

Research Article

A Branch and Bound Approach to Solve the Preemptive Resource Leveling Problem

Behrouz Afshar-Nadjafi, Zeinab Khalaj, and Esmail Mehdizadeh

Faculty of Industrial and Mechanical Engineering, Islamic Azad University, Qazvin Branch, P.O. Box 34185-14161, Qazvin, Iran

Correspondence should be addressed to Behrouz Afshar-Nadjafi; afsharnb@alum.sharif.edu

Received 10 March 2013; Accepted 5 September 2013

Academic Editors: A. Che, G. Dessen, A. Lockamy, and G. Onwubolu

Copyright © 2013 Behrouz Afshar-Nadjafi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study resource constrained project scheduling problem with respect to resource leveling as objective function and allowance of preemption in activities. The branch and bound algorithms proposed in previous researches on resource leveling problem do not consider preemption. So, representing a model for the problem, a branch and bound algorithm is proposed. This algorithm can handle preemption in resource leveling problem. Comparing the resource leveling problem and the preemptive resource leveling problem, it is observed that considering preemption in the problem leads to better results in the objective function. This improvement imposes additional time to solve the problem. Coding the algorithm in MATLAB and checking it on the projects with 8 and 10 activities, results show that the proposed algorithm is efficient.

1. Introduction

Increasing international competition enforces utilization of very expensive resources (e.g., heavy machines) by most of the companies. Thus, resource constrained project scheduling problem (RCPSP) has attracted more attention. Several objective functions are studied in RCPSP. Resource leveling is one of them, in which the variation of resource utilization is to be minimized. In addition to resource constraints, minimum and maximum time lags between different activities have to be observed in general.

Several studies have been done for RCPSP. The first mathematical formulation of the RCPSP was given by Pritsker et al. [1]. Then, Kaplan [2], Olaguíbel and Goerlich [3], and Klein and Scholl [4] continued their job. In addition, Blazewicz et al. proved that RCPSP is NP-hard [5]. Kastor and Sirakoulis analyzed the effectiveness of three resource leveling problem tools on RCPSP: Primavera p6.0, Microsoft Project 2007, and Open Workbench 1.1.6 [6].

For resource leveling problems with minimum time lags, several exact and heuristic solution procedures have been studied.

Exact algorithms contain enumeration, integer programming, or dynamic programming techniques. Tavares represented the effectiveness of resource leveling problem in costs

[7]. Ahuja [8], Easa [9], Bandelloni et al. [10], Demeulemeester [11], and Younis and Saad [12] presented exact procedures for resource leveling problem. Nübel developed a branch and bound procedure (BB) upon minimal delaying alternative and disjunction precedence constraints [13]. Demeulemeester and Herroelen proposed a model for project scheduling problem with resource constraints and preemption and a model for resource leveling problem and solved it with some methods, such as BB [14]. Gather et al. proposed exact methods for solving the resource leveling problem [15].

Some studies contain both exact and heuristic methods. Neumann and Zimmermann introduced three groups of objective functions for resource leveling problem [16, 17]. They proved that the issue of resource leveling with each studied objective function is NP-hard and proposed an exact and a heuristic procedure for the problem expressed in the paper. In their paper, BB and truncated BB procedures with and without resource limitations are discussed. Coughlan et al. presented a branch-and-price approach together with a new heuristic to solve resource leveling problem [18].

There are some works to solve resource leveling problem using heuristics. To solve resource leveling problem with general objective function and maximum time lags, Brinkmann and Neumann [19] and Neumann and Zimmermann [20] proposed heuristic methods upon priority rules. Savin et al.

used neural network to solve resource leveling problem [21, 22]. Moder and Philips [23], Moder et al. [24], Harris [25, 26], and Takamoto et al. [27] presented heuristic procedures to solve resource leveling problem.

Preemption is an important factor in our daily projects and jobs. Stopping a job when the night comes, interrupting a course as the class time is out, putting the call on hold to answer someone else, drinking coffee while reading newspaper, and so forth are some examples of preemption in our life. By the aspect of preemption, there are three types of activities: activities which will be continued after preemption (preempt-resume type), activities which should be restarted after preemption (preempt-repeat type), and activities for which preemption is not allowed. Because of its role in our life, preemption has attracted some researchers to work on it using both exact methods and heuristics.

