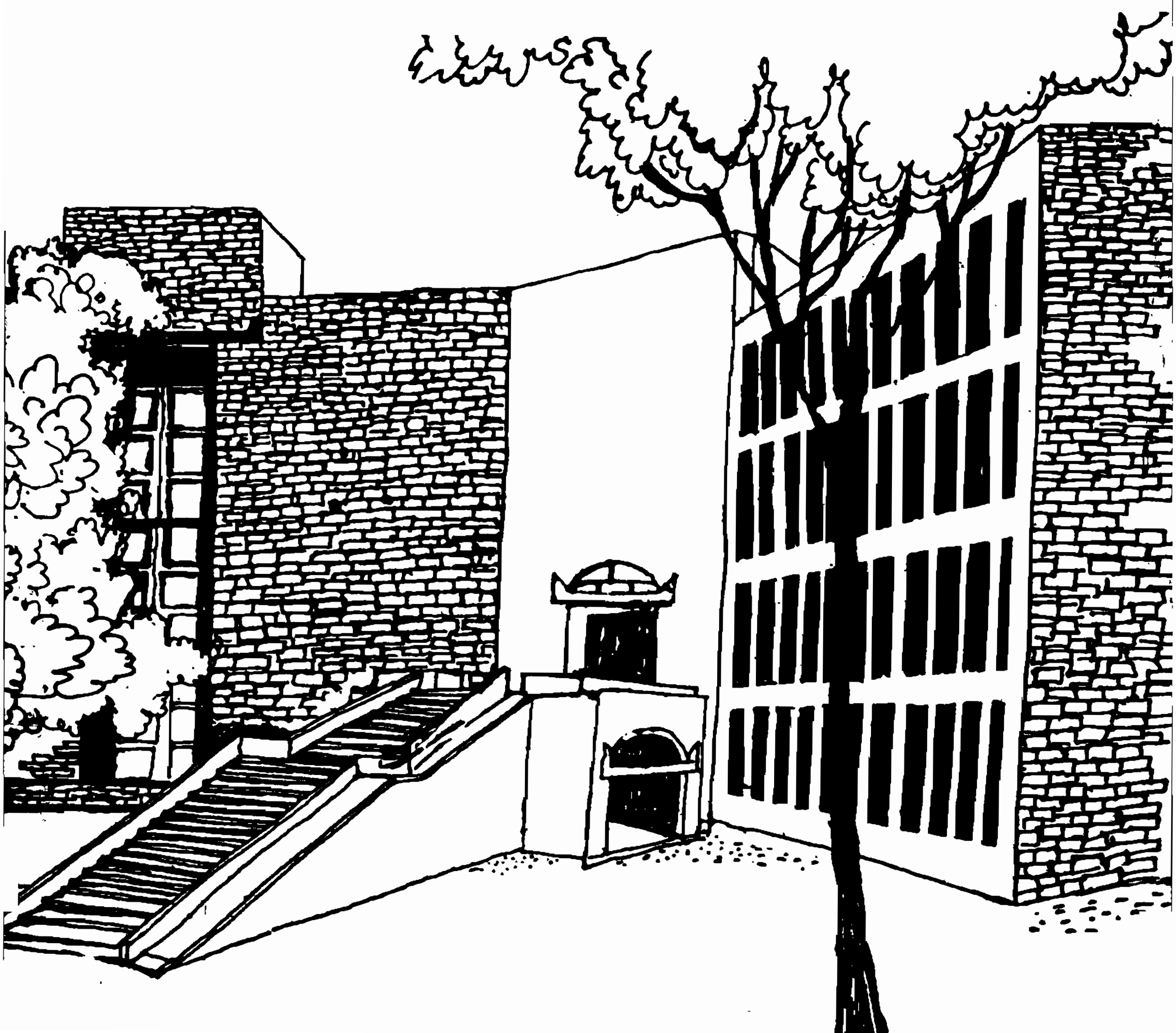
विद्याविनियोगाद्विकासः

IIM AHMEDABAD

Working Paper



COMMENT ON ICMELI AND ERENGUC'S "A BRANCH AND BOUND PROCEDURE FOR THE RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM WITH DISCOUNTED CASH FLOWS"

By

Sanjay Verma
N.P. Dhavale
Amitava Bagchi

W.P.No. 2000-04-03
April 2000 1595

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 015
INDIA

Comment on Icmeli and Erenguc's "A Branch and Bound Procedure for the Resource Constrained Project Scheduling Problem with Discounted Cash Flows"

