidden set F.

The central notion of this approach will be illustrated using the following simple example with six activities (Figure 2.1 and 2.2): Consider a simple RCPSP example with six activities. The activities are numbered 1 through 6 (plus the dummy activities 0 and 7). There are two resource types. Only one unit is available from every resource type. The minimal resource feasible makespan is $T = 9$. The total number of network feasible schedules is $|\Re| = 2520$. The total float of the project is 26, which is an upper bound of the resource feasible total project float RCTPF. The presented earliest CPM schedule in not resource feasible. There is over-utilization in period 1(2). Figure2.2 and Table 2.1 illustrate the essence of the example problem. The activities are represented by bars, and the network relations by lines. The unique dummy source (sink) is represented by the >(<) symbol. Let $E = \{0 \to 1, 0 \to 2, 2 \to 3, 0 \to 4, 1 \to 5, 3 \to 6, 4 \to 7, 5 \to 7, 6 \to 7\}$ denote the set of network relations.

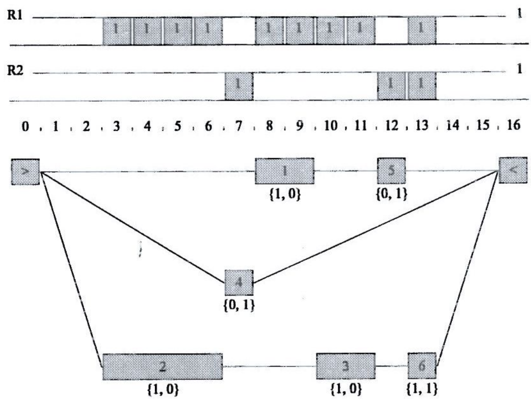


Figure 2.1 A simple project with six activities and two resources

$|\Re| = 2520$

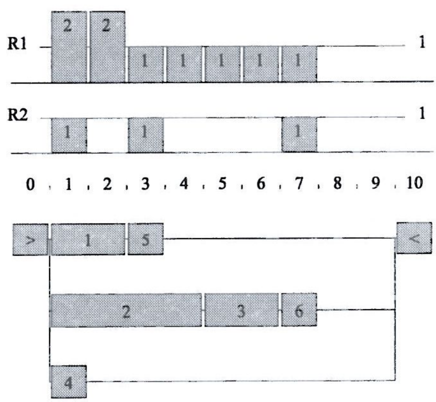


Figure 2.2 A solution for a simple project with six activities and two resources

Table 2.1 A simple project: durations, earliest (latest) start times, immediate predecessors, resource availabilities

i	D_i	$\underline{X}_i$	$\overline{X}_i$	R_1	R_2	Pred(i)
0	1	0	0			
1	2	1	7	1	0	{0}
2	4	1	3	1	0	{0}
3	2	5	7	1	0	{2}
4	1	1	9	0	1	{0}
5	1	3	9	0	1	{1}
6	1	7	9	1	1	{3}
7	1	10	10			{4, 5, 6}
				1	1	

Table 2.2 shows the forbidden sets and their explicit (implicit) repairing sets in the presented earliest CPM schedule. In this schedule every conflict is feasible. A feasible conflict may be "visible" or "hidden". Note that a hidden conflict is "invisible" in the earliest CPM schedule, but might be visible in a shifted schedule. In the earliest schedule only the last conflict is visible.

The proposed RCTPF measure an irregular measure of performance, therefore we must replace the traditional "visible conflict" oriented approach by a "feasible conflict" oriented one. In other words, we have to repair every feasible resource usage conflict regardless of whether it is "visible" or "hidden".

Let us denote with Y the set of different conflict repairing relations and characterize a resource feasible solution by the set inserted explicit repairs, which is a subset of Y. In our simple problem:

$$\mathrm{Y} = \{1\to2, 1\to3, 1\to6, 2\to1, 3\to1, 4\to5, 4\to6, 5\to4, 5\to6, 6\to4, 6\to5\}, \qquad (7)$$

and the total number of the repairing relations is $|\mathrm{Y}| = 11$.

Table 2.2 Forbidden sets and their explicit and implicit repairing relations in the earliest schedule

i	F_i	Status_i	$\text{ExpRel}(F_i)$	$\text{ImpRel}(F_i)$
1	$\{4,5\}$	Feasible - Hidden	$\{4\to5, 5\to4\}$	
2	$\{5,6\}$	Feasible - Hidden	$\{5\to6, 6\to5\}$	
3	$\{4,6\}$	Feasible - Hidden	$\{4\to6, 6\to4\}$	
4	$\{1,3\}$	Feasible - Hidden	$\{1\to3, 3\to1\}$	$\{1\to2\}$
5	$\{1,6\}$	Feasible - Hidden	$\{1\to6\}$	$\{1\to2, 1\to3, 5\to6\}$
6	$\{1,2\}$	Feasible - Visible	$\{1\to2, 2\to1\}$	$\{3\to1\}$

This simple project has four - feasible conflict free - solutions (solution sets), which are shown in Figure2.3. In the solution sets every movable activity can be shifted without affecting the resource feasibility. Easy to realize that from managerial point of view solution 1 is the "best" schedule, because in this solution the resource constrained total float (the scheduling flexibility) is maximal. In solution 4 every activity is critical (unmovable), so an activity delay will delay the completion of the entire project. The relation set, which is describe a solution set, a non-redundant subset of the union of the original network relations and the additional conflict repairing relations.