Several studies on RCPSP use exact methods to solve the problem. Demeulemeester and Herroelen proposed an optimal solution procedure for the preemptive resource constrained project scheduling problem to minimize the project makespan [28]. The maximum number of preemptions for each activity was considered specific for the activity. A model and a BB algorithm were presented for solving the problem. Patterson et al. [29] developed a BB algorithm upon precedence tree, which was reviewed by Sprecher [30]. Stinson et al. [31], Demeulemeester and Herroelen [32], Igelmund and Radermacher [33], and Reyck and Herroelen [34] studied BB procedures for the RCPSP with generalized precedence relations. Afshar-Nadjafi et al. have provided a branch and bound procedure for resource leveling problem in multimode RCPSP [35].

As an example for heuristics, Ballestín et al. studied preemption in project scheduling with resource constraints [36, 37]. They show that one preemption to be allowed for each activity creates a proper state and more than one preemption just increases computational time with a slight improvement in the final solution. In their model, the number of preemptions allowed for each activity is specific for the activity. Then, an evolutionary algorithm with a coding and a crossover operator for the model has been presented. Vanhoucke et al. investigate the effect of variable activity durations under fixed work content, the possibility of allowing activity preemption, the use of fast tracking on the total lead time, and the total resource utilization of resource constrained projects [38].

To the best of our knowledge, no research has studied the usefulness of preemption in resource leveling problem. In this paper, we study effectiveness of preemption in resource leveling problem. We consider the preempt-resume type of activities. Here, resources are considered renewable and constrained. We will propose a new model for the problem and a B&B procedure to schedule activities in order to achieve the most leveled project.

The rest of the paper will be as follows. In Section 2 we will introduce the new model for preemptive resource leveling problem. In Section 3, the proposed BB algorithm will be described for the problem. Finally, computational results will be represented in Section 4.

2. Problem Definition

In this paper, we are supposed to schedule activities with maximum p preemptions allowed for each activity, with respect to resource leveling as the objective function. If preemption is justified for the activity, we will detect the proper period in which it occurs. In this paragraph, we will study the components of the model. In Section 2.1 we will study objective function. In Section 2.2 problem assumptions are stated. Constraints are described in Section 2.3. Parameters and variables are proposed in Section 2.4. Finally, we will get familiar with the model of the problem in Section 2.5.

2.1. Objective Function. Resource leveling problem determines a sequence for the activities of a project which will minimize the amount of variation. This issue is important in determining the optimal sequence of activities to minimize the level of hiring and firing workers or that of utilization and unemployment of machines. Leveling limited resources or unlimited resources prevents the excessive fluctuations of the resources and reduces their cost. The chosen objective function for this study is to level the cumulative amount of resources used in each moment to the former one. We consider the resource amount used in both the first day and the last day. If the project is finished a day before the deadline, the resource used in the last day is considered by the formula. But if it is finished on the deadline, we must add the consumption to the objective function.

2.2. Problem Assumptions. In the project scheduling problem, n activities which use k resources are intended. Finish-start relationships (FS) are the most commonly used. The activity on arc network can easily be converted to the activity on node network, so we use the second one. The assumptions of this problem are as follows.

- (i) The project involves n activities, of which the initial and the final ones are dummies.
- (ii) The project is presented by an activity on node network.
- (iii) The number of resources is K .
- (iv) General relations of activities are FS⁰.
- (v) Each activity can be preempted for at most p times.
- (vi) Preemption is free of time and cost.
- (vii) Resources are renewable.

2.3. Constraints. In RCPSP, activities should be scheduled so that the time constraints and resource constraints are respected. Obviously, the model should consider precedence relations between activities. Hence, the problem constraints of the model are considered as follows.

- (i) Maximum resource k availability in each period is R_k units.
- (ii) Project deadline is at the end of the period T .
- (iii) Precedence relations should always be respected.

