

Associate Professor Yinbin LIU, PhD

E-mail: yinbinliu@126.com

Ziyi HU, Master student

E-mail: 512608474@qq.com

Associate Professor Hongbo LI, PhD (Corresponding author)

School of Management, Shanghai University, China

E-mail: ishongboli@gmail.com, hongbo_li@shu.edu.cn

Associate Professor Hongli ZHU, PhD

School of Business Administration, Shandong Technology and Business University, Yantai, China

E-mail: zhilirz@163.com

DOES PREEMPTION LEAD TO MORE LEVELED RESOURCE USAGE IN PROJECTS? A COMPUTATIONAL STUDY BASED ON MIXED-INTEGER LINEAR PROGRAMMING

***Abstract.** As an effective project resource scheduling technique, resource leveling has been extensively investigated. A common assumption in this technique is that activities are not allowed to be interrupted during execution. Little attentions have been paid to the preemptive resource leveling problem (PRLP) in the literature. However, preemption is often adopted in project management practice. In this paper, we discuss whether preemption leads to more leveled resource usage in projects. We propose a mixed-integer linear programming model for the PRLP. Computational experiments are performed on a large number of randomly generated instances to verify the improvement gained by incorporating preemption to the RLP. Additionally, a real-world case study is used to further illustrate whether preemption improves resource leveling in practical project management.*

***Keywords:** project scheduling; resource leveling; preemption; mixed-integer linear programming.*

JEL Classification: M11, C44, C61

1. Introduction

One-third of the global GDP is generated by projects, and the total value generated by project management is as high as 12 trillion dollars (Dobrovolskienė

and Tamošiūnienė 2016). Project scheduling plays a critical role in the success of a project (Mehdizadeh and Dadgar 2014, Li et al. 2018b, Mehdizadeh and Akbari 2017). As an effective project resource scheduling technique, resource leveling has been widely adopted. The resource leveling problem (RLP) aims at minimizing the variations of the project resource utilization by scheduling activities (Demeulemeester and Herroelen 2002). Effective resource leveling can help companies to utilize expensive renewable resources (e.g. machines, equipment, or highly qualified manpower) more efficiently by avoiding the peak resource usage.

Many exact and heuristic procedures have been devised to solve the RLP. Main exact approaches for the RLP include dynamic programming (Bandelloni et al. 1994), branch-and-bound procedures (Gather et al. 2011), and integer programming (Rieck et al. 2012, Kreter et al. 2013). For the heuristics, most of them rely on shift-based methods (Burgess and Killebrew 1962, Woodworth and Willie 1975, Neumann and Zimmermann 1999, Christodoulou et al. 2010). In addition, metaheuristics based on tabu search (Koulinas and Anagnostopoulos 2013, Li et al. 2014), iterated greedy algorithm (Ballestín et al. 2007), path-relinking (Ranjbar 2012), genetic algorithm (Li and Demeulemeester 2016, Li et al. 2018a), and estimation of distribution algorithm (Li and Dong 2018) are also studied.

A common assumption in the RLP is that activities are not allowed to be interrupted during execution. Little attention has been paid to the preemptive resource leveling problem (PRLP) in the literature. However, preemption is often adopted in project management practice. For example, in software projects, developers may be required to interrupt the current development task, so that they can be assigned to other more urgent tasks. For exact algorithms that are used for solving the PRLP, one of the earliest linear program model for the PRLP was proposed by Son and Mattila (2004). Hariga and El-Sayegh (2011) considered cost of preemption in their optimization model for the PRLP. Nadjafi et al. (2013) proposed a branch and bound algorithm for the PRLP. For heuristic algorithms, Doulabi et al. (2011) designed a hybrid genetic algorithm equipped with a solution feasibility repair mechanism, and instances with up to 2000 activities have been solved.

In the above-mentioned research on the PRLP, Doulabi et al. (2011) and Nadjafi et al. (2013) compared the RLP and the PRLP and showed that considering preemption in resource leveling leads to more leveled resource usage. However, Doulabi et al. (2011) only reveals that heuristically solving the PRLP results in better

resource leveling performance compared to the RLP without considering preemption. It is still not unclear whether exactly solving the PRLP leads to the same conclusions. On the other hand, although Nadjafi et al. (2013) solve the PRLP with an exact algorithm. Their conclusions are obtained based on relatively small instances with up to 10 activities. Therefore, larger instances need to be further investigated. To fill this crucial gap, we discuss whether preemption leads to more leveled project resource usage based on mixed-integer linear programming (MILP) and large-scale computational experiments. The main contributions of this paper are as follows:

(1) We formulate a non-linear optimization model for the PRLP and linearized it to an MILP model, such that the resulting model can be solved by commercial solvers more efficiently.

