

The Use of Simulated Annealing and Genetic Algorithm in Solving Resource Leveling Problem in Multi-Project Mode

Seyed Mahdi Ebrahimi ^{a,*}, Mohammad Kananizadeh ^b

^a Master of Industrial Engineering, Science and Art University, Yazd, Iran

^b PhD. Student of Industrial Engineering of Yazd University, Yazd, Iran

ARTICLE INFO

Article history:

Received: 2022-10-25

Received in revised form: 2023-02-26

Accepted: 2023-10-09

Keywords:

Resource Leveling
Multi-project Mode
Genetic Algorithm
Simulated Annealing

ABSTRACT

Having a functionally efficient plan with regards to Resource Leveling is among the significant determining factors when it comes to reducing the costs of any project; especially, when multiple projects are carried out simultaneously, this notion proves to have even more importance. In the present article, after exploring the general concept and notion of Resource Leveling and Problem Modelling in a single project, the problem will be scrutinized in multi-project mode. To proceeding with doing so, a mathematical model is proposed whose objective is to minimize the changes in the levels of different resources which are used by all projects.

Given the fact that Resource Leveling Problem is an NP-hard problem and reaching the optimized solution is not generally possible, methods such as Genetic Algorithm and Simulated Annealing Algorithm are used by which to reach an approximate optimal result. Additionally, to elucidate the quality of the eventual answer which was achieved by the two aforementioned algorithms, a real-life example of resource planning in a software company has been used.

1. Introduction

The daily expansion of projects as well as the complications of activity management and scheduling with regards to the impact and significance of resources in the timely execution of activities and completion of the projects has directed the attention of researchers towards the way in which resources are used in this regard and context. Therefore, Resource Leveling has always been considered as one of the seminal and substantial issues in the field of Project Management. Nevertheless, the majority of researches in this filed have been directed towards Resource Leveling in a single project, while multiple projects are carried out simultaneously in many establishments during certain timespans. Accordingly, the subject matter of the present article deals with

Resource Leveling Problems in multi-project mode, thus reducing either the fluctuations in the need to resources or omitting the need to resources. The daily expansion of projects as well as the complications of activity management and scheduling with regards to the impact and significance of resources in the timely execution of activities and completion of the projects has directed the attention of researchers towards the way in which resources are used in this regard and context. Therefore, Resource Leveling has always been considered as one of the seminal and substantial issues in the field of Project Management. Nevertheless, the majority of researches in this filed have been directed towards Resource Leveling in a single project, while multiple projects are carried out simultaneously in many establishments during certain timespans. Accordingly, the subject matter of the present article deals with Resource Leveling Problems in multi-project mode, thus reducing either the fluctuations in the need to resources or omitting the need to resources.

* Corresponding author.

E-mail address: s.mahdi.e69@gmail.com

Resource management is an inherent element of project management; resource management guarantees that the project is completed on time and at cost as previously defined. In fact, project scheduling problems are one of the most important problems that performers deal while scheduling, chiefly when they need to achieve the most efficient resource usage without increasing the prescribed makespan of a project. The objective of the resource-leveling problem is to reduce resource fluctuation related costs (hiring and firing costs) without violating the project deadline [1].

2. Literature Review

In the related literature of this field, many researches have been carried out regarding the project scheduling with the objective of Resource Leveling and Resource Allocation. Khanzadi et al. (2016) proposes that, regarding the issue of scheduling projects, Resource Leveling is one of the most preferred topics among Construction Management researchers throughout the last couple of decades. And, in the present article, Colliding Bodies Optimization and Charged System Search were utilized. Moreover, two practical instances of such problems are solved in order to show the practicality and function of the algorithm [2].

