



A branch-and-bound procedure for resource leveling in multi-mode resource constraint project scheduling problem

Afshar-Nadjafi Behrouz*, Najjarbashi Hojjat and Mehdizadeh Esmaeil

Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, IRAN

Available online at: www.isca.in

Received 24th March 2012, revised 3rd March 2012, accepted 2nd April 2012

Abstract

This paper presents an exact solution for MRCPSP model (Multi Mode Resource Constraints Project Scheduling Problem). Each activity could be performed in various ways (mode) and in every style of performance (mode) a specific amount of time and resources are used for each activity. The solution used here includes branch and bound and the study aims to level resources. In the end some sample problems are solved using expanded algorithm and the results are reported and analyzed.

Key words: project scheduling, branch and bound, resource leveling, MRCPSP.

Introduction

For many real-life projects (e.g. in civil engineering, process industries) it is possible to perform the individual activities in alternative ways (modes). These modes differ in processing time, time lags to other activities, and resource requirements. They reflect time–resource trade-offs and resource–resource trade-offs.

This assay follows a branch of project timing issues with a special objective function (resource leveling) that have usually been rarely considered. The MRCPSP model is extended form of the RCPSP model (resource-constrained project scheduling problem) that has been chosen for this research for resource leveling.

The assumptions of the multi-mode RCPSP can be summarized as follows. An activity j must be performed in one of its modes which are labeled $1; \dots; M_j$ with M_j being the number of modes. Once started in one of its modes, an activity must be completed in that mode; mode changes and preemption are not permitted. The processing time of activity j being executed in mode m is given by p_{jm} . The request of activity j being executed in mode m for resource k is r_{jmk} . In addition to renewable resources, often also nonrenewable resources are considered in multi-mode models. The standard MRCPSP model consists of choosing a mode for each activity and assigning a time to start each activity so that resource constraints and precedence constraints are preserved and the project completion time reaches a minimum. Several exact and heuristic approaches to solve the MRCPSP have been proposed in recent years. An overview of the available exact and heuristic procedures in literature is given.

Exact procedures: In the first solution method for the multi-mode problem presented a one-stage and two-stage linear programming approach¹. Also for this problem, proposed a

depth-first branch-and-bound algorithm, but it have shown that this algorithm may be unable to find the optimal solution for instances with two or more renewable resources²⁻³. More recently, some authors, presented branch-and-bound algorithms⁴⁻⁷.

Heuristics: However, none of exact procedures can be used for solving large-sized realistic projects, since they are unable to find an optimal solution in areasonable computation time. Therefore, different single-pass heuristic and meta-heuristic procedures are presented⁸⁻¹⁶.

Meta-heuristics: Genetic algorithms have been presented for larg scale problems¹⁷⁻²⁰. Also, simulated annealing approach and particle swarm and hybrid scatter-search used to tackle the MRCPSP²¹⁻²⁴.

For resource leveling problems with special objective functions and only special minimum time lags, several exact and heuristic solution procedures have been proposed. Exact algorithms based upon (implicit) enumeration, integer programming, or dynamic programming²⁵⁻²⁸.

Material and Methods

Problem description and model presenting: Assume a directed acyclic graph with several real, certain and clear activities. Let the members of V set represent these activities. And the net is of the kind of activity on node(AoN). Vectors represent precedence constraints. Also activities are numbered in a way so that each activity's number is more than all its prerequisite activities. The activity 0, $N+1$ is added to the beginning and the end of the project that shows respectively the beginning and end of the project and both are dummies. The activity 0 is prerequisite for all the activities and the $N+1$ activity is requisite after every activity. Each activity $u \in V$ is performed in a mode m_u , which is chosen out of a set of $|M_u|$

different execution modes ($M_u = \{1, 2, \dots, |M_u|\}$). The activity 0, n+1 could only be performed in one mode ($M_0 = M_{n+1} = 1$) and its performance time and requiring resource is zero, ($r_{0,1} = r_{n+1,1} = 0$), ($d_{0,1} = d_{n+1,1} = 0$). No gap is allowed in activities and each activity is performed exactly after the ending of the preceding one with no change in performance style. The project has a specific and predetermined due date that cannot be delayed. The zero-one programming model as follows:

The objective in equation (1) minimizes the changing level of the resource. Equation (2) ensures that every activity is programmed only in one mode and only in one specific time to start the program. Equation (3) every activity is started after its prerequisite activities. Equation (4) ensures that no activity use more than a specific state line and Equation (5) is the assurance of due date of the project.