(2) We conduct extensive computational experiments on a large number of test instances with up to 30 activities. To verify whether preemption leads to more leveled resource usage, we introduce two scenarios in the experiments: non-preemption and preemption.

(3) A case from a real-world construction project is further used to illustrate the benefits of preemptive resource leveling.

The remainder of the paper is organized as follows: In Section 2, we state the PRLP. Section 3 presents our MILP for the PRLP. Computational results are given in Section 4. A real-world project case study is provided in Section 5. The last section presents our conclusions.

2. The preemptive resource leveling problem

The PRLP is stated as follows. A project is represented by an acyclic activity-on-node network $G = (N, A)$. In the project network, the set of nodes $N = \{0, 1, 2, \dots, n, n + 1\}$ indicates activities. Activities 0 and $n + 1$ are dummy activities representing the start and end of the project, respectively. The set of directed arcs $A \subseteq N \times N$ indicates the precedence relations. The project is required to be finished no later than the deadline \bar{d} . Each non-dummy activity i has a fixed duration d_i . s_i denotes the start time of activity i . Both d_i and s_i are non-negative integers. Each non-dummy activity i requires r_{ik} units of resource k during execution. In addition, dummy activities have a duration of zero and do not consume any resources.

In project management practice, it is not uncommon to interrupt some activities such that resources can be re-allocated to other more critical activities. Therefore,

we assume that preemption is allowed in our problem. This means that an activity can be interrupted during execution. Without loss of generality, for activity i , we assume that the maximum number of preemption is $P_i = d_i - 1$, which means that activity i can be interrupted after each time unit. We use p to denote the number of preemption for activity i . It can be seen that $0 \leq p \leq P_i$: If $p = 0$, then activity i is executed without preemption; If $p = P_i$, then activity i is interrupted after each time unit.

If activity i is interrupted p times, then activity i can be viewed as $p + 1$ sub-activities and these sub-activities form the set $I = \{i_1, i_2, \dots, i_{p+1}\}$. For each sub-activity $i_q \in I (q = 1, 2, \dots, p + 1)$, its duration and start time are denoted as d_{iq} and s_{iq} , respectively.

The aim of the PRLP is to minimize the variations in resource utilization by specifying the starting time and duration of each sub-activity under the preemption environment, while satisfying the precedence relation constraints and the project deadline constraint.

3. Model formulation of the PRLP

Before presenting our model for the PRLP, we need to introduce a new conception: unit time network G' . G' is obtained by dividing each activity of G into sub-activities with duration being 1, and additional precedence relations are added to these sub-activities that belong to the same activity. This means that we assume that each activity in G is preempted every time unit. It is obvious that the network G' is equivalent to the original network G , since for any two sub-activities i_{q1} and i_{q2} that are from the same activity of G , there will be no interruption if the finish time of i_{q1} is equal to the start time of i_{q2} .

Based the unit time network G' , the non-linear optimization model of the PRLP can be conceptually formulated as follows:

$$\text{Minimize } f(\mathbf{s}, \mathbf{d}) = \sum_{k=1}^K \sum_{t=1}^{\bar{d}} c_k \cdot (u_{kt})^2 \tag{1}$$

Subject to

$$s_{0,1} = 0 \tag{2}$$

$$s_{n+1,1} \leq \bar{d} \tag{3}$$

$$s_{i,P_i+1} + d_{i,P_i+1} \leq s_{j,1} \quad \forall (i, j) \in A \tag{4}$$

$$s_{iq} + d_{iq} \leq s_{i,q+1} \quad \forall i \in N; \quad q = 1, 2, \dots, P_i \tag{5}$$

$$\sum_{i \in V_t} r_{ik} \leq u_{kt} \quad k = 1, 2, \dots, K; \quad t = 1, 2, \dots, \bar{d} \tag{6}$$

$$d_i = \sum_{q=1}^{P_i+1} d_{iq} \quad \forall i \in N \quad (7)$$

$$s_{iq}, d_{iq} \in N^+ \quad \forall i \in N; q = 1, 2, \dots, P_i + 1 \quad (8)$$

The objective function (1) minimizes the variations in resource utilization, where u_{kt} represents the usage of resource type k at each time unit t (Demeulemeester and Herroelen 2002). c_k is the weight of resource type k .