Bandelloni et al. [3] have researched the notion of unlimited resources, proposing an optimized method for doing so. In this article, their objective was to schedule all the activities of the project on particular days between the floats/slacks, so that resource utilization is done in an ideal and preferable manner. Their method was adequate for small and medium problems. Khattab and Soyland [4] composed an article named "Limited-resource Allocation in Construction Projects". This issue is prevalent in many construction projects when there are different limitations to the resources of the contractor. Their article used a heuristic method for scheduling the activities based on Primavera software. Younis and Saad [5] proposed an algorithm for reaching an optimal solution to Resource Leveling problems. Their algorithm covers multi-resource modes. Abbasi and Garmsiri [6] explored Resource Leveling Problem in projects using Ant Colony Optimization method. "Resource Scheduling" is the title of an article by Gordon and Tulip [7]. In this research, their objective was to present a history, and review the technical and fundamental principles, of this science. Moreover, in order to aid the majority of readers in fathoming some of scheduling softwares, a brief history of some of such commercial softwares was also included.

In an article whose goal was a competent management of construction projects, Elawany et al. [8] realized the necessity of establishing a project schedule for resource utilization. Moreover, Zimmerman and Newman [9] carried out extensive and exhaustive researches about Resource Leveling; they firstly reviewed the previous accomplishments of this area and then proceeded with presenting procedures and processes for Resource Leveling.

Regarding the notion of unlimited-resource investment, Demeulemeester [10] has proposed a precise method for project scheduling problem with the purpose of minimizing the project cycle based on Branch & Bound Algorithm. This article includes an analysis of applied performance for projects with up to twenty activities and six resources.

Newman and Zimmerman [11] explored the resource-limited project scheduling problems using non-normal objective functions

in which normal time limitations, which impose minimum and maximum lateness between activities, are mentioned. Throughout the last number of years, a series of researches have been conducted about multi-project scheduling. The following examples are among such endeavors.

In his research, Shadkam (2021) discusses the fact that most researches in the field of Resource Leveling are only done in single-project mode while many organizations, such as project-oriented companies, carry out a number of projects simultaneously. Therefore, in a research using Cuckoo Optimization Algorithm (COA), which is inspired by the nature and is among the best and most powerful methods of evolutionary optimization, Shadkam compared the results achieved by solving the problem using COA, which is an approximate method, with the results from doing so using Branch and Bound, which is a precise method. Based on the outcome of such a comparison, it was concluded that, on smaller scales, using precise methods is more preferred. And, the more a problem's scale grows, the more COA is able to come up with proper result in shorter time and faster pace [12].

Kurtulus and Davis [13] carried out a research about multi-project scheduling which resulted in an article. In said article, they recommended the use of applied and experimental procedures for project scheduling problems. Dekro et al. [14] developed the use of Analysis Algorithm for solving resource-limited, multi-project problems. The analysis approach for solving resource-limited, multi-project scheduling problems proposes a computational approach which is also capable of reaching the solution in large and complex scheduling environments.

Lawrence and Morton [15] explored the resource-limited, multi-project scheduling problems using the function objective of minimizing weighted lateness costs. This notion usually exists in practice as well. Contractors, engineering firms, repair personnel, research & development teams and other similar organizations face this problem.

In a research, Altun et al. (2020) developed an optimization model using Mixed-integer Programming (MIP) to minimize the peak needs of various resources in numerous construction projects. The performance and functionality of the proposed model was tested with four case-study projects each of which had different project settings. The solutions which were achieved using the proposed method were then compared to the optimized solutions of traditional Leveling models in which the resource utilization is optimized for each project. The results separately showed the proposed method significantly improved the performance of the portfolio in comparison with the results of traditional methods based on separate leveling of projects [16].

Esperanza and Vercelli [17] proposed a model for multi-project problems without precedence which was based on a two-stage hierarchical analysis of scheduling and planning process. In this model, resources were categorized into two groups of renewable and nonrenewable. And, activities could be carried out based on the conditions related to different combinations of resource utilization, cost and time cycle. Eventually, these two researches proposed a group of Branch & Bound procedures with which to solve this model. Moreover, Lova et al [18] proposed a heuristic multiple-criteria method by means of which to improve resource allocation in multi-project scheduling.