Branch and Bound Algorithm: In this section, we give a description of the branch-and-bound procedure.

Branching Strategy: Some of the different strategies defined for branching in branch and bound algorithm for RCPSP that could be mentioned here are precedence tree, minimal delaying alternatives, extension alternatives and minimal forbidden sets of course, the branching styles are different according to the problem type. The branching style we have used here is a combined style so that the first branching is converting MRCPS to RCPSP sub-branch problems and then precedence tree strategy is used for branching. For further dissection more than one numeral example are used in the explanations. Look at the figure-1 network.

Table-2 shows each activity's duration and each activity's use of renewable resources in different modes. One renewable resource exists.

Resources are convertible to each other and we can add these changeable by β_k :

$$Z = \min \sum_{k=1}^K \beta_k \left(\sum_{t=1}^{DD+1} \left| \sum_{n=1}^N \sum_{m=1}^{M_n} r_{nmk} \cdot \sum_{s=\max(t-d_{nm}+1, es_n)}^{\min(t, ls_n)} X_{nms} - \sum_{n=1}^N \sum_{m=1}^{M_n} r_{nmk} \cdot \sum_{s=\max(t-1-d_{nm}+1, es_n)}^{\min(t-1, ls_n)} X_{nms} \right| \right) \quad (1)$$

$$\sum_{m=1}^{M_n} \sum_{t=es_n}^{ls_n} X_{nmt} = 1 \quad n = 1, 2, \dots, N \quad (2)$$

$$\sum_{m=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + d_{im}) X_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=es_j}^{ls_j} t X_{jmt} \quad \forall (i, j) \in E \quad (3)$$

$$\sum_{n=1}^N \sum_{m=1}^{M_n} r_{nmk} \cdot \sum_{s=\max(t-d_{nm}+1, es_n)}^{\min(t, ls_n)} X_{nms} \leq R_k \quad t = 1, 2, \dots, T, \quad k = 1, 2, \dots, K \quad (4)$$

$$ls_N + d_{N+1,1} \leq DD \quad m = 1, 2, \dots, M_n \quad (5)$$

Table-1
Parameters for MRCPS

Parameter	Definition	Parameter	Definition
N	Number of activities indexed by n	K	Number of renewable resources indexed by k
ls _n	Latest finish time of activity n	r _{nmk}	Per-period usage of renewable resource k required to execute activity n in mode m
es _n	Earliest finish time of activity n		
d _{nm}	Time required to perform activity n in mode m	R _k	Per-period availability of renewable resource k
		β _k	convertible resource coefficient
M _n	Number of modes of activity n, indexed by m	X _{nmt}	$\begin{cases} 1; & \text{if activity n is performed in mode m} \\ & \text{and start in period t} \\ 0; & \text{otherwise} \end{cases}$
DD	Due date		
E	the set of edges (arcs)		

Table-2
Duration and resource required for each activity in each mode

activity	0		1		2		3		4		5	
mode	d _{0m}	r _{0m}	d _{1m}	r _{1m}	d _{2m}	r _{2m}	d _{3m}	r _{3m}	d _{4m}	r _{4m}	d _{5m}	r _{5m}
1	0	0	2	2	3	1	4	2	1	3	0	0
2			1	3	2	3	3	4				

The project's due date, is 8 time unites after the start of the project and 4 unites of renewable resource in each time unite are available. This problem will be converted to 8 RCPSP problems 3 of which are summarized in the table-3.

Table-3
Number of RCPSP for example Figure-1

Activity	0	1	2	3	4	5	rr _o	Z _o
1	1	1	1	1	1	1	18/8	6
2	1	2	1	1	1	1	17/8	6
3	1	1	2	1	1	1	21/8	6
4	1	1	1	2	1	1	22/8	6
5	1	2	2	1	1	1	20/8	6
6	1	2	1	2	1	1	21/8	6
7	1	1	2	2	1	1	25/8	8
8	1	2	2	2	1	1	24/8	6

Numbers on the left of the table are the RCPSP problem number, numbers up the table are the activity number, and numbers in the table are the mode allocated to each activity. Now each of the problems should be considered separately and all their time scheduling should be calculated considering the objective function quantity.