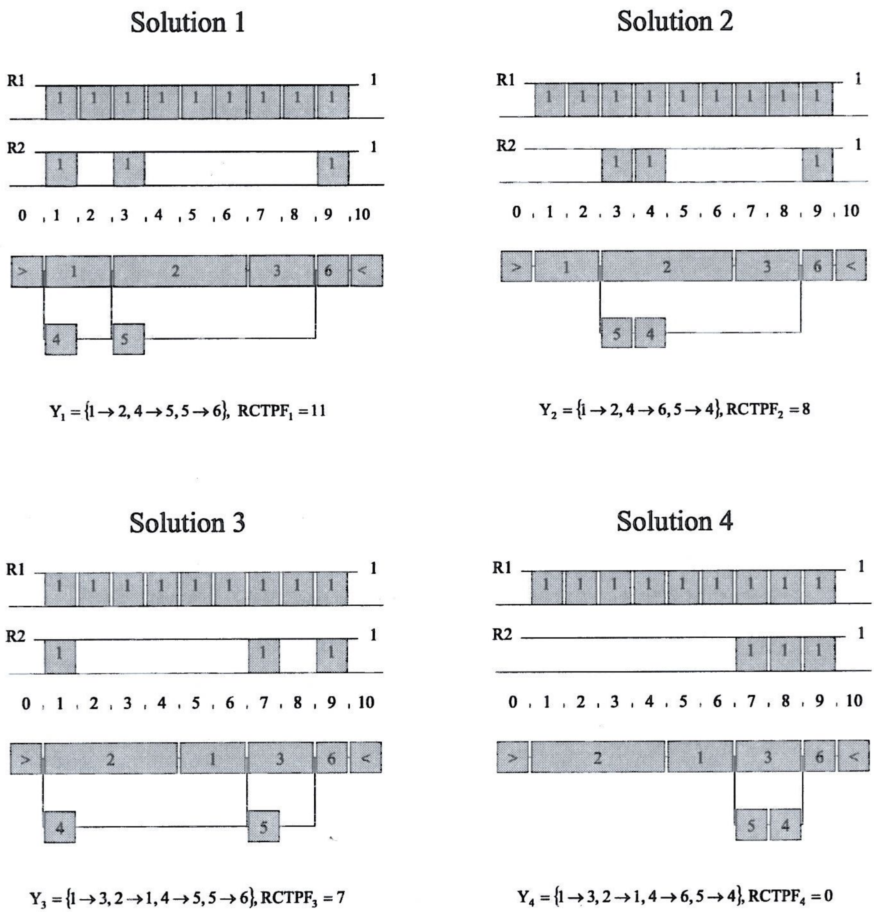


Figure 2.3 Resource feasible solutions

According to the results of our simple example, easy to realize that the proposed RCTPF is a function of the conflict repairing relations:

$$\mathrm{RCTPF} = \mathrm{RCTPF}(Y). \qquad (8)$$

Therefore, we must reformulate the traditional MILP description of RCPSP, according to this fact.. In the proposed model the total number of zero-one variables is $|Y|$, and the formulation is based on well-known "big-M" constraints.

The decision variables are defined as follows:

$$Y_{i\,j} = \begin{cases} 1 & \text{if } i \to j \text{ inserted} \\ 0 & \text{otherwise} \end{cases} \text{, where } i \to j \in Y, \tag{9}$$

The following MILP model arises:

$$\text{Maximize } RCTPF = \sum_{i=1}^{N} \left(\overline{X}_i - \underline{X}_i\right) \tag{10}$$

subject to:

$$\sum_{i \to j \in S_f} Y_{i,j} \geq 1, \text{ where } S_f = \text{ExpRel}(F_f) \cup \text{ImpRel}(F_f), \text{ for every } f \in \{1, \ldots, |F|\} \tag{11}$$

$$\underline{X}_i + D_i \leq \underline{X}_j + (ES_i - LS_j + D_i) \times (1 - Y_{i,j}), \text{for every } i \to j \in Y, \tag{12}$$

$$\overline{X}_i + D_i \leq \overline{X}_j + (ES_i - LS_j + D_i) \times (1 - Y_{i,j}), \text{ for every } i \to j \in Y \tag{13}$$

$$\underline{X}_i + D_i \leq \underline{X}_j, \text{ for every } i \to j \in E, \tag{14}$$

$$\overline{X}_i + D_i \leq \overline{X}_j, \text{ for every } i \to j \in E, \tag{15}$$

$$Y_{i,j} \in \{0,1\}, \text{ for every } i \to j \in Y. \tag{16}$$

The objective function (10) maximizes the proposed resource constrained total project float measure. Constraint set (11) assures the resource feasibility (each resource conflict must be repaired explicitly or implicitly, therefore at least one element must be chosen from each conflict repairing set).Constraint sets (12) and (13) take into consideration the conflict repairing precedence relations (these are the "big-M" constraints in the formulation, where the "big-M" constants are as tight as possible).

The MILP formulation of our simple problem (according to the "state-of-the-art" CPLEX (MPL) solver syntactical requirements) is presented in Table2.3.