A certain model for controlling projects was developed by Tsubakitani and Deckro [19]. In their article, they encoded the Shortest Activity from the Shortest Project (SASP) Rule for the scheduling of multi-project problems with more than 50 projects that

can have more than 100 activities. Fatemi Ghomi and Ashjari [20] also proposed a simulated model for multiple-project resource allocation. Sen Leu et al. [21] proposed resource Leveling problem in single-project mode for the purpose of planning construction projects, using the metaheuristic method of Genetic Algorithm for solving it. Kastor et al. [22] used the notion of Resource Leveling to introduce a solution to Resource-constrained Project Scheduling Problem (RCPSp), using two real-life example as case studies by which to elaborate on the proposed solution.

Resource Leveling Problem is indeed an NP-hard problem and reaching an optimized solution is not generally possible. Thus, to reach an approximate solution, a series of heuristic and metaheuristic methods are to be used. In the present article, a solution will be proposed which is based upon the metaheuristic methods of Genetic Algorithm and Simulated Annealing Algorithm which has not yet been carried out in multi-project mode.

3. Problem Modelling in Multi-Project Mode, Assuming the Existence of Multiple Resources

In this condition, the objective function of the problem can be depicted as follows.

$$z = \text{Min} \sum_{i=1}^{T_{i1}} \left(\sum_{j \in \bar{Y}} (r_{ij}^1)^2 + \sum_{j \in \bar{Y}} (r_{ij}^2)^2 + L + \sum_{j \in \bar{Y}} (r_{ij}^n)^2 \right) + \sum_{i=1}^{T_{i2}} \left(\sum_{j \in \bar{Y}} (r_{ij}^1)^2 + \sum_{j \in \bar{Y}} (r_{ij}^2)^2 + L + \sum_{j \in \bar{Y}} (r_{ij}^n)^2 \right) + K + \sum_{i=1}^{T_n} \left(\sum_{j \in \bar{Y}} (r_{ij}^1)^2 + \sum_{j \in \bar{Y}} (r_{ij}^2)^2 + L + \sum_{j \in \bar{Y}} (r_{ij}^n)^2 \right) \tag{1}$$

$$\begin{aligned} \text{st : } & \sum_{t=1}^{T_{t1}} r_t^1 = R_1 \\ & \sum_{t=1}^{T_{t1}} r_t^2 = R_2 \\ & \text{M} \\ & \sum_{t=1}^{T_{t1}} r_t^n = R_n \\ & \sum_{t=1}^{T_{t2}} r_t^1 = R_1 \\ & \sum_{t=1}^{T_{t2}} r_t^2 = R_2 \\ & \text{M} \\ & \sum_{t=1}^{T_{t2}} r_t^n = R_n \\ & \text{M} \\ & \sum_{t=1}^{T_{tn}} r_t^1 = R_n \\ & \sum_{i=1}^p R_i = Q \\ & r_{ij} \geq 0 \end{aligned} \tag{2}$$

In this function r_k^t is the number of resources needed from the kind on the date of t. P is the number of projects, and Q the sum of all the existing resources in the project organization. In the abovementioned objective function, the only goal is to minimize the fluctuations of resources while possessing said resources. And, the number of extra resources which are provided for the project(s) is not shown in the objective function. Therefore, another objective function must be defined in which the cost of extra resources on a daily basis is also considered. Also, the costs of employing and firing each one of the resources are different from one another; therefore, a new objective function must be defined, one which carries out two tasks.