2.4. *Variables and Parameters.* In this model, we use these parameters and variables:

n : number of activities,

R_k : maximum availability of resource k ,

T : deadline of the project,

K : number of resources,

p : maximum number of preemptions,

$r_{i,k}$: requirement of activity i to the resource type k ,

d_i : duration of activity i ,

E : the set of precedence relations between activities,

es_i : earliest start time of activity i ,

lf_i : latest finish time of activity i ,

$y_{i,t}$: $\begin{cases} 1, & \text{activity } i \text{ is in progress at period } t, \\ 0, & \text{else,} \end{cases}$

$x_{i,t}$: $\begin{cases} 1, & \text{activity } i \text{ is preempted at period } t, \\ 0, & \text{else,} \end{cases}$

f_t : the set of activities which can be in progress at period t :

$$f_t = \{i \mid \exists i \in \{2, \dots, n-1\} : es_i \leq t < lf_i\}. \quad (1)$$

Note that when you work on the job i at period t , $y_{i,t}$ gets the value one, but if the job is interrupted before t and is not restarted yet, $y_{i,t}$ gets the zero one. Again, if the job i is preempted just at period t , then $x_{i,t}$ gets the value one. But if it is still interrupted at the next period, t' , $x_{i,t'}$ gets zero. It is seen that $x_{i,t}$ and $y_{i,t}$ cannot get the value one, simultaneously.

2.5. *The Proposed Model.* The model proposed in this paper is as follows:

$$\min z = \sum_{t=1}^T \sum_{k=1}^K \left| \sum_{i \in f_t} r_{i,k} y_{i,t} - \sum_{i \in f_{t-1}} r_{i,k} y_{i,t-1} \right| \quad (2)$$

subject to

$$\sum_{i \in f_1} y_{i,1} > 0, \quad (3)$$

$$\sum_{t=es_i}^{lf_i} y_{i,t} = d_i, \quad i = 1, \dots, n, \quad (4)$$

$$y_{j,t} \leq \frac{1}{d_i} \sum_{t'=es_i}^{\min(t-1, lf_i)} y_{i,t'}, \quad (i, j) \in E, \quad (5)$$

$$t = es_j, \dots, lf_j; \quad i \neq 1,$$

$$\sum_{t=es_i+1}^{lf_i-1} x_{i,t} \leq p, \quad i = 2, \dots, n-1, \quad (6)$$

$$\sum_{i \in f_t} r_{i,k} y_{i,t} \leq R_k, \quad t = 1, \dots, T, \quad k = 1, \dots, K, \quad (7)$$

$$x_{i,t} = \begin{cases} 1; & y_{i,t} = 0, y_{i,t-1} = 1, \sum_{t'=es_i}^t y_{i,t'} < d_i, \\ 0; & \text{else,} \end{cases} \quad (8)$$

$$i = 2, \dots, n-1, \quad t = es_i + 1, \dots, lf_i - 1,$$

$$x_{i,t}, y_{i,t} \in \{0, 1\}; \quad i = 1, \dots, n, \quad t = es_i, \dots, lf_i. \quad (9)$$

Equation (2) presents an objective function that minimizes the resource utility period to period. Note that the objective function includes the resource used in the first day and the last one too. So, we should add the resource usage of the first day to the objective function. Again, if the project is finished on the deadline, we must add the amount of resources used in the day to the function. By restriction (3), the earliest start time of zero is assigned to the project (at least one activity is scheduled at the first day) and (4) guarantees that each activity is in progress for d_i time units. Restriction (5) considers generalized precedence relations of activities (while the preceding activity is not completely scheduled, the predecessor will not be scheduled), while (6) stipulates the preemption constraints. Restriction (7) specifies the resource constraints and (8) has formulated the period in which preemption occurs for each activity. The variables are defined as integers in (9).

3. Branch and Bound Procedure

Here, a BB algorithm is proposed to solve the preemptive resource leveling problem. In Section 3.1 we will have some definitions used in the algorithm. In Section 3.2 calculation of the function value at each node is stated. Upper bound and dominance rules are considered in Sections 3.3 and 3.4. Finally, we will illustrate the algorithm with a numerical example in Section 3.5.