Each of the level one nodes show a RCPSP problem whose allocated mode to the activities are mentioned in table-3. Other than bounding that will be mentioned in next sections in the next level we open the first node that will be chosen according to the choosing strategy. Choosing strategy is like this:

1-Calculate the average of the required day resource at each node and find the objective function based on that.

$$rr_0 = \left(\sum_{i=1}^N d_{i, Mode(i,0)} * r_{i, Mode(i,0)} \right) / DD \quad (6)$$

$$Z_0 = 2 * ([rr_0] + 1) \quad (7)$$

rr₀ shows the average of the required day resource, DD the project due date and the function mode (i,o) shows the allocated mode to the ith activity in number o, RCPSP problems.

2-Organize the finding numbers in descending order. The branch the node with least quantity based on the branching rules. The results are mentioned in the last two columns of table-3.

The minimum quantity of the objective function belongs to RCPSP problem number 2. Then we open this node according to level 2 branching rules. Notice the fact that the quantity found for the objective function in Z_o is an ideological number and the appropriate solutions to the RCPSP problem can never be better than that. The real quantity for the objective function will be found after the end of time scheduling.

Bounding Rules: As explained in the last section, to solve a MRCPSP problem we divide it to several RCPSP problems. Then we choose the RCPSP problem with the least rr_o, (o=1, 2, ..., O) and thence we perform the rules of level 2 branching. While level 2 branching, three bounding rules that cause minimizing the search space will be used to get an appropriate solution. Then we use the real quantity of the objective function and we shorten the RCPSP problem.

First bounding rule: Lemma one: if a time period in which no activity is being done exists in a scheduling program, the quantity of objective function of this problem will be found by another problem.

Proof: A and B in both figures-2-a and 2-b include similar activities (the required time and resource and the order of settlement). Activities i and j are exactly the same. Only in picture 2-b between activities i and j exists a time span with no activity. The quantity found for the objective function in both part A and B is the same in both figures.

$$\text{Final } Z_{(\text{Figure-2a})} = \Delta V_A + \Delta V_{Ai} + \Delta V_{ij} + \Delta V_{jB} + \Delta V_B$$

$$\text{Final } Z_{(\text{Figure-2b})} = \Delta V_A + \Delta V_{Ai} + \Delta V_{iL} + \Delta V_{Lj} + \Delta V_{jB} + \Delta V_B \quad \frac{\Delta V_{Lj} = \Delta V_{iL} + \Delta V_{ij}}{=}$$

$$\text{Final } Z_{(\text{Figure-2b})} = \Delta V_A + \Delta V_{Ai} + 2\Delta V_{iL} + \Delta V_{ij} + \Delta V_{jB} + \Delta V_B$$

$$\text{Final } Z_{(\text{Figure-2a})} - \text{Final } Z_{(\text{Figure-2b})} = \Delta 2\Delta V_{iL}$$

ΔA and ΔB represent the changes of the objective function in the A and B activity sets.

Using this lemma the node 3 and node 4 and etc. in figure 3 could be bounded. The bounded nodes with this rule will show by LB1 sign.

Second bounding rule: Because of the special structure that branching strategy based on prerequisite tree needs, during the branch process sequences might be produced whose result is a similar timing scheme. To avoid counting the repeated timing sequences this bordering rule can be used. This rule is defined as this: assume two time scheduled activities as i and j are time scheduled in two previous levels and the current level of the algorithm tree. If these activities have started at the same time (S_i = S_j) and i > j, then the current time scheduling is not needed to be completed. Node 16 signed as LB2 is noted in figure-3.

Third bounding rule: After reaching the first appropriate solution and calculating the real quantity of the objective function, consider this solution as the upper bound (to minimize the changes if the resource level) and we omit branches with more or equal solutions. This upper bound is able to be used for other RCPSP problems. After finding an solution with the better objective function (if existed) we replace this solution with the upper bound. The bounded nodes with this rule are signed as LB3.