Sanjay Verma
Indian Institute of Management Ahmedabad
Vastrapur, Ahmedabad - 380 015, India
E-mail: sverma@iimahd.ernet.in

and

N P Dhavale and Amitava Bagchi
Indian Institute of Management Calcutta
Joka, D H Road, P O Box 16757
Calcutta - 700 027, India
E-mail: bagchi@iimcal.ac.in

Abstract: In a recent paper, Icmeli and Erenguc (1996) present an algorithm to maximize the Net Present Value (NPV) of a given project under renewable resource constraints. It is shown here, with the help of an example, that in some cases the algorithm fails to find an optimal solution. A simple modification in the last step of the algorithm resolves the problem.

Keywords: Project Scheduling; Resource Constraints; Net Present Value

1 Introduction: Icmeli and Erenguc (1996) have proposed a branch and bound procedure to solve the resource constrained project scheduling problem with discounted cash flows. The procedure avoids resource violations by introducing additional precedence relationships between some pairs of project activities. The bounds are computed by solving Payment Scheduling Problems (Grinold, 1972), which can be formulated as linear programs.

Unfortunately, it seems there are problem instances for which the method fails to output optimal solutions. The difficulty can be rectified if a simple change is made in the algorithm. We illustrate the difficulty with an example, then suggest a modification to resolve the problem.

2 Problem Statement: The Resource Constrained Project Scheduling Problem with Discounted Cash Flows (RCPSPDC) can be stated as follows (Icmeli and Erenguc 1996):

$$\text{(P) Max } z(P) = \Sigma \{ q_i \cdot \exp(-\alpha t_i) \mid 1 \leq i \leq N \} \qquad (1)$$

Subject to

$$\Sigma \{ r_{ik} \mid i \in S_u \} \leq R_{uk} \quad \forall k, u, \qquad (2)$$

$$t_j - t_i \geq d_j, \ (i,j) \in H, \qquad (3)$$

$$t_N \leq DD, \qquad (4)$$

where

t_i is an integer decision variable specifying the completion time of activity i and t_N denotes the completion time of the project

N is the last activity

DD is the deadline by which the project must be completed

q_i is the compounded value of cash flows that occur during the implementation of activity i at its completion time t_i

$$q_i = \sum_{L=1}^{d_i} f_{iL} \ e^{\alpha(d_i - L)}, i = 1,\dots N$$

d_i is the duration of activity i

f_{iL} is the cash flow for activity i in period L of its duration, $L = 1,\ldots, d_i$, f_{iL} may be negative, zero or positive.

α is the discount rate

r_{ik} is the per period usage of resource k by activity i

R_{uk} is the availability level of resource k, $k = 1, \ldots, K$, at time period u

H are the set of precedence relationships among the activities;

$H = \{(i,j) \mid i \text{ precedes } j\}$.

S_u are the set of activities that are active at time period u. S_u is a function of the choice of t_j.

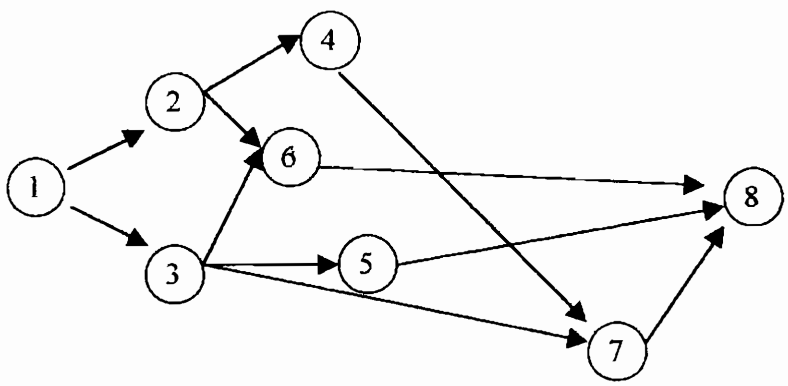


Figure 1: Activity on Node Project Network for Example 1

3 An Example: We show with a simple example that the algorithm may not give optimal solution in all the cases.

Example 1: Consider a project with eight activities and eleven precedence relationships as shown in Figure 1. The deadline of the project is 30 time units and the discount rate α is 0.0032 per period per money unit. The terminal cash flows, activity durations and resource requirements are as shown in Table 1. There is only one resource type with a total availability of 6 units.