- Minimizing the number of fluctuations of resources with respect to different costs of employing and firing in different timespan
- Minimizing the number of extra resources in such timespans

Given the fact that minimizing these two objectives is not generally possible, due to different and distinct costs and weights of each of them, they must first become of the same kind by multiplying them in the amount of their influence in the objective function, minimizing the new objective function afterwards. There exists a variety of factors which determine these numbers. Some of these factors include the type of the project, the type of resources utilized in the project, and the timespan during which it is carried out. Consequently, one cannot propose a single formula to calculate these influence coefficients

Given the mentioned factors and the particular project under study in the present article, during the interview with the original project manager and the CEO of the company in charge of the project, the cost of existence of each unit of extra resource per week is 100 units. Moreover, the constant cost of employing each new force is 80 unit; and, the cost caused by the consequence of firing each workforce is considered to be 70 units. According to the abovementioned information, the new, and final objective function for the problem is as follows.

$$\text{Min} \left\{ \sum_1^{n-1} 80(r'_{t+1} - r_t) [\text{Sign}(r'_{t+1} - r_t)] + \sum_1^{n-1} 70(r_t - r'_{t+1}) [1 - \text{Sign}(r'_{t+1} - r_t)] + \sum_1^n 100(r_t - r_t) \right\} \tag{3}$$

In this function, r_t is the number of resources in the basic solution (the minimum number of resources needed per week). And, r'_t is the number of resources gained using the algorithms. Given the objective function which is mentioned, when $(r'_{t+1} - r_t) > 0$, in other words when the number of resources gained in Week t+1 is more than the number of resources gained in Week t, we have had employed in that week. And, if $(r'_{t+1} - r_t) < 0$, then we have had fired in that week.

Therefore, in the first condition, the cost of each unit of difference is 80. And, in the second condition, the cost of each unit of difference is 70. This influences the formula by means of sign $(r'_{t+1} - r_t)$ function; in other words, if the variable inside the function is positive, the value function is 1. Otherwise, the function calculates the cost of employing and firing by returning the value to zero. It must be noted that, since the basic solution in this case is considered as the minimum number of resources needed, $(r'_t - r_t)$ is never negative and there is no need to define Sign function in this case.

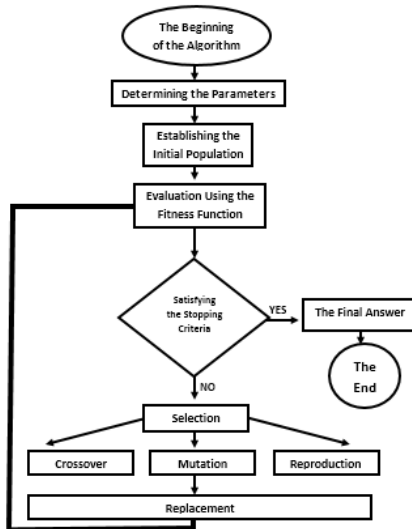
4. The Description of the Method of Achieving the Solution

A. Genetic Algorithm

Genetic Algorithm was first introduced in 1975 by John H. Holland. Numerous authors have used this algorithm in their articles. The main stages of executing the algorithm are as follows:

1. Establishing the initial population
2. Determining a fitness function by which to evaluate the answers
3. Selection strategy
4. Genetic operators

The general procedure of this algorithm is shown in the following diagram.



It must be noted that Genetic Algorithm deals with the coded/encoded form of parameters and variables of the problem, instead of working on them directly. The current problem is coded as such.

4	1	1	1	2	2	2	2	1	1
	1	3	6	0	0	0	1	8	9

As it was previously mentioned, the case that the present article studies is a project for implementation and execution of an Enterprise Performance Management (EPM) software which will later be scrutinized alongside the stages of its execution. However, it is vitally important to mention that the timespan for execution of said project was estimated to be about 20 weeks. Moreover, the

number of resources, software workforces, which the project needs during each two-week period has also been mentioned.

Accordingly, in the abovementioned code, the number of resources needed for each two-week period is placed in one of the genes. And, the following is the process of executing and implementing the proposed Genetic Algorithm. In this problem, the number of genes is considered to be equal to the number of days needed for completing the longest project. Next, all of the projects are considered as the initial sample or population. And, to organize the initial population, the well-known Burgess method in Resource Allocation is used.