The constraint (2) ensures that the project is started at time zero. The constraint (3) makes sure that the project is completed no later than the pre-given deadline. The precedence relation constraints are given by (4), which means that the successor activities cannot be started until the last sub-activities of its predecessor activities have been finished. The constraints (5) indicate that the sub-activities of each activity should be executed sequentially. The constraints (6) are used to calculate the resource usages for a given resource type and time unit, where V_t denotes the set of activities that are in progress during time period t . The constraint (7) mean that, given an activity i , the sum of its sub-activities' duration should be equal to the duration of activity i . The constraints (8) make sure that the duration and start time of each sub-activity are non-negative integers.

It can be seen that objective function (1) is non-linear. In order to solve this problem more efficiently using commercial solver, we will linearize the objective function (1) and transform the model into an MILP (Rieck et al. 2012).

First, we introduce a binary variable y_{kth} that indicates whether the usage of resource k during time period t reaches h :

$$y_{kth} = \begin{cases} 1, & \text{If the usage of resource } k \text{ reaches } h \text{ during time period } t \\ 0, & \text{Otherwise} \end{cases} \quad (9)$$

Then, we use H_k to indicate the maximum usage of resource type k when each sub-activity starts at the earliest start time. The variable u_{kt} in the objective functions (1) and the auxiliary variable y_{kth} can then be linked by the following equation:

$$u_{kt} = \sum_{h=1}^{H_k} y_{kth} \quad (10)$$

In addition, we introduce a parameter $2h - 1$ that is equal to the difference between h^2 and $(h - 1)^2$ for each y_{kth} (Rieck et al. 2012). According to equations (9) and (10), the objective function (1) is linearized as:

$$f(\mathbf{s}, \mathbf{d}) = \sum_{k=1}^K c_k \sum_{t=1}^{\bar{d}} \sum_{h=1}^{H_k} (2h-1)y_{kth} \quad (11)$$

We introduce another binary variable x_{iqt} which indicates whether sub-activity i_q starts at time t :

$$x_{iqt} = \begin{cases} 1, & \text{If sub-activity } i_q \text{ starts at time } t \\ 0, & \text{Otherwise} \end{cases} \quad (12)$$

By using the binary variables x_{iqt} , the start time $s_{i,q}$ of each sub-activity can be calculated as:

$$s_{i,q} = \sum_{t \in w_{i,q}} tx_{iqt} \quad (13)$$

Where $w_{i,q} \in [es_{i,q}, ls_{i,q}]$ represents every possible start time of sub-activity i_q . $es_{i,q}$ and $ls_{i,q}$ indicate the earliest and latest start time of sub-activity i_q , respectively.

Based on equations (12) and (13), the constraints (2)-(5) can be linearized as follows:

$$\sum_{t \in w_{1,1}} tx_{0,1,t} = 0 \quad (14)$$

$$\sum_{t \in w_{n,1}} tx_{n+1,1,t} \leq \bar{d} \quad (15)$$

$$\sum_{t \in w_{j,1}} tx_{j,1,t} - \sum_{t \in w_{i,P_i+1}} tx_{i,P_i+1,t} \geq 1 \quad \forall (i,j) \in A \quad (16)$$

$$\sum_{t \in w_{i,q+1}} tx_{i,q+1,t} - \sum_{t \in w_{i,q}} tx_{iqt} \geq 1 \quad \forall i \in N; q = 1, 2, \dots, P_i \quad (17)$$

By using binary variables x_{iqt} and y_{kth} , the constraints (6) can be linearized as follows:

$$\sum_{i \in N} r_{ik} \sum_{q=1}^{P_i+1} x_{iqt} - \sum_{h=1}^{H_k} y_{kth} \leq 0 \quad k = 1, \dots, K; t = 1, \dots, \bar{d} \quad (18)$$

Finally, the resulting MILP model is shown as follows:

Minimize (11)