Forth bounding rule: As mentioned the level two branching rules, open a RCPSP problem and the bounding rules 1,2,3, and 4 shorten it and this process goes on until we find an appropriate solution. After opening a RCPSP problem and reaching the end we go the next RCPSP problem in the organized list (based on rr_o). Of course if the idealistic quantity of the objective function of the new RCPSP problem is not better than the real quantity of the objective function, (z , in figure-3 showed with a circle) in the previous RCPSP problem there is no need to open a new problem. As you can see the idealistic quantity of all the RCPSP problems are less than the founded real quantity in node 32 therefore all the nodes could be bounded.

Pseudo code: All of what was presented to solve a problem in our branch and bound algorithm is mentioned in the following Pseudo code. The Z_o and rr_o parameters are respectively the idealistic quantity of the target function and the source day average. BS is the best found solution, S_i is the starting time of the process of the i th activity, z is the real quantity of the objective function after calculation and Ea is the the set of activities able to be chosen in each node.

Step 1: For all RCPSP, Calculate rr_o and Z_o , Sort RCPSPs rr_o form smallest to largest, $BS = \infty$,

Step 2: Select the RCPSP with smallest rr_o (if this, RCPSP not selected before), $S_0 = 1$, Set Ea .

Step 3: Do $\forall i \in Ea$, Do for $t = ES_i$ to LS_i , If activity i can be start at t , Create a new node, $S_i = t$,

Step 4: For each new node product in step 2, Set Ea , If $Ea = \emptyset$ this branch is ended, Calculate the performance measure z , If $z \leq BS$, $BS = z$, Else go to step 2

Results and Discussion

30 different problems with different features are resolved for studying of the branch and bound algorithm and its compatibility with the issue MRCPSP. We used RANGEN II software for production of these problems. Table-4 shows the parameters involved in the issues.

Table-4
The parameter settings for the problem set

Parameter	Description	Value
N	Number of activities	8, 10, 12
M	Number of modes per activity	2, 3
D_{im}	Duration of each mode	[1 5]
	Maximum number of predecessors and successors	4
K	Number of renewable resource types	1
r_{imk}	Renewable resource demand	[1,5]
RF	Resource factor for renewable resources	1

The produced sample problems solved with the algorithm developed by the software package MATLAB version 7.11.0.584 (R2010b). This algorithm has been coded by a personal computer with 2.66GHZ CPU and the Intel Core2Duo processor and 3GB RAM and Windows 7 Ultimate Edition 2009. Table 5 represents the average and the standard deviation of CPU-time, in seconds, for a different number of activities and execution modes.

Table-5
Computational results on the MRCPSP

activities	8		10		12	
	μ	σ	μ	σ	μ	σ
modes						
2	0.576436	0.63268	3.553614	16.85578	424.1988	577199.7
3	2.949574	4.92352	75.60519	11084.33	-	-