3.1. *Definitions Used in the Algorithm.* Some definitions used in the BB algorithm should be cleared. Here, we define “node,” “branch,” and “critical path” in the algorithm.

Node: in this algorithm, each node represents the activities scheduled at the current day.

Branch: here, a branch means a partial scheduling. Indeed, a branch contains activities which are scheduled from the first day up to now.

Here, when an activity has not been started, its critical path contains its duration. But when an activity has been started and is scheduled for a while, the critical path contains its remaining duration.

The algorithm presented in this paper is a depth-first one. The process of branching continues until all of the nodes are checked. In fact, in branching, all the activities which can be

scheduled are checked (if the preceding activities are scheduled before and the activity has not been completed yet); nodes contain all combinations of these activities. For each node all the constraints (resource requirement and maximum preemption) are checked. If the constraints are satisfied, we will use the dominance rules. Searching all nodes, the process ends. Components of the algorithm are proposed as follows.

3.2. *Calculating the Function Value at Each Node.* Function value at each node is the total variance in resource utility from day to day (this amount includes the amount of resources used on the first day). It is notable that the final value of the objective function of a branch, which is returned at the end of the branch, applies the resource used in the last day of the project (the final node of the branch).

3.3. *Upper Bound.* At the first level, the upper bound is double the total amount of resource requirements of all activities. This amount will be achieved if all the activities are scheduled one by one without parallel execution with at least one-day break between them (Figure 1). So, in any feasible solution without a break, the function value will not be worse. Continuously, the objective function of the first feasible solution achieved is accepted to be the upper bound. Achieving a better solution, it will be considered as the upper bound.

3.4. *Dominance Rules.* As mentioned before, the dominance rules are needed to close branches without a better solution, but they should not remove a promising solution. Only the branches with nonoptimal or unfeasible solution shall be discontinued at the earliest time possible. For all the nodes, resource and preemption constraints are checked. If the constraints are satisfied, then the dominance rules will be used; if not, we will continue with another node. Now, the rules for this algorithm are as follows.

- (a) If the cumulative time of the current node and the critical path of activities scheduled in the node (maximum critical path of the activities) exceed the deadline of the project, the node is interrupted. For example, if in second day of a project, an activity with a four-day-long critical path is scheduled and the deadline is five, then the branch is interrupted. Without the rule you will continue branching and it will be finished at the fifth day instead of the second.
- (b) Obviously, if in the current node there is an activity for which more than p preemption occurred, the node is bounded.
- (c) If the cumulative amount of the objective function at a node and the resource used at the node exceed the upper bound, the node is interrupted. As an example, when the upper bound is six units, a node by objective function amount of four units that uses three units of the resource exceeds the upper bound ($4 + 3 > 6$), so it is interrupted.
- (d) It is obviously seen that if the resource utility and the objective function of the current node and a node in

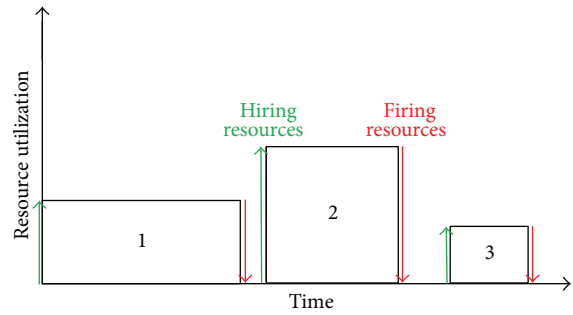


FIGURE 1: Upper bound.