Table 2.3 The MILP formulation of our simple problem (CPLEX

```
MAXIMIZE L1-E1+L2-E2+L3-E3+L4-E4+L5-E5+L6-E6;

SUBJECT TO:

P4S5+P5S4>=1; P5S6+P6S5>=1; P4S6+P6S4>=1; P1S3+P3S1+P1S2>=1;
P5S6+P1S3+P1S6+P1S2>=1; P3S1+P1S2+P2S1>=1;

10P4S5+E4-E5<=9; 10P5S4-E4+E5<=9; 10P5S6+E5-E6<=9; 10P6S5-
E5+E6<=9; 10P4S6+E4-E6<=9;
10P6S4-E4+E6<=9;   9P1S3+E1-E3<=7;   9P3S1-E1+E3<=7;   9P1S6+E1-
E6<=7;  9P1S2+E1-E2<=7;
7P2S1-E1+E2<=3;
10P4S5+L4-L5<=9; 10P5S4-L4+L5<=9; 10P5S6+L5-L6<=9; 10P6S5-
L5+L6<=9; 10P4S6+L4-L6<=9;
10P6S4-L4+L6<=9;   9P1S3+L1-L3<=7;   9P3S1-L1+L3<=7;   9P1S6+L1-
L6<=7;  9P1S2+L1-L2<=7;
7P2S1-L1+L2<=3;
E1>=1; E2>=1; E4>=1; -E1+E5>=2; -E2+E3>=4; -E3+E6>=2; -E4+DS>=1; -
E5+DS>=1; -E6+DS>=1;
L1>=1; L2>=1; L4>=1; -L1+L5>=2; -L2+L3>=4; -L3+L6>=2; -L4+DS>=1; -
L5+DS>=1; -L6+DS>=1;
DS=10;

BINARY P4S5 P5S4 P5S6 P6S5 P4S6 P6S4 P1S3 P3S1 P1S6 P1S2 P2S1;
```

Legend: $\{\underline{X}_i \Leftrightarrow E\&i,\ \overline{X}_i \Leftrightarrow L\&i,\ Y_{ij} \Leftrightarrow P\&i\&S\&i,$ Dummy Sink $\Leftrightarrow$ DS$\}$

3. A New Resource Constrained Total Float Measure for Projects problem-An Implicit Enumeration Algorithm

3.1 An upper bounding technique for the RCTPF measure

The resource constrained project-scheduling problem (RCPSP) which was defined in section 2 is considered here again. Let S_i for $i \in \{0,1,\dots,N,N+1\}$ denote the starting time of activity i. Because preemption is not allowed, the ordered set $S=\{S_1,\dots,S_N\}$ defines a schedule of the project. Note that $\underline{S}_i\left(\overline{S}_i\right)$ denotes the earliest (latest) starting time of activity i considering both precedence relations and resource constraints.

Let $PS=\{i\rightarrow j \mid i\neq j, i\in\{0,1,\dots,N\}, j\in\{0,1,\dots,N\}\}$ denote the set of predecessor-successor relations, where $i\rightarrow j$ means that activity j cannot start before activity i is completed.

A schedule is precedence (network) feasible if it satisfies the predecessor-successor relations:

$$S_i + D_i \leq S_j, \text{ if } i\rightarrow j \in PS. \qquad (1)$$

The objective of the traditional resource-constrained project scheduling problem (RCPSP) is to schedule the activities so that precedence and resource constraints are obeyed and the makespan of the project is minimized. Let $ES_i(LS_i)$ for $i\in\{1,\dots,N\}$ denote the precedence but not necessarily resource feasible earliest (latest) starting time of activity i, according to the fixed makespan.

The proposed upper bounding technique is based on the forbidden set concept that was already mentioned. A resource conflict can be repaired by inserting a network feasible precedence relation between two forbidden set members, which will guarantee that not all members of forbidden set can be executed concurrently.

Let $FS=\{FS_f=\{F_{fi}\,|\,i\in\{1,2,...,|FS_f|\},F_{fi}\in\{1,2,...,N\}\}\,|\,f\in\{1,2,...,|FS|\}\}$ denote the set of the feasible forbidden sets of the project. Our objective is to repair every feasible resource conflict such that the proposed RCTPF measure is maximized.

In order to develop an appropriate upper bounding technique,

let S_{it}, where $ES_i \le t \le LS_i$, denote a zero-one decision variable:

$$S_{it}=\begin{cases}1 & \text{if activity i is started in period t}\\ 0 & \text{otherwise}\end{cases},\ i\in\{1,...,N\}. \tag{2}$$

According to the applied notation:

$$S_i=\sum_{t=ES_i}^{LS_i} t*S_{it},\ \sum_{t=ES_i}^{LS_i} S_{it}=1,\ i\in\{1,...,N\}. \tag{3}$$