Table 1: Data for Example 1

Activity	1	2	3	4	5	6	7	8
q_i	0	53	12	109	17	125	38	0
d_i	0	4	1	10	1	10	5	0
r_i	0	1	3	6	0	5	6	0

The search tree is shown in Figure 2, and the corresponding schedule for each node in the search tree is given in Table 2. At node S0, which is at level 0, the upper bound obtained by solving the unconstrained problem using Grinold's method (Grinold 1972) is 340.69. Since there is a resource conflict between activities 4 and 6, two child nodes S1 and S2 at Level 1 are generated. At node S1 activity 6 follows activity 4, and at node S2 activity 4 follows activity 6. Node S1 has an upper bound of 336.92 and node S2 has an upper bound of 336.28. Since node S1 has a higher upper bound, it is selected for expansion. On expansion of node S1, two child nodes S3 and S4 are generated at Level 2. Among these nodes, S3 has highest upper bound value (= 335.80) and it is also resource feasible. Therefore we update the lower bound obtained so far to the upper bound of S3 (=335.80), and fathom all states at level 2.

We now move back to level 1 (Step 3 of algorithm). We are supposed to fathom each node at level 1 if the upper bound is no more than the lower bound. At level 1 there is only one node S2 and its upper bound is 336.28 which is more that 335.80, so we do not fathom this node.

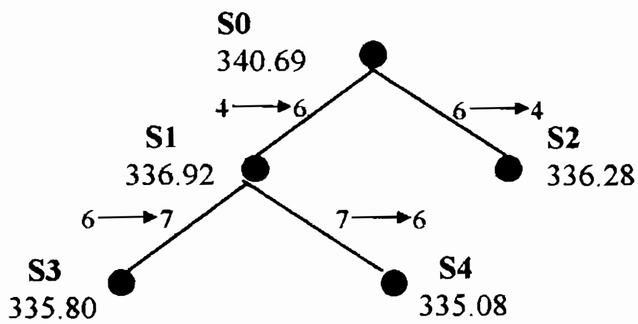


Figure 2: The Branch and Bound Tree for Example 1

Table 2: Start Times of Activities in Search Tree

	Nodes				
Activity	**0**	**1**	**2****	**3***	**4**
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	4	4	14	4	4
5	1	1	1	1	1
6	4	14	4	14	19
7	14	14	24	24	14
8	19	24	29	29	29

* Schedule outputted

** Optimal Schedule

Node S2 is now selected for expansion. At node S2 there is no resource conflict. We therefore backtrack to level 0 (Step 3). There is no other node to be expanded, so the solution (supposedly optimal) is outputted. The precedence constraints at node S1, S3, and S2 are shown in Figures 3, 4, and 5 respectively. The solution (with NPV = 335.80) is obtained at node S3; the corresponding schedule is shown in Table 2. However we miss the optimal schedule at node S2, which has an NPV of 336.28. □