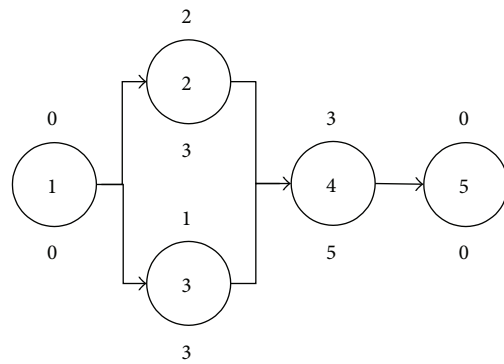


FIGURE 2: Project example.

a different branch are the same and if the activities scheduled in the current branch are a subset of the others, the current branch will not lead to a better solution. So, the current node will be interrupted and will not be continued. As an example, consider a branch that has scheduled at least a part of activities $\{i_1, i_2, i_3, i_4\}$ with objective function of five units and resource utility of 4 units in the node l . Now, suppose that the current node (m) has scheduled at least a part of activities $\{i_1, i_2, i_3\}$ with objective function of five units and resource utility of 4 units. Then the node m is not better than node l and will not be continued.

3.5. *Numerical Example.* Here, to better understand the proposed algorithm, an example is given. For this purpose, we choose the project network as depicted in Figure 2. The numbers above the nodes show the activity durations and the numbers below show the amount of resources required for that activity. The first activity and the last one are dummy activities as shown in Figure 2.

Here, we assume that the amount of available resources is 6 units and the project deadline is at the end of the fifth day. The first upper bound is $22: 2 * (0 + 3 + 3 + 5 + 0)$.

The first node of the branch and bound procedure puts up activity 1. In Level 2, activities 2 and 3 have the ability to be scheduled; three nodes are created: put up activity 2 alone, activity 3 alone, or both 2 and 3. The proposed algorithm selects the first node and the branching continues. This process is also repeated for the second day. On day 4 only

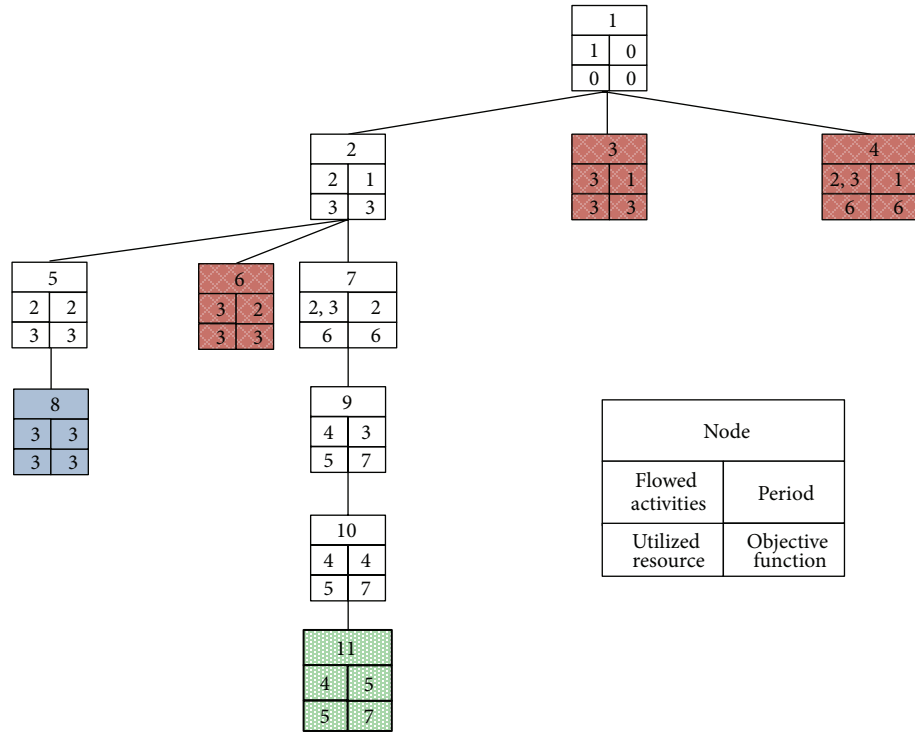


FIGURE 3: Branch and bound tree for project example.

activity 3 can be scheduled. But then the project will be done in 6 days. So, the first rule closes the branch (node 8). Then node 6 is selected. But using the fourth rule, it is bounded by node 8. Branching continues from node 7. A solution is achieved at node 11. So, the new upper bound is 12 (7 + 5). The last nodes are, respectively, interrupted nodes 6 and 7 by the fourth rule. The only solution achieved is the optimal one (Figure 3).