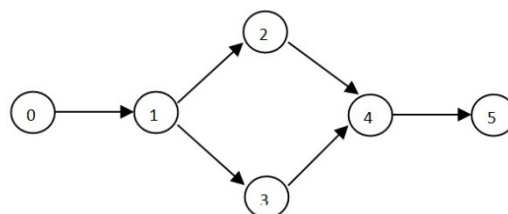


Figure-1
Example network

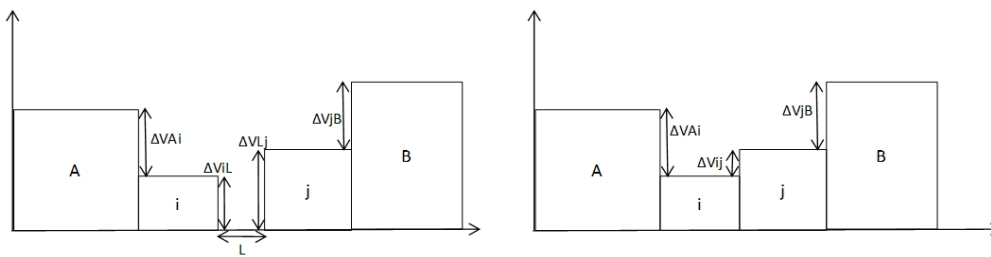


Figure-2b

Figure-2a

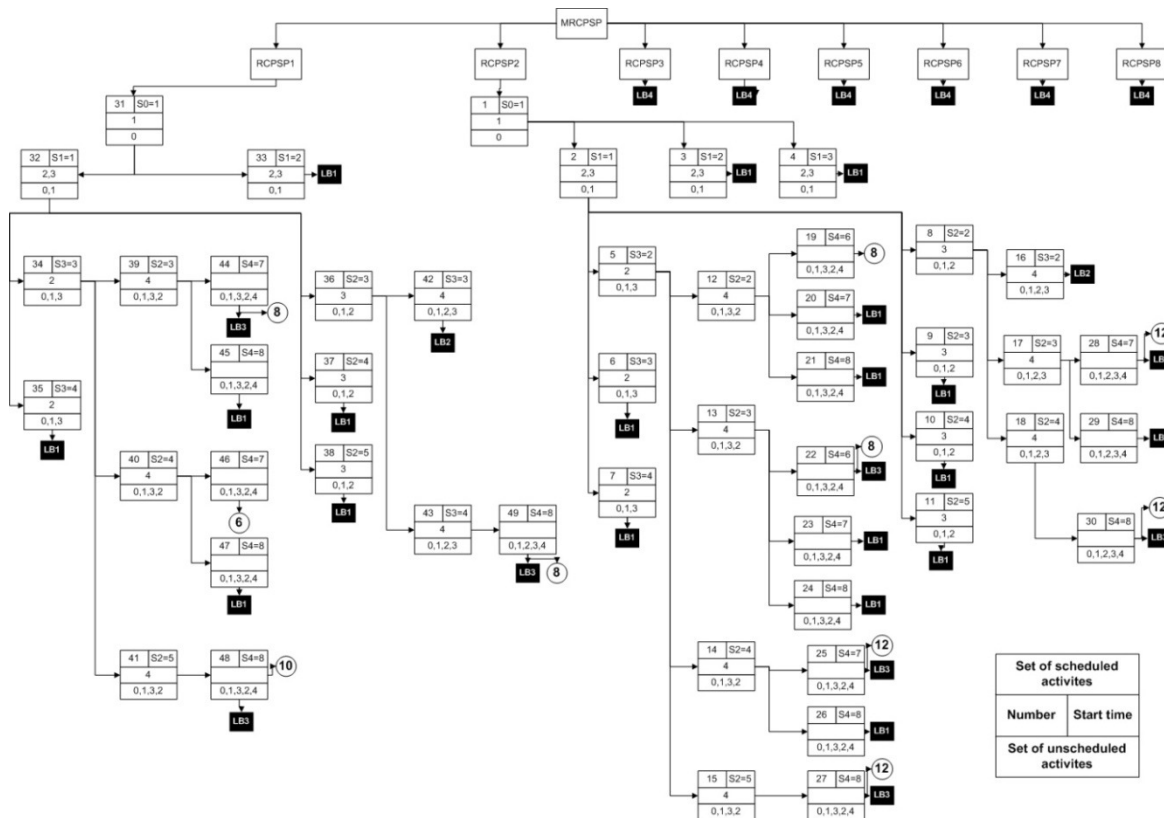


Figure-3
 Tree of example Figure-1

Conclusion

This paper reports on an exact branch and bound procedure for multi-mode resource constrained project scheduling problem with resource leveling objective. Depth first branching is based on precedence tree approach. Four rules are used for node fathoming. Finally, the new branch and bound procedure is used for solving some test problems and computational results demonstrate that the proposed branch and bound is in fact capable of solving problems in acceptable time. Statistical analysis revealed that, solving is incredibly increasing when the number of activities are enhancing.