The traditional zero-one RCPSP model is the following:

$$\text{Minimize (Maximize)}\left[\sum_{i=1}^{N} S_i\right]=\underline{S}\ (\bar{S}) \tag{4}$$

subject to:

$$\sum_{t=ES_i}^{LS_i} S_{it}=1, i\in\{1,2,...,N\} \tag{5}$$

$$S_i=\sum_{t=ES_i}^{LS_i} t*S_{it}, i\in\{1,2,...,N\} \tag{6}$$

$$\sum_{t=ES_j}^{LS_j} t*S_{jt} \ge \sum_{t=ES_i}^{LS_i} t*S_{it}+D_i, i\rightarrow j\in PS \tag{7}$$

$$\sum_{i=1}^{N} R_{ir} * \sum_{s=t-D_i+1}^{t} S_{is} \le R_r, r\in\{1,2,...,R\}, t\in\{1,2,...,T\} \tag{8}$$

$$S_{it}\in\{0,1\},\ i\in\{1,2,...,N\}, t\in\{ES_i,...,LS_i\} \tag{9}$$

The objective function (4) minimizes (maximizes) the starting time of the activities. Constraint set (5) assures that to each activity a unique start time within its time window is assigned. Constraints (6) describe the relation between the integer and binary start time variables. Constraints (7) take into consideration the precedence relations between each pair of activities $i \to j$, where i immediately precedes j. Finally, constraint set (8) limits the total resource usage within each period to the available amount. We can replace the traditional precedence constraint set (7) with a totally unimodular formulation:

$$\sum_{s=t}^{LS_i} S_{is} + \sum_{s=ES_j}^{t+D_i-1} S_{js} \leq 1, i \to j \in PS, t \in \{ES_j - D_i + 1, \ldots, LS_i\} \tag{10}$$

Constraint set (10) assures that activity j must not be begun before time $t + D_i$ if activity i is started at time t or later. Note that the LP relaxation (4)-(6), (8), and (10) is stronger then (4)-(8). See, for example, Demeulemeester and Herroelen. [5]. We will now show that the traditional resource constraint set (8) can be replaced by a new forbidden set oriented formulation. Let $\underline{P}_f$ ($\overline{P}_f$) denote the first (last) time period in which forbidden set FS_f, $f \in \{1, 2, \ldots, |FS|\}$ may be active:

$$\begin{aligned} \underline{P}_f &= \max\{ES_i \mid i \in FS_f\} \\ \overline{P}_f &= \min\{LS_i + D_i - 1 \mid i \in FS_f\} \end{aligned} \tag{11}$$

The forbidden set oriented formulation can be described as follows:

$$\sum_{i \in FS_f} \sum_{s=t-D_i+1}^{t} S_{is} \leq |FS_f| - 1, r \in \{1, 2, \ldots, R\}, t \in \{\underline{P}_f, \ldots, \overline{P}_f\}, f \in \{1, 2, \ldots, |FS|\} \tag{12}$$

Constraint (12), according to the definition, simply describe the fact that the concurrent execution of the forbidden set members is prohibited in every affected time period. Note that the LP relaxation (4)-(6), (10), and (12) may be weaker or stronger than (4)-(6), (8), and (10). Therefore, the LP relaxation (4)-(6), (8), (10), and (12) will be at least as strong as (4)-(6), (10), and (12) or (4)-(6), (8), and (10). Theoretically, (4)-(6), (8)-(10), and (12) is a redundant MILP model, in which either (8) or (12) is not necessary, but the redundant constraints, as valid cuts, greatly strengthen the LP relaxation of the model. Relaxing the integrality assumption, we get an LP problem, which - using a fast interior point solver - can be solved in reasonable time. When we solve the minimization and maximization problems, the sum of the differences of the optimal latest and earliest starting times gives an upper bound for the RCTPF measure:

$$\overline{\mathrm{RCTPF}} = \lfloor \overline{\mathrm{S}} \rfloor - \lceil \underline{\mathrm{S}} \rceil \tag{13}$$

The LP relaxation provides good quality upper bounds for the RCTPF measure, which is essential in a tree search process. According to the progress of the tree search process, the schedules become more and more resource constrained. The more constrained a schedule, the smaller the gap between the estimated and the true upper bounds.

3.2 A new implicit enumeration algorithm

The optimal schedule searching process is formulated as a tree search problem with effective pruning rules. The tree-building process is based on the forbidden set concept. The nodes of the tree correspond to "partial" schedules. In our model, any partial schedule satisfies all original precedence constraints and assigns a start time to all activities. But it is "partial" because it may violate one or more "visible" or "hidden" resource constraints. The nodes are characterized by the non-redundant subset of the original network relations and the additional resource conflict repairing relations, the feasible subset of the original forbidden sets, and the precedence but not necessarily resource feasible earliest (latest) starting times:

$$\left\{RS^{(n)}, FS^{(n)}, ES^{(n)} = \left\{ES_i^{(n)} \mid i \in \{1,2,\ldots,N\}\right\}, LS^{(n)} = \left\{LS_i^{(n)} \mid i \in \{1,2,\ldots,N\}\right\}\right\} \quad (14)$$

where $RS^{(n)}$ denotes the set of resource conflict repairing relations, and $FS^{(n)} \subseteq FS^{(0)} = FS$. Leaf nodes of search tree are resource feasible or pruned schedules. Our node evaluation (fitness) function is very simple: It assigns to each $n \in \{0,1,2,\ldots\}$ node the estimated upper bound value $\overline{RCTPF}^{(n)}$ of the RCTPF measure. Thus, at each step of the tree-building process, we select the most promising node, which has been generated but not expanded. A parent node is transformed into a set of child nodes by repairing its "least" resource conflict all the possible ways. Note that, in this context, "least" means a conflict with minimal number of possible repairing relations. According to our "best-first" searching strategy, a node without feasible resource conflict will be a solution of the RCPSP. Note that an inserted explicit conflict repairing relation (as its side effect) may be able to repair one or more other conflicts implicitly, at the same time.