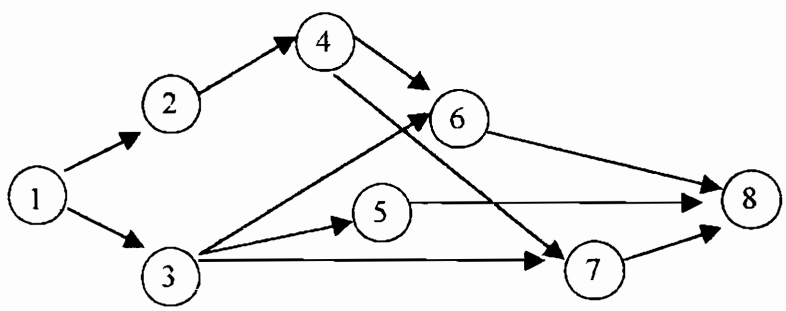


Figure 3: AON Network at Node S1 for Example 1

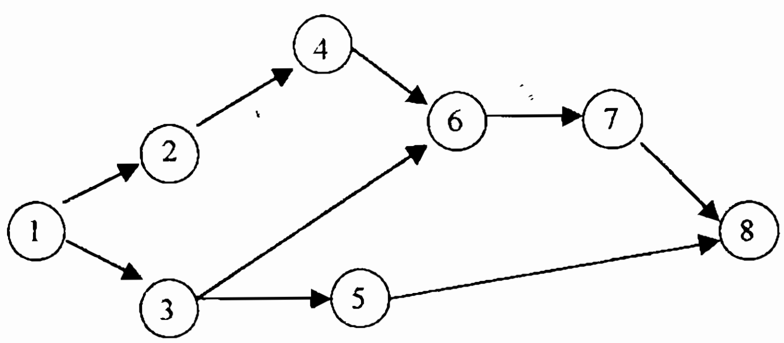


Figure 4: AON Network at Node S3 for Example 1

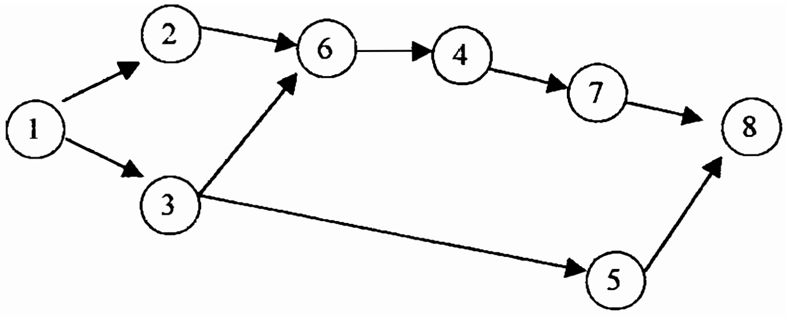


Figure 5: AON Network at Node S2 for Example 1

4. Modification: The method of Icmeli and Erenguc (IE) can be modified slightly to give the optimal solution. Only Step 3 needs to be changed as shown below.

Let P' be the problem obtained from the given problem P by ignoring the resource constraints. Grinold solves a problem equivalent to P' with the help of the *Fixed Deadline Algorithm* (FDA). IE uses FDA in its branch and bound procedure to solve the sub-problem at each node of the branch and bound tree. Below, we describe modified algorithm of Icmeli and Erenguc.

Let

p: Level in the branch and bound tree

C^k: All activities in progress in node Sk at the earliest resource violation in the schedule

DA^k: Sets of activities in C^k which need to be delayed so that no resource violation occurs

B^k: Total number of sons generated at node k of the tree, *i.e.*, number of sets in DA^k

H^k: Set of precedence relations at node k of the tree, including the initial precedence relations and those that have been subsequently appended.

x_k: The solution obtained by solving P'(k)

x_{INC}: Project schedule corresponding to the best solution found so far for P

LB: Objective function value for x_{INC}, which is a lower bound on the optimal value of P. Initially, LB is set equal to the value of a feasible solution if one is available, otherwise it is set equal to $-\infty$.

UB^k: The optimal objective function value of the sub-problem at node k.

NB: The candidate list, *i.e.*, the set of nodes yet to be expanded (branched from)

PR_k^b: The set of precedence relations added to problem P'(k) at branch b of node k to resolve the resource conflict; at the newly generated node that corresponds to branch b, the set of precedence relations becomes $(H^k \cup PR_k^b)$

θ_p: Set of all unfathomed nodes of level p.

δ_k: The length of the minimum spanning tree of the schedule obtained at node k.

ε: Accuracy desired for the solution (for optimality, $\varepsilon = 0$).

Step 3 (Modified): Backtrack.

3.1 Set $p \leftarrow (p-1)$.

If $p = 0$, then go to Step 4.

Else continue.

3.2 Fathom each node $n \in \theta_p$, if $UB^n \leq LB(1 + \varepsilon)$, and delete it from NB and θ_p.

3.3 If there exists no unfathomed node at this level (*i.e.*, if $\theta_p = \phi$) go to Step 3.1 (backtrack).

Else continue.

3.4 Find the node $n \in \theta_p$ that has maximum UB value, $k \leftarrow n$.

If x_k is not resource feasible then go to Step 1.

Else continue.

3.5 Set LB $\leftarrow$ UB^k, $x_{INC} \leftarrow x_k$. Fathom all nodes in θ_p and delete them from NB and θ_p. Go to Step 3.1 (backtrack).

Suppose we make this modification and solve Example 1. Then, before backtracking from level 1 to 0, the lower bound will be updated to the upper bound of S2. Here S2 is the resource feasible node with the maximum upper bound value. So the correct optimal solution will be outputted.

5. References:

Grinold, R. C., "The Payment Scheduling Problem," *Naval Research Logistics Quarterly*, 19, 1 (1972), 123-136.

Icmeli-Tukel, O. and Erenguc, S. S., "A Branch and Bound Procedure for the Resource Constrained Project Scheduling Problem with Discounted Cash Flows," *Management Science*, 42, 10 (1996), 1395-1408.