References

1. Slowinski R., Two approaches to problems of resource allocation among project activities – A comparative study, *J. Oper. Res. Soci.*, **8**, 711–723 (1980)
2. Speranza M. and Vercellis C., Hierarchical models for multi-project planning and scheduling, *Eur. J. Oper. Res.*, **64**, 312–325 (1993)
3. Hartmann S. and Sprecher A., A note on hierarchical models for multi-project planning and scheduling, *Eur. J. Oper. Res.*, **94**, 377– 383 (1996)

4. Sprecher A., Hartmann S. and Drexl A., An exact algorithm for the project scheduling with multiple modes, *OR. Spe.*, **19**, 195–203 (1997)
5. Hartmann S. and Drexl A., Project scheduling with multiple modes: A comparison of exact algorithms, *Net.*, **32**, 283–297 (1998)
6. Sprecher A. and Drexl A., Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm, *Eur. J. Oper Res.*, **107**, 431–450 (1998)
7. Zhu G., Bard J. and Tu G., A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem, *J. Comp.*, **18(3)**, 377–390 (2006)
8. Boctor F., Heuristics for scheduling projects with resource restrictions and several resource-duration modes, *Int. J. of Prod. Res.*, **31(11)**, 2547–2558 (1993)
9. Drexl A. and Grünewald J., Non preemptive multi-mode resource-constrained project scheduling, *IIE Trans.*, **25**, 74–81 (1993)
10. Ozdamar L. and Ulusoy G., A local constraint based analysis approach to project scheduling under general resource constraints, *Eur. J. Oper Res.*, **79**, 287–298 (1994)
11. Boctor F., A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes, *Eur. J. Oper Res.*, **90**, 349–361 (1996)
12. Kolisch R., Drexl A., Local search for non-preemptive multi-mode resource constrained project scheduling, *IIE. Trans.*, **29**, 987–999 (1997)
13. Knotts G., Dror M., Hartman B., Agent-based project scheduling, *IIE. Trans.*, **32 (5)**, 387–401(2000)
14. Lova A., Tormos P. and Barber F., Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules, *Intelig. Artif.*, **30**, 69–86 (2006)
15. Patel M., Jain S., Jain D. and Patel B., Prevalence of Different Factors Responsible for Infertility, *Res. J. Recent Sci.*, **1(ISC-2011)**, 207-211 (2012)
16. Zangeneh M., Simulation of Perturbation in the PG, *Res. J. Recent Sci.*, **1(1)**, 77-80 (2012)
17. Özdamar L., A genetic algorithm approach to a general category project scheduling problem, *IEEE. Trans. on Sys., Man, and Cybernetics, Part C: Applications and Reviews* **29**, 44–59 (1999)
18. Hartmann S., Project scheduling with multiple modes: A genetic algorithm, *Ann. of Oper. Res.*, **102**, 111-135 (2001)
19. Alcaraz J., Maroto C. and Ruiz R., Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms, *J. Oper. Res. Soci.*, **54**, 614–626 (2003)
20. Lova A., Tormos P., Cervantes M. and Barber F., An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes, *Int. J. of Prod. Econ.*, **117 (2)**, 302-316 (2009).
21. Jozefowska J., Mika M., Rozycki R., Waligora G. and Weglarz J., Simulated annealing for multi-mode resource-constrained project scheduling, *Ann. of Oper. Res.*, **102**, 137–155 (2001)
22. Bouleimen K. and Lecocq H., A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple modeversion, *Eur. J. Oper Res.*, **149**, 268–281 (2003)
23. Zhang H., Tam C. and Li H., Multimode project scheduling based on particle swarm optimization, *Comp Aided. Civ. and Infra. Eng.*, **21**, 93–103(2006)
24. Ranjbar M., De Reyck B., and Kianfar F., A hybrid scatter-search for the discrete time/resource trade-off problem in project scheduling, *Eur. J. Oper Res.*, **193 (1)**, 35–48 (2008)
25. Bandelloni M., Tucci M. and Rinaldi R., Optimal resource leveling using non-serial dynamic programming, *Eur. J. Oper Res.*, **78**, 162-177 (1994)
26. Demeulemeester E., Minimizing resource availability costs in time-limited project networks, *Manag. Sci.*, **41**, 1590-1598 (1995)
27. Savin D., Alkass S. and Fazio P., A procedure for calculating the weight-matrix of a neural network for resource leveling, *Adv. in Eng. Soft.*, **28**, 277-283 (1997)
28. Neumann K. and Zimmermann J., Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints, *Eur. J. Oper Res.*, **127**, 425-443 (2000)