In the traditional forbidden set oriented RCPSP solving strategy a parent node is transformed into a set of child nodes by repairing its first resource conflict all the possible ways, where "first" always means the earliest conflict in time interval $[1,T]$. The reason is very simple: in the traditional case we would like to get a resource feasible solution as early as possible and after that the searching process terminates.

In our case: (1) we have to generate all the solutions of the RCPSP, (2) the first RCPSP solution will not necessarily be optimal for the RCTPF measure, so we have to find other solutions (when we use the modified "best" conflict repairing strategy, the tree usually will be smaller than the traditional tree), and (3) the RCTPF measure is not a regular measure of performance, therefore a traditional regular pruning rule may "over-prune" the searching tree. Therefore, we have to develop special pruning rules to cut down the effective branching factor of the search tree. In this study we applied three pruning rules which are able to substantially reduce the number of generated nodes.

(1) The first "cyclic repairs rule" is a special consistency check, which can help to visibilize the "invisible" inconsistencies. The basic idea of this rule is very simple: After inserting a conflict repairing relation and updating the schedule, our tree-building process "looks ahead" and in a cyclically repeatable repairing process repairs each resource conflict in the child node which has exactly one repairing possibility. The cyclical repairing process immediately terminates and the child node is discarded if one ore more conflicts become non-repairable or the total project duration exceeds the prescribed maximal project duration.

(2) The second "at least as shiftable rule" is a straightforward modification of the well-known "left shiftable rule" which is an efficient regular pruning rule. Let $MS^{(n)}, n \in \{0,1,2,...\}$ denote the non-redundant subset of the predecessor-successor relations for the movable (non-critical) activities:

$$MS^{(n)} = \left\{ i \to j \,\middle|\, i \to j \in \text{NonRed}\left\{PS \cup RS^{(n)}\right\}, ES^{(i)} < LS^{(i)}, ES^{(j)} < LS^{(j)} \right\}, \tag{15}$$

The applied NonRed operator eliminates the redundant predecessor-successor relations, for example, $\text{NonRed}\{i \to j, j \to k, i \to k\} \Rightarrow \{i \to j, j \to k\}$.

The modified rule compares two nodes: If $MS^{(a)} \supseteq MS^{(b)}$, $ES^{(a)} \geq ES^{(b)}$, $LS^{(a)} \leq LS^{(b)}$, and $FS^{(a)} = FS^{(b)}$, then node a can be immediately pruned (node a is dominated by node b).

(3) The third rule is based on the proposed new MILP formulation of RCPSP. A child node is discarded if the relaxed LP solution is primal infeasible. In this study, to solve the relaxed LP, a very fast primal-dual interior method developed by Mészáros (1996) was used. In the tree building process, the applied new formulation is able to detect the resource unfeasibility earlier than the traditional formulation does, therefore results in a smaller tree. The algorithm maintains the dynamically changing $\{\text{Best Node}, \text{Best RCTPF}\} = \{\text{best}, \text{RCTPF}^{(\text{best})}\}$ set. A child node c can be immediately discarded if $\overline{\text{RCTPF}}^{(c)} \leq \text{RCTPF}^{(\text{best})}$.

Now, consider an example with twenty activities and only one resource type (Figure 3.1) . The activities are numbered 1 through 20 (plus the dummy activities 0 and 21). Only twelve units are available from the resource type. The minimal resource feasible makespan is $\underline{T}=19$, and we fix the position of the finishing milestone in $S_{21} = T+1 = 20$. The total number of network feasible schedules is $|\Re| = 110{,}525{,}184$, and the total number of resource feasible schedule sets is $|\overline{\Re}| = 459$, where a schedule set is characterized by the non-redundant subset of the network relations and the additional conflict repairing relations. The size of the search tree, which is needed to generate all the network and resource feasible solution sets, is 2642, using the implicit enumeration algorithm developed by Csébfalvi (2002). The RCTPF measure of the schedule sets is varying between 0 to 20. Table 3.1 and Figure 3.2 illustrate the essence of the example. The unique dummy source (sink) is represented by the $>$ ($<$) symbol.

Table 3.1 Project data and CPM results

i	D_i	ES_i	LS_i	R_{i1}	Immediate Predecessors
0	1	0	0		
1	1	1	4	3	{0}
2	1	1	6	5	{0}
3	6	1	2	5	{0}
4	3	2	5	4	{1}
5	3	2	8	5	{2}
6	5	2	7	3	{2}
7	4	7	8	2	{3}
8	4	5	8	3	{4}
9	3	5	11	3	{5}
10	1	7	12	3	{6}
11	2	11	12	4	{7}
12	4	9	12	1	{8}
13	1	8	14	2	{9}
14	2	8	13	5	{10}
15	1	13	14	1	{11}
16	1	13	16	4	{12}
17	5	10	15	3	{13,14}
18	3	14	15	5	{15}
19	3	14	17	2	{16}
20	2	17	18	1	{18}
21	1	20	20		{17,19,20}
Resource Availability				12	