Subject to

(14)-(18)

$$\sum_{t \in w_{i,q}} x_{iqt} = 1 \quad \forall i \in N; q = 1, 2, \dots, P_i + 1 \quad (19)$$

$$x_{iqt} \in \{0,1\} \quad \forall i \in N; q = 1, 2, \dots, P_i + 1; t = 0, 1, \dots, d-1 \quad (20)$$

$$y_{kth} \in \{0,1\} \quad k = 1, \dots, K; t = 1, \dots, \bar{d}; h = 1, \dots, H_k \quad (21)$$

The constraints (19) mean that each sub-activity has only one start time. Constraints (20) - (21) make sure that both x_{iqt} and y_{kth} are binary variables.

The PRLP can be reduced to the RLP by allowing each activity to be interrupted 0 time. This means that the PRLP is a generalization of the NP-hard RLP (Neumann et al., 2003). Therefore, the PRLP is also NP-hard.

4. Computational experiments

We use CPLEX 12.1 to solve the proposed MILP model for the PRLP. Our computational experiments are conducted on a PC with an Intel i5 3.20 GHz CPU. The test instances and experimental setup are introduced in Section 4.1. The computational results are reported in Section 4.2. In Section 4.3, the impacts of different factors on the resource leveling are analyzed.

4.1. Experimental setup

The test instances in our experiments are generated by using *RanGen* (Demeulemeester et al. 2003), which can generate activity-on-node networks with different parameter settings. The following three parameters are adopted to characterize the generated instances in *RanGen*: the number of activities ($|N|$), the number of resource types (K), and the order strength (OS). OS indicates the density of the project network and is calculated as the ratio between the number of precedence relationships and the theoretical maximum number of precedence relationships in the project network (Li et al. 2018a).

As shown in Table 1, we specify 3 values for the number of activities and 3 values for the OS , respectively. The number of resource types is fixed at 4. For each combination of $|N|$ and OS , 90 instances are produced, resulting in $3 \times 3 \times 90 = 810$ instances in total.

Table 1. Parameter settings

Parameter	Value
Number of activities ($ N $)	10; 20; 30
Order strength (OS)	0.3; 0.5; 0.7
Number of resource types (K)	4
Weight of resource type k (c_k)	1

In addition, we specify two different deadlines for each instance: the tight project deadline ($\bar{d} = 1.0 \cdot ES_{n+1}$) and the loose project deadline ($\bar{d} = 1.2 \cdot ES_{n+1}$). ES_{n+1} is the length of the critical path that is obtained by the critical path method (Demeulemeester and Herroelen 2002).

4.2. Computational results

In order to verify whether preemption leads to more leveled resource usage, we introduce two scenarios for each instance: non-preemption and preemption. For the non-preemption (preemption) scenario, CPLEX is used to solve the corresponding MILP model for the RLP (PRLP). For each scenario, we impose a CPU limit of 600 seconds. This means that an optimal solution will be output if it can be found within 600 seconds. Otherwise, the best feasible solution is output. The following results are obtained by comparing the non-preemption and preemption scenarios.

First, we need to find the instances that can be solved optimally for both the non-preemption and preemption scenarios. After solving the 810 instances under two scenarios, the number of instances that are solved optimally for both scenarios is shown in Tables 2, 3 and 4 with different number of activities.

Observe the results in Tables 2, 3 and 4, we find that as the $|N|$ and the OS increase, the number of instances that can be solved optimally decreases. When the project size become large ($|N| = 20$ and $|N| = 30$), a tight project deadline leads to more instances that can be solved optimally. However, it is interesting that when the project size is small ($|N| = 10$), a loose project deadline leads to more instances that can be solved optimally. Intuitively, instances with a tight deadline mean that there are less feasible solutions, and these instances should be solved optimally more easily. Our results do not show this behavior for small size project instances. This may reflect that our model cannot handle small instances well.