4. Computational Results

The branch and bound algorithm of Section 3 has been coded in MATLAB version 7.10.0.499 (R2010a). We have tested the algorithm on a computer with an Intel Core 2 Duo processor 4 GB Ram. We have used PROGEN as the problem generator. The deadline of each project is considered the critical path plus 20% of it. We have tested the algorithm for projects with 8 and 10 activities. For each group 50 examples are tested simultaneously in both problems: resource leveling problem and preemptive resource leveling problem. For easier computation, we have considered one resource and maximum one preemption for each activity. Considering 8 activities, resource leveling problem could not find a solution for 8 examples. And, considering 10 activities, it could not find a solution for 13 examples. So, comparing the results with those of nonpreemptive resource leveling problem, we notice 35% improvement in number of solved problems with 10 activities (a scheduling is returned for 37 projects in resource leveling problem and for 50 projects in preemptive resource leveling problem) and 19% improvement in number of solved problems with 8 activities (a scheduling is returned for 42

TABLE 1: Number of solved examples for projects with 8 and 10 activities.

Number of activities	Resource leveling problem	Preemptive resource leveling problem	% improvement
8	42	50	%19
10	37	50	%35

projects in resource leveling problem and for 50 projects in preemptive resource leveling problem) (Table 1). By the solved problem, we mean the program has found a solution which is finished on time, or maybe in time, with respect to constraints. In fact, see Table 1.

Table 2 contains computational results. As seen in the table, for projects with 8 activities in resource leveling problem, an average objective function of 26.14 is gained with a deviation of 4.03. These results are reached in an average computational time of 0.52 seconds with a deviation of 1.00 seconds. For projects with 10 activities in resource leveling problem, the average objective function is 29.46 with a deviation of 4.80. Here, the average computational time achieved is 6.09 seconds with a deviation of 7.22 seconds. In preemptive resource leveling problem, for projects with 8 activities, an average objective function of 24.21 with a deviation of 3.16 seconds is gained, with an average computational time of 3.00 seconds with deviation of 4.88 seconds. Finally, in preemptive resource leveling problem, for projects with 10 activities, the average objective function reached is 26.16 with a deviation of 4.77. Computational time has the average of 16.03 seconds

TABLE 2: Computational results for resource leveling problem and preemptive resource leveling problem with 8 and 10 activities.

Number of activities	Objective function average		CPU time average (sec.)	
	Resource leveling	Preemptive resource leveling	Resource leveling	Preemptive resource leveling
8	26.14 ± 4.03	24.21 ± 3.16	0.52 ± 1.00	3.00 ± 4.88
10	29.46 ± 4.80	26.16 ± 4.77	6.09 ± 7.22	16.03 ± 18.27

TABLE 3: Efficiency of each dominance rule.

Number of activities	Amount of nodes bounded by rule a	Amount of nodes bounded by rule c	Amount of nodes bounded by rule d
8	0.3418	0.3983	0.2599
10	0.4534	0.2903	0.2563

with a deviation of 18.27 seconds. Because of recursion limit of 500 in MATLAB, more than 10 activities in a project could not be done for most of the time.

The results verify the improvement of optimal solutions obtained; when preemption is applied into the time it is not allowed. With the increasing number of activities, time to solve the problem dramatically increased. To compare the results, each rule is excluded in all the examples. Comparing the nonpromising branches bounded with rules, their efficiency is checked. Efficiency of the dominance rules is reported in Table 3. Results show that bounding rule b is more efficient for problems with 8 activities, whereas bounding rule a is more efficient for problems with 10 activities. Also, none of the rules are useless.

The algorithm presented by Neumann and Zimmermann solved resource constrained resource leveling problem instants with 15 activities in less than 10 seconds and 99.2 of them in less than 100 seconds [17]. For 10 activities, it solved 98.0 of the instants in less than 10 seconds. As reported, our algorithm solved the problem in the average of 6.09 seconds. Also, our algorithm, taking into account the preemptions, is preferable.