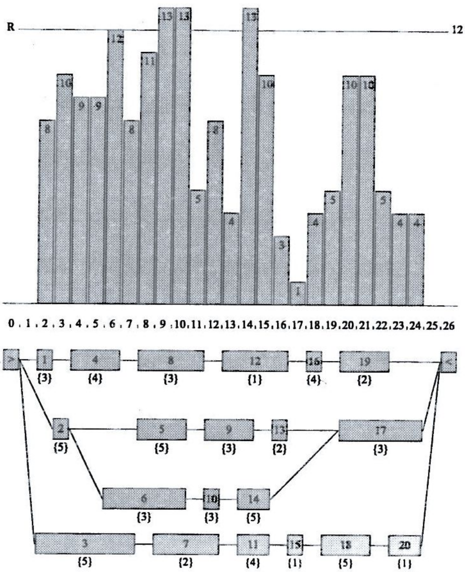


Figure 3.1 An example with twenty activities and only one resource type.

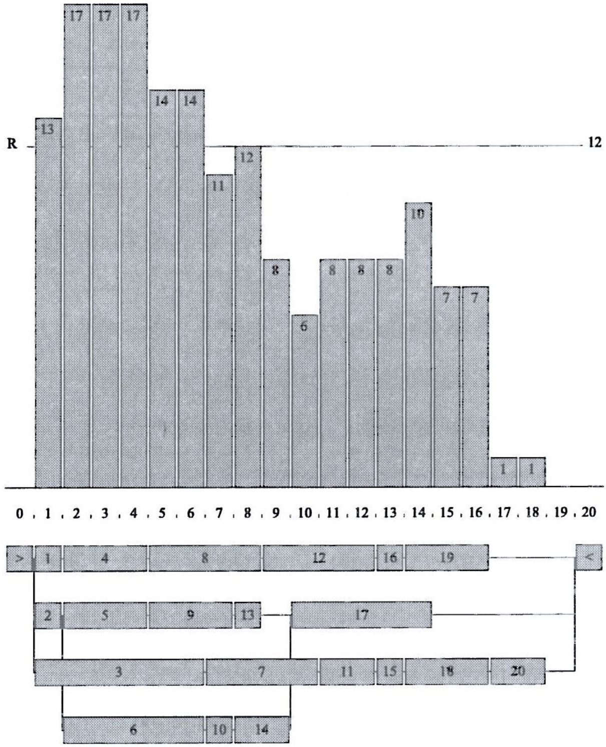


Figure 3.2 Early Start CPM Schedule: The unconstrained total project float UCTPF = 67.

Table 3.2 shows the forbidden sets and their explicit repairing sets in the presented earliest CPM schedule (Figure 3.2). In this schedule every conflict is feasible. A feasible conflict may be "visible" or "hidden". Note that a hidden conflict is "invisible" in the early CPM schedule, but might be visible in a shifted schedule. In the early schedule only four conflicts are visible.

Table 3.2 Forbidden Sets and Repairing Sets

f	Feasible -Visible Conflict	Conflict Interval	Forbidden Set	Repairing Set
1	Feasible -Visible	[02,07]	{3, 5, 6}	{3→5, 3→6, 5→6, 6→5}
2	Feasible -Visible	[02,07]	{3, 4, 5}	{3→5, 4→5, 5→4}
3	Feasible -Visible	[05,07]	{3, 6, 8, 9}	{3→6, 3→8, 3→9, 6→8, 6→9, 8→9, 9→8}
4	Feasible	[02,04]	{1, 3, 5}	{1→3, 1→5, 3→5}
5	Feasible	[05,06]	{2, 3, 8}	{2→3, 2→8, 3→8}
6	Feasible	[07,07]	{3, 8, 9, 10}	{3→8, 3→9, 3→10, 8→9, 9→8, 8→10, 10→8, 9→10, 10→9}
7	Feasible	[09,10]	{5, 7, 12, 14}	{5→7, 5→12, 5→14, 7→12, 7→14, 12→14, 14→12}
8	Feasible	[13,13]	{9, 14, 15, 16}	{9→14, 14→9, 9→15, 9→16, 14→15, 14→16, 15→16, 16→15}
9	Feasible	[02,06]	{2, 3, 4}	{2→3, 2→4, 4→2}
10	Feasible	[05,07]	{3, 4, 6, 9}	{3→6, 3→9, 4→6, 4→9, 6→9}
11	Feasible	[14,14]	{13, 14, 18, 19}	{13→14, 14→13, 13→18, 13→19, 14→18, 14→19, 18→19}
12	Feasible -Visible	[01,04]	{1, 2, 3}	{1→2, 2→1, 1→3, 2→3}
13	Feasible	[07,07]	{3, 4, 9, 10}	{3→9, 3→10, 4→9, 4→10, 9→10, 10→9}
14	Feasible	[13,13]	{11, 14, 16}	{11→14, 14→11, 11→16, 14→16}
15	Feasible	[14,14]	{12, 13, 14, 18}	{12→13, 13→12, 12→14, 14→12, 12→18, 13→14, 14→13, 13→18, 14→18}
16	Feasible	[05,07]	{3, 5, 8}	{3→5, 3→8, 5→8}
17	Feasible	[07,10]	{5, 7, 8, 10}	{5→7, 5→8, 5→10, 10→5, 7→10, 10→7, 8→10, 10→8}
18	Feasible	[07,07]	{3, 5, 10}	{3→5, 3→10, 5→10, 10→5}
19	Feasible	[08,10]	{5, 8, 14}	{5→8, 5→14, 8→14}
20	Feasible	[07,07]	{4, 5, 7, 10}	{4→5, 5→4, 4→7, 4→10, 5→7, 5→10, 10→5, 7→10, 10→7}
21	Feasible	[07,10]	{5, 6, 7, 8}	{5→6, 6→5, 5→7, 5→8, 6→7, 6→8}
22	Feasible	[11,13]	{9, 11, 12, 14}	{9→11, 9→12, 9→14, 14→9, 11→14, 14→11, 12→14, 14→12}
23	Feasible	[14,14]	{14, 16, 18}	{14→16, 14→18, 16→18}
24	Feasible	[08,11]	{7, 8, 9, 14}	{7→9, 9→7, 7→14, 8→9, 9→8, 8→14, 9→14, 14→9}
25	Feasible	[11,11]	{8, 9, 11, 14}	{8→9, 9→8, 8→11, 8→14, 9→11, 9→14, 14→9, 11→14, 14→11}
26	Feasible	[11,11]	{6, 8, 9, 11}	{6→8, 6→9, 6→11, 8→9, 9→8, 8→11, 9→11}
27	Feasible	[07,07]	{4, 5, 6, 7}	{4→5, 5→4, 4→6, 4→7, 5→6, 6→5, 5→7, 6→7}
28	Feasible	[11,11]	{8, 9, 10, 11}	{8→9, 9→8, 8→10, 10→8, 8→11, 9→10, 10→9, 9→11, 10→11}
29	Feasible	[11,11]	{8, 11, 13, 14}	{8→11, 8→13, 8→14, 11→13, 13→11, 11→14, 14→11, 13→14, 14→13}