Table 2. Number of optimal instances with $|N|=10$

		<i>OS</i>			
		0.3	0.5	0.7	Sum
\bar{d}	Tight	36	34	14	84
	Loose	42	41	42	125
	Sum	78	75	56	209

Table 3. Number of optimal instances with $|N|=20$

		<i>OS</i>			
		0.3	0.5	0.7	Sum
\bar{d}	Tight	27	21	15	63
	Loose	16	13	8	37
	Sum	43	34	23	100

Does Preemption Lead to More Leveled Resource Usage in Projects?
A Computational Study Based on Mixed-Integer Linear Programming

Table 4. Number of optimal instances with $|N|=30$

		<i>OS</i>			Sum
		0.3	0.5	0.7	
\bar{d}	Tight	18	10	14	42
	Loose	4	3	0	7
Sum		22	13	14	49

From Tables 2, 3 and 4, it can be seen that a total of 358 instances is solved optimally for both scenarios. Next, we will study how much improvement can be gained by incorporating preemption into the RLP. In order for a fair comparison, the following results are obtained on the 358 instances, which have optimal solutions for both the preemption and the non-preemption scenarios. We use average improvement (AI) as the performance metric:

$$AI = \frac{\sum_{i=1}^{358} \frac{O'_i - O_i}{O_i}}{358} \times 100\% \quad (22)$$

Where $O'_i(O_i)$ is the objective function value of instance i under the non-preemption (preemption) scenario. A larger AI value means that the more improvement is gained by incorporating preemption. The experimental results are shown in Table 5. It should be noted that, when $|N| = 30, OS = 0.7$ and deadline is loose, no optimal solution is found.

From Table 5, it is obvious that considering preemption improves the objective function value and results in more leveled resource usage. The largest improvement is 42.48%. Tables 5 reveals that the improvement of preemption with the tight deadline is more significant than the loose one. In addition, when the OS is high, more improvement is obtained.

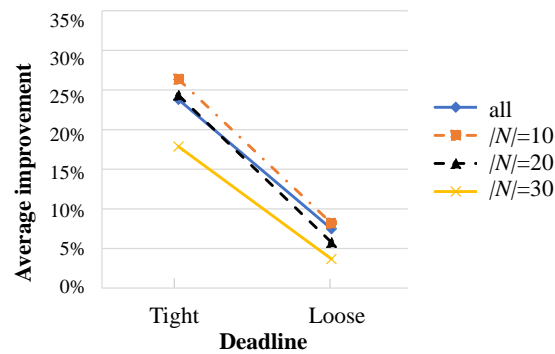
Table 5. The average improvement in resource leveling after incorporating preemption.

\bar{d}	$ N $	<i>OS</i>	ARD
Tight	10	0.3	19.56%
		0.5	26.91%
		0.7	42.48%
	20	0.3	14.17%
		0.5	29.19%
		0.7	35.88%

		0.3	10.03%
	30	0.5	22.25%
		0.7	24.80%
		0.3	9.80%
	10	0.5	7.21%
		0.7	7.55%
Loose		0.3	5.04%
	20	0.5	5.86%
		0.7	7.13%
		0.3	2.62%
	30	0.5	5.08%

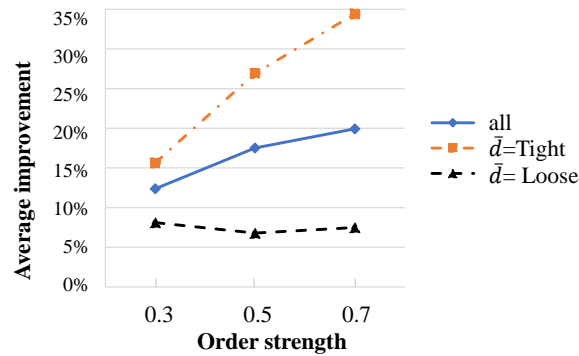
4.3. Sensitivity analysis

In this section, the influence of the project deadline, the number of activities and the order strength on the AI is further investigated. The influence of the three factors is shown in Figure 1. For any studied two factors, every sub-figure in Figure 1 shows the AI against one factor for all levels of the other factor. From Figure 1, we summarize the following three patterns:

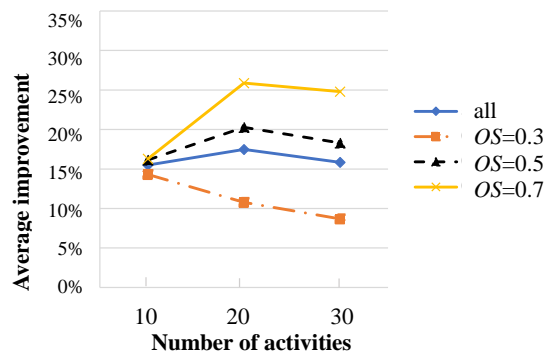


(a)

Does Preemption Lead to More Leveled Resource Usage in Projects?
A Computational Study Based on Mixed-Integer Linear Programming



(b)



(c)

Figure 1. The impact of different factors

First, when the project deadline is tight or the project size is large, the OS has a positive impact on the AI. The reason for this pattern is mainly that, with the increase of the OS , the resource use tends to be unlevelled, and the space for preemptive optimization will be increased.