5. Conclusions

In this paper, we proposed a model for preemptive resource leveling problem. The objective of the problem is to minimize the variance in resource utilization, as is traditional for the resource leveling problem. We defined a limited maximum number of preemptions per activity, as one of the model constraints. A branch and bound algorithm is represented to solve the problem. Finally, the program is used for solving the problem, with and without allowance of preemption. It is obvious that considering preemption leads to a better value for the objective function but this solution is reached in more time.

The comparison results showed that the proposed branch and bound algorithm is competitive with the algorithm presented by Neumann and Zimmermann to solve resource constrained resource leveling problem without preemption [17].

For future research, metaheuristic algorithms can be used as a method to solve the problem with large scales problems.

Also, one can consider nonrenewable resources and generalized precedence relations (GPRs) to tackle real-life situations. The results of this work have two fundamental usages. First, it can be used as benchmark for comparison with future developed methods, especially metaheuristics. Second, the solutions obtained from proposed method can be used by project managers looking for more leveled schedule using the opportunity of preemption.

References

- [1] A. A. B. Pritsker, L. J. Watters, and P. M. Wolfe, "Multi-project scheduling with limited resources: a zero-one programming approach," *Management Science*, vol. 16, no. 1, pp. 93–108, 1969.
- [2] L. A. Kaplan, *Resource-constrained project scheduling with preemption of jobs [Ph.D. thesis]*, University of Michigan, 1988.
- [3] R. A.-V. Olaguibel and J. T. Goerlich, "The project scheduling polyhedron: dimension, facets and lifting theorems," *European Journal of Operational Research*, vol. 67, no. 2, pp. 204–220, 1993.
- [4] R. Klein and A. Scholl, "Computing lower bounds by destructive improvement: an application to resource-constrained project scheduling," *European Journal of Operational Research*, vol. 112, no. 2, pp. 322–346, 1999.
- [5] J. Blazewicz, J. K. Lenstra, and A. H. G. R. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, 1983.
- [6] A. Kastor and K. Sirakoulis, "The effectiveness of resource levelling tools for Resource Constraint Project Scheduling Problem," *International Journal of Project Management*, vol. 27, no. 5, pp. 493–500, 2009.
- [7] L. V. Tavares, "A review of the contribution of Operational Research to Project Management," *European Journal of Operational Research*, vol. 136, no. 1, pp. 1–18, 2002.
- [8] H. N. Ahuja, *Construction Performance Control By Networks*, John Wiley & Sons, 1976.
- [9] S. M. Easa, "RL in construction by optimization," *Journal of Construction Engineering and Management*, vol. 115, pp. 302–316, 1989.
- [10] M. Bandelloni, M. Tucci, and R. Rinaldi, "Optimal resource leveling using non-serial dyanamic programming," *European Journal of Operational Research*, vol. 78, no. 2, pp. 162–177, 1994.
- [11] E. Demeulemeester, "Minimizing resource availability costs in time-limited project networks," *Management Science*, vol. 41, pp. 1590–1598, 1995.
- [12] M. A. Younis and B. Saad, "Optimal resource leveling of multi-resource projects," *Computers and Industrial Engineering*, vol. 31, no. 1-2, pp. 1–4, 1996.
- [13] H. Nübel, "A branch-and-bound procedure for the resource investment problem with generalized precedence constraints," Tech. Rep. WIOR-516, 1998.
- [14] E. Demeulemeester and W. Herroelen, *Project Scheduling A Research Handbook*, 2002.