The search tree of our example is presented in Figure5.3 in the dissertation. Figure 3.3 illustrate the optimal schedule set for the proposed RCTPF measure. In the optimal (most flexible) schedule set $RCTPF = 20$. The size of the search tree is 34. The proposed implicit enumeration algorithm solved the problem very quickly, the computation time was 0.134 sec.

Using exactly the same computational environment and the very sophisticated and fast CPLEX engine, the total computation time of the original repairing set oriented MILP model developed by Csébfalvi and Levi for the RCTPF measure was 2.0 sec (Iterations: 1514, Integer nodes: 262, Constraints: 303, Variables: 116, Binaries: 75, Non zeros: 958, Density: 3 %). This result reveals the fact that for medium size problems a standard MILP algorithm is not competitive with a specialized implicit enumeration algorithm. Sometimes the computation time needed to obtain optimal solutions is not much better than the time requirement of a brutal-force search.

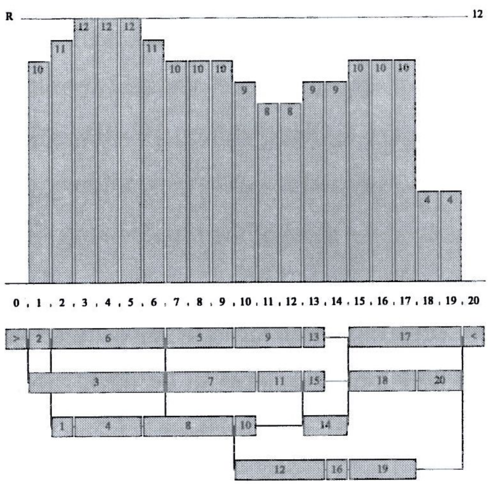


Figure 3.3 Optimal Schedule: RCTPF = 20

4. Computational results

In order to illustrate the essence and viability of the proposed new approach, detailed computational results for a subset (PAT16-PAT57) from the well-known Patterson's set are presented. Each problem in the investigated subset has twenty two activities and three resource types. The time limit on the project length for each project was assigned the value of the optimal solution to the corresponding RCPSP. See, for example, Bell and Park (1990) (For detailed computational results please refer to the dissertation).

As a MILP solver, the popular and very fast "state-of-the-art" CPLEX 8.1 (called from the MPL modeling environment) was used. Naturally, this solver can be replaced any other commercial MILP solver. The automatic MPL input file generator of the proposed model has been programmed in Visual C++® Version 6.0. The algorithm, as a DLL, was built into the *ProMan* system developed by Ghobadian and Csébfalvi (1995), Csébfalvi (2002). The computational results were obtained by running ProMan and MPL (CPLEX 8.1) on a 1.8 GHz Pentium IV IBM PC with 256 MB of memory under Microsoft Windows XP® operation system. The applied CPLEX algorithm solved the most of the problems very quickly, which is a promising and competitive result.

The presented benchmark problems can be used for testing the quality of exact and heuristic solution procedures to be developed in the future for this problem type.