Second, for a given level of the number of activities or the OS , the project deadline has a negative impact on the AI. The reason for this pattern may be that when the deadline is loose, there is less room for preemptive scheduling optimization.

Third, when the deadline is loose or the project size is small, the OS has a weak impact on the AI. This is because when the deadline is loose or the number of activities is small, the usage of the resources tends to be leveled, so there will be no significant change when the OS varies.

5. Case study

In this section, a real-world project is used to further verify our research question, i.e., whether preemption leads to more leveled resource usage.

The project employed in this case study is a tramways construction project that was performed in China (Ma et al. 2015). In this case study, we focus on human resource leveling. Therefore, we consider only one resource type, i.e., human resource. This project consists of 15 non-activities. The project network is shown in Figure 2, where the circles represent activities and the arcs indicate precedence relations. The number in each circle is the activity number with activities 0 and 16 being dummy. The duration and resource requirement of each activity is shown above the circle. The work content for non-dummy activities are presented in Table 6. It should be noted that in this project the time unit is 10 days. The weight c_1 of the single resource is equal to 1.

Table 6. Work content for each non-dummy activity

Activity	Work content
1	Large structures of earth
2,3,4,9	Building works
5,6,7,8,10,14,15	Scheme setting up
11,12	Drape wall manufacturing
13	Beautification and fixture manufacturing

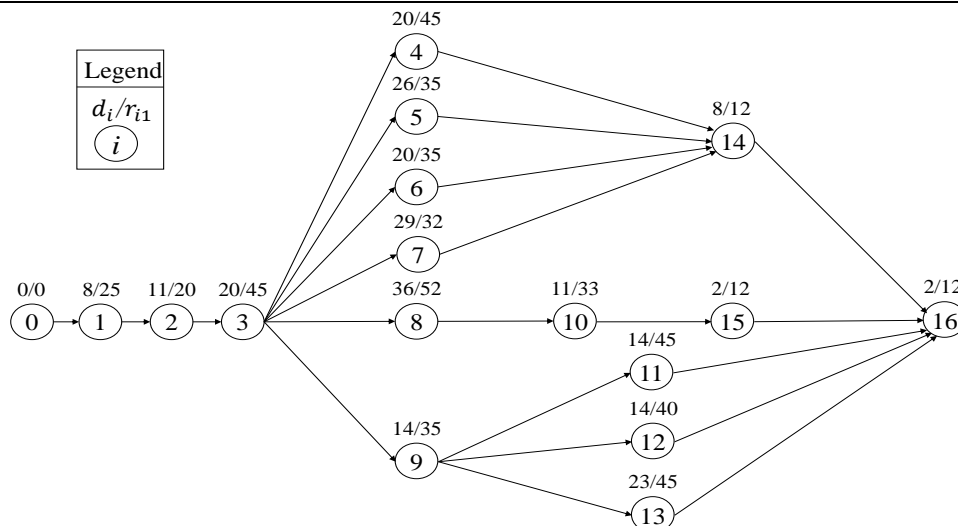


Figure 2. Project network of the case project

Does Preemption Lead to More Levelled Resource Usage in Projects?
A Computational Study Based on Mixed-Integer Linear Programming

We use CPLEX 12.1 to solve this project under two scenarios: non-preemption and preemption. The computational results are reported in Table 7. The column labeled " s_i " provides the start time of each activity when preemption is not allowed. The column labeled " $s_{iq}(d_{iq})$ " presents the start time and duration of each sub-activity under preemption. The objective function value for the non-preemption (preemption) scenario is 1160908 (1136368). The AI is 2.1%. This shows that the schedule obtained under preemption has a better performance compared to the non-preemption one. In other words, considering preemption leads to more leveled resource usage in this project.