- [15] T. Gather, J. Zimmermann, and J.-H. Bartels, "Exact methods for the resource levelling problem," *Journal of Scheduling*, vol. 14, no. 6, pp. 557–569, 2011.
- [16] K. Neumann and J. Zimmermann, "Heuristic methods for resource-constrained project scheduling with regular and non-regular objective functions and schedule-dependent time windows," in *Project Scheduling: Recent Models, Algorithms, and Applications*, J. Weglarz, Ed., pp. 261–287, Kluwer Academic Publishers, Boston, Mass, USA, 1999.
- [17] K. Neumann and J. Zimmermann, "Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints," *European Journal of Operational Research*, vol. 127, no. 2, pp. 425–443, 2000.
- [18] E. Coughlan, M. Lübbecke, and J. Schulz, "A branch-and-price algorithm for multi-mode resource leveling," in *Experimental Algorithms*, vol. 6049 of *Lecture Notes in Computer Science*, pp. 226–238, 2010.
- [19] K. Brinkmann and K. Neumann, "Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: the resource-leveling and minimum project duration problems," *Journal of Decision Systems*, vol. 5, pp. 129–155, 1996.
- [20] K. Neumann and J. Zimmermann, "Resource levelling for projects with schedule-dependent time windows," *European Journal of Operational Research*, vol. 117, no. 3, pp. 591–605, 1999.
- [21] D. Savin, S. Alkass, and P. Fazio, "Construction resource leveling using neural networks," *Canadian Journal of Civil Engineering*, vol. 23, no. 4, pp. 917–925, 1996.
- [22] D. Savin, S. Alkass, and P. Fazio, "A procedure for calculating the weight-matrix of a neural network for resource leveling," *Advances in Engineering Software*, vol. 28, no. 5, pp. 277–283, 1997.
- [23] J. J. Moder and C. R. Phillips, *Project Management with CPM and PERT*, Van Nostrand Reinhold, 1970.
- [24] J. J. Moder, C. R. Phillips, and E. W. Davis, *Project Management With CPM, PERT and Project Diagramming*, Van Nostrand Reinhold, 1983.
- [25] R. B. Harris, *Precedence and Arrow Networking Techniques for Construction*, John Wiley & Sons, 1978.
- [26] R. B. Harris, "Packing method for resource leveling (pack)," *Journal of Construction Engineering and Management*, vol. 116, no. 2, pp. 331–350, 1990.
- [27] M. Takamoto, N. Yamada, Y. Kobayashi, H. Nonaka, and S. Okoshi, "Zero-one quadratic programming algorithm for resource leveling of manufacturing process schedules," *Systems and Computers in Japan*, vol. 26, no. 10, pp. 68–76, 1995.
- [28] E. L. Demeulemeester and W. S. Herroelen, "An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 90, no. 2, pp. 334–348, 1996.
- [29] J. H. Patterson, R. Slowinski, F. B. Talbot, and J. Weglarz, "An algorithm for a general class of precedence and resource constrained scheduling problems," *Advances in Project Scheduling*, vol. 187, pp. 3–28, 1989.
- [30] A. Sprecher, *Resource-Constrained Project Scheduling—Exact Methods For the Multi-Mode Case*, Lecture Notes in Economics and Mathematical Systems, 1994.
- [31] J. P. Stinson, E. W. Davis, and B. M. Khumawala, "Multiple resource-constrained scheduling using branch-and-bound," *AIIE Trans*, vol. 10, no. 3, pp. 252–259, 1978.
- [32] E. Demeulemeester and W. S. Herroelen, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem," *Management Science*, vol. 38, pp. 1803–1818, 1992.
- [33] G. Igelmund and F. J. Radermacher, "Preselective strategies for the optimization of stochastic project networks under resource constraints," *Networks*, vol. 13, no. 1, pp. 1–28, 1983.
- [34] B. De Reyck and W. Herroelen, "A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations," *European Journal of Operational Research*, vol. 111, no. 1, pp. 152–174, 1998.
- [35] B. Afshar-Nadjafi, H. Najjarbashi, and E. Mehdizadeh, "A branch-and-bound procedure for RL in multi-mode resource constraint project scheduling problem," *Research Journal of Recent Sciences*, vol. 1, no. 7, pp. 33–38, 2012.
- [36] F. Ballestín, V. Valls, and S. Quintanilla, "Pre-emption in resource-constrained project scheduling," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1136–1152, 2008.
- [37] F. Ballestín, V. Valls, and S. Quintanilla, "Scheduling projects with limited number of preemptions," *Computers and Operations Research*, vol. 36, no. 11, pp. 2913–2925, 2009.
- [38] M. Vanhoucke and D. Debels, "The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects," *Computers and Industrial Engineering*, vol. 54, no. 1, pp. 140–154, 2008.