5. New results1

The thesis contributes the following new results:

1. The introduction of a new robust Resource Constrained Total Project Float (RCTPF) measure for resource constrained projects which consists of two parts: The first part of the thesis introduces a new exact explicit enumeration algorithm for small size problem. Theoretically the optimal schedule searching process is formulated as a mixed integer linear programming (MILP) problem with big-M constraints, which can be solved directly for small-scale projects in a reasonable time. The presented resource conflict oriented MILP model, which is a reformulation of the traditional time oriented resource constrained project scheduling MILP model, can be used as a new resource constrained project scheduling model on its own right. The second part of the thesis presents a new exact implicit enumeration algorithm for the RCTPF measure. The algorithm is formulated as a tree-search problem with three effective pruning rules. According to the NP-hard nature of the problem, the proposed implicit enumeration algorithm provides exact solutions for small to medium size problems in a reasonable time. Large-scale problems can be managed by introducing an optimality tolerance which drastically decreases the size of the search tree. The obtained results illustrate the essence and viability of the proposed new algorithm and prove its feasibility.
2. A margin for several interesting improvements is left: (I) In the presented new MILP model the objective function can be replaced by any other objective function, which can be described as a function of

the earliest (latest) starting time variables. An interesting problem would be to develop a decision support system for the resource constrained "hammock" activities. (II) The definition of the objective function can be improved by adding relative weight to each activity, so that the optimal solution will take into consideration the preferences of the project manger regarding the allocation of total float to the more important and problematic activities. (III) In the proposed approach two LP problems must be solved in order to get an upper bound value. It is an open and very hard question, what would be the "best" big-M free formulation, which would be able to produce tighter upper bounds in one step. (IV)In the node-expanding phase very simple rules were applied in order to select the most promising node and conflict. It is a very interesting and challenging question, what would be the "best" selection-expansion strategy, which would be able to produce smaller trees. (V) An interesting point will be to examine the influence of this new approach on the management.

3. When integrated with the already formulated MILP solution to the RLP (Konstantinidis, 2002), the combined model (RLP and RAP) will provide an overall solution to both resource and time constrained projects.

---

[1] **Acknowledgment**

The author would like to express his thanks to Cs. Mészáros for making the unlimited WIN95/NT DLL version of the BPMPD (www.sztaki.hu/~meszaros/bpmpd) interior point solver available.

References

- Bell, C. E. and Park, K. 1990. "Solving Resource-constrained Project Scheduling Problems by A* Search". ***Naval Research Logistics, Vol. 37, pp. 61-84.***

- Bowers, J. A. 1995. "Criticality in Resource Constrained Networks". ***Journal of the Operational Research Society, Vol. 46, pp. 80-91.***

- Bowers, J. A. 2000. "Interpreting float in resource constrained projects". ***International Journal of Project Management, Vol. 18, pp. 385-392.***

- Csébfalvi, G. 2002. ProMan Manual (Version 2.1). ***Hungary: University of Pécs, Faculty of Business and Economics.***

- Demuelemeester, E. and Herroelen, W. 2002. Project Scheduling: A Research Handbook. ***Norwell, Massachusetts: Kluwer.***

- Ghobadian, A. and Csébfalvi, G. 1995. "Workshop on Developing Interactive Learning Material for Project Management," ***Proc. of 1995 Annual Meeting of Decision Sciences Institute, Boston, US.***

- Mészáros, Cs. 1996. "The Efficient Implementation of Interior Point Methods for Linear Programming and their Applications". ***Unpublished PhD Thesis, Eötvös Loránd University of Sciences, Hungary.***

- Ragsdale, C. 1989. "The current state of network simulation in project theory and practice". ***Omega, Vol. 17, pp. 21-25.***

- Raz, Tzvi and Bob Marshall. 1996. "Effect of resource constraints on float calculations in project networks". ***International Journal of Project Management, Vol. 14, pp. 241-248.***

- Weist, J. D. 1967. "A heuristic model for scheduling large projects with limited resources". ***Management Science, Vol. 13, pp. B-359-B-377.***

- Willis, R. J. 1985. "Critical path analysis and resource constrained scheduling – theory and practice". ***European Journal of Operational Research, Vol. 23, pp. 149-155.***

Publication list

Publications

- Csébfalvi, G. and Levi, R. 2004. "Criticality in Resource Constrained Projects - A New Flexibility Oriented Approach". ***Central European Journal of Operations Research (CEJOR) (under refereeing process).***

- Levi, R. 2004. "A New Resource Constrained Total Float Measure for Projects-An Implicit Enumeration Algorithm". ***Central European Journal of Operations Research (CEJOR) (under refereeing process).***
-

Conferences

- Hungarian Operation Research Conference XXVI, May 2004, Hungary.

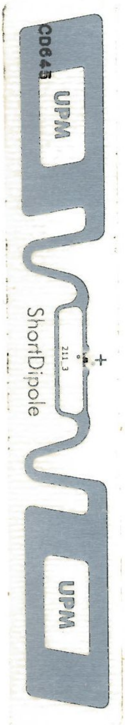
- APMOD 2004 – Applied Mathematical Programming and Modeling, Brunel University June 2004, U.K.

- VOCAL 2004 –Veszprém Optimization Conference: Advanced Algorithms), Veszprém, Dec. 2004, Hungary.

Presentations

- Technion Haifa – A panel for the presentation of contemporary research, June 2004, Israel

- Technion Haifa – A panel for the presentation of contemporary research, December 2004, Israel


CD645
UPM
ShortDipole
UPM