Table 7. Project schedule under two scenarios

Activity	s_i : non-preemption	$s_{iq}(d_{iq})$: preemption
0	1	1
1	1	1(8)
2	9	9(11)
3	20	20(20)
4	73	40(2),45(2),48(3),52(11),83(1),87(1), 97(1)
5	40	40(5),46(2),50(3),56(1),58(1),61(2), 64(5),75(1),79(1),90(3),94(1),96(1)
6	40	41(1),47(1),49(1),57(1),67(1),69(1), 71(5),77(3),82(1),84(2),92(2),96(1)
7	64	53(3),57(3),69(2),75(2),80(19)
8	40	40(1),42(10),53(1), 55(1),60(1),62(2), 65(7),73(2),76(7),84(1),88(3)
9	59	41(6),48(1),51(2),54(1),56(2),59(2)
10	76	91(3),96(2),99(6)
11	93	64(1),71(1),85(2),89(1),98(9)
12	93	63(1),72(1),77(1),83(1),91(1),93(3), 98(5),126(1)
13	84	61(1),63(4),68(1),70(1),72(5),78(1), 80(2),86(3),95(1),103(4)
14	98	99(8)
15	91	105(2)

16	107	107
Obj. function value	1160908	1136368

6. Conclusions and future research

Preemption is often neglected in current research on project resource leveling. However, we have shown in this paper that, considering preemption in the RLP leads to more leveled resource usage.

We presented a non-linear optimization model for the PRLP and linearized it into an MILP model. Then CPLEX was used to obtain optimal solutions of the MILP model. We have performed comprehensive experiments on a set of randomly generated test instances under two scenarios (preemption and non-preemption) by solving the corresponding MILP models. Our computational results show that when the project deadline is tight or the order strength is high, incorporating preemption into the RLP gains the largest improvement. In addition, a real-world case study further reveals that preemption improves resource leveling in practical projects. For future research, it may be fruitful to develop heuristics for the large scale PRLP with more than 30 activities. Furthermore, it will also be interesting to study some extensions of the PRLP, such as the multi-mode PRLP and the multi-objective PRLP.

7. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation of China (Grant numbers 71602106, 71702097, 71801013), the Humanities and Social Sciences Foundation of the Ministry of Education of China (Grant numbers 15YJCZH077, 17YJC630238, 17YJC630177) and the Shandong Provincial Natural Science Foundation (Grant numbers ZR2016GQ07, ZR201709240250).

REFERENCES

- [1] Ballestín, F., Schwindt, C., Zimmermann, J., Ballestín, F. & Zimmermann, J. (2007), *Resource Leveling In Make-To-Order Production: Modeling and Heuristic Solution Method*. International Journal of Operations Research, 4(1), 50-62;
- [2] Bandelloni, M., Tucci, M. & Rinaldi, R. (1994), *Optimal Resource Leveling Using Non-Serial Dynamic Programming*. European Journal of Operational Research, 78(2), 162-177;
- [3] Burgess, A. R. & Killebrew, J. B. (1962), *Variation in Activity Level on a Cyclical Arrow Diagram*. Journal of Industrial Engineering, 13(2), 76-83;

- [4] **Christodoulou, S. E., Ellinas, G. & Michaelidou-Kamenou, A. (2010)**, *Minimum Moment Method for Resource Leveling Using Entropy Maximization*. Journal of Construction Engineering and Management, 136(5), 518-527;
- [5] **Demeulemeester, E. L. & Herroelen, W. S. (2002)**, *Project Scheduling: A Research Handbook*. New York: Kluwer Academic;
- [6] **Demeulemeester, E., Vanhoucke, M. & Herroelen, W. (2003)**, *RanGen: A Random Network Generator for Activity-on-the-Node Networks*. Journal of scheduling, 6(1), 17-38;
- [7] **Dobrovolskienė, N. & Tamošiūnienė, R. (2016)**, *Sustainability-oriented Financial Resource Allocation in a Project Portfolio through Multi-criteria Decision-making*. Sustainability, 8(5), 485;
- [8] **Doulabi, S. H. H., Seifi, A. & Shariat, S. Y. (2011)**, *An Efficient Hybrid Genetic Algorithm for Resource Leveling via Activity Splitting*. Journal of Construction Engineering and Management, 137(2), 137-146;
- [9] **Gather, T., Zimmermann, J. & Bartels, J. H. (2011)**, *Exact Methods for the Resource Leveling Problem*. Journal of Scheduling, 14(6), 557-569;
- [10] **Hariga, M. & El-Sayegh, S. M. (2011)**, *Cost Optimization Model for the Multi-resource Leveling Problem with Allowed Activity Splitting*. Journal of Construction Engineering and Management, 137(1), 56-64;
- [11] **Koulinas, G. K. & Anagnostopoulos, K. P. (2013)**, *A New Tabu Search-based Hyper-heuristic Algorithm for Solving Construction Leveling Problems with Limited Resource Availabilities*. Automation in Construction, 31(3), 169-175;
- [12] **Kreter, S., Rieck, J. & Zimmermann, J. (2014)**, *The Total Adjustment Cost Problem: Applications, Models, and Solution Algorithms*. Journal of Scheduling, 17(2), 145-160;
- [13] **Li, H. & Demeulemeester, E. (2016)**, *A Genetic Algorithm for the Robust Resource Leveling Problem*. Journal of Scheduling, 19(1), 43-60;
- [14] **Li, H. & Dong, X. (2018)**, *Multi-mode Resource Leveling in Projects with Mode-dependent Generalized Precedence Relations*. Expert Systems with Applications, 97, 193-204;
- [15] **Li, H., Xiong, L., Liu, Y. & Li, H. (2018a)**, *An Effective Genetic Algorithm for the Resource Leveling Problem with Generalised Precedence Relations*. International Journal of Production Research, 56(5), 2054-2075;
- [16] **Li, H., Xu, Z. & Wei, W. (2018b)**, *Bi-Objective Scheduling Optimization for Discrete Time/Cost Trade-Off in Projects*. Sustainability, 10(8), 2802;

- [17] **Li, H., Xu, Z. & Demeulemeester, E. (2014),** *Scheduling Policies for the Stochastic Resource Leveling Problem*. Journal of Construction Engineering and Management, 141(2), 04014072;
- [18] **Ma, G., Gu, L. & Li, N. (2015),** *Scenario-based Proactive Robust Optimization for Critical-chain Project Scheduling*. Journal of Construction Engineering and Management, 141(10), 04015030;
- [19] **Mehdizadeh, E. & Akbari, H. (2017),** *A Novel Vibration Damping Optimization Algorithm for Resource Constrained Multi-project Scheduling Problem*. *Economic Computation and Economic Cybernetics Studies and Research*, 51(2); ASE Publishing;
- [20] **Mehdizadeh, E. & Dadgar, S. N. (2014),** *Using Vibration Damping Optimization Algorithm for Resource Constraint Project Scheduling Problem with Weighted Earliness-Tardiness Penalties and Interval Due Dates*. *Economic Computation and Economic Cybernetics Studies and Research*, 48(1); ASE Publishing;
- [21] **Nadjafi, B. (2014),** *Using Grasp for Resource Availability Cost Problem with Time Dependent Resource Cost*. *Economic Computation and Economic Cybernetics Studies & Research*, 48(1), 201-215; ASE Publishing;
- [22] **Nadjafi, B., Khalaj, Z. & Mehdizadeh, E. (2013),** *A Branch and Bound Approach to Solve the Preemptive Resource Leveling Problem*. *International Journal of Manufacturing Engineering*, (1), 24-26;
- [23] **Neumann K., Schwindt C. & Zimmermann J. (2003),** *Project Scheduling with Time Windows and Scarce Resources: Temporal and Resource-constrained Project Scheduling with Regular and Nonregular Objective Functions*. Springer;
- [24] **Neumann, K. & Zimmermann, J. (1999),** *Resource Leveling for Projects with Schedule-dependent Time Windows*. *European Journal of Operational Research*, 117(3), 591-605;
- [25] **Ranjbar, M. (2013),** *A Path-relinking Metaheuristic for the Resource Leveling Problem*. *Journal of the Operational Research Society*, 64(7), 1071-1078;
- [26] **Rieck, J. & Gather, T. (2012).** *Mixed-integer Linear Programming for Resource Leveling Problems*. *European Journal of Operational Research*, 221(1), 27-37;
- [27] **Son, J. & Mattila, K. G. (2004),** *Binary Resource Leveling Model: Activity Splitting Allowed*. *Journal of Construction Engineering and Management*, 130(6), 887-894.