After creating the initial sample, it is necessary to define a selection mechanism by which to choose a number of chromosomes from among the existing ones. And, with the help of genetic operators, the next generation is then created. One of the well-known methods of doing so is Roulette Wheel method which is used in the present article. Moreover, the fitness function of the problem is defined as follows in which t is the number of the gene and z is the value of the objective function in each stage.

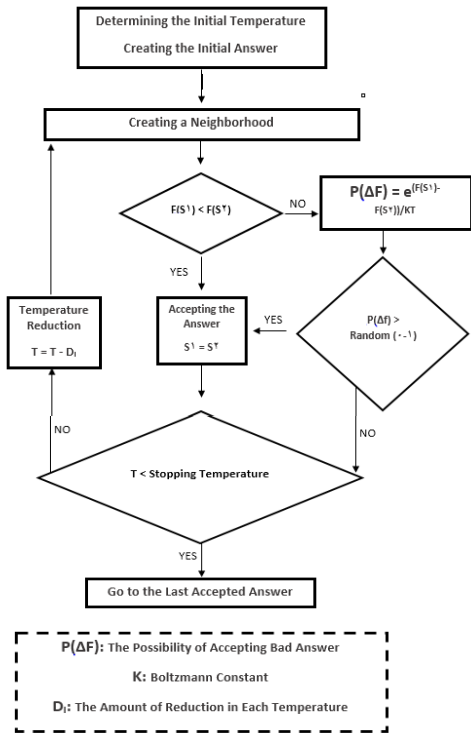
$$f(t) = \frac{1}{z} \tag{4}$$

The genetic operator used in this problem is mutation. And, the condition for the algorithm to finish is reaching the 1000th generation.

B. Simulated Annealing (SA)

Simulated Annealing (SA) algorithm is a well-known metaheuristic method for local search which is used for optimization problems. This method is more practical and functional for solving discrete optimization problems compared to continuous ones. It is important to note that this method provides a tool with which to escape the local optimum and reach the global optimum by accepting worse answers with a certain probability. The structured form of this algorithm was first proposed in 1983 by Kirkpatrick et al. (3). The main underlying idea of this method is based on the physical phenomenon of refrigeration whose objective is to reduce the temperature of the substance and direct it towards the lowest energy level. During this process, the substance's temperature gradually reduces from the initial temperature. And, upon reaching a final temperature, the cooling process stops. By doing so, the energy levels of the substance gradually decrease. And, this decreasing process is eventually stopped. Kirkpatrick considered this energy level as the value of the objective function in solving various problems.

In this section, the main goal is to compare the solution of the model proposed in Simulated Annealing algorithm with that of Genetic Algorithm in order to identify the more functional algorithm in doing so. It is self-evident that, due to the fact that the stages of the two algorithms are not identical, and also since there is difference in the way new population is created, it doesn't seem logical to compare the two algorithms with respect to the number of repetitions of the algorithms. Therefore, it has been tried to compare the results during equal timespans from the initiation of the two algorithms. The general procedure of Simulated Annealing algorithm is depicted in the following diagram.



Determining the initial value of the system’s temperature directly impacts whether or not answers are accepted. That is due to the fact that, in the case where the initial temperature is high, the system’s energy is very high as well. And, this is the considered to be a preferred condition when it comes to finding the best path towards reducing the temperature in reaching the stable state of the system. When the initial temperature is selected to have a low value, the possibility of accepting worse answer decreases; and, it is possible for the system to stay in local optimum. In 1983, White introduced the idea of initial temperature being equal to the standard deviation of system costs from average cost. Accordingly, the initial temperature is equal to the standard deviation of the objective values with respect to the number of times the program has been run in an unstable state. In this case, OBJ(j) is equal to the value of the objective function with respect to the answer of j.

$$Mean = \frac{\sum_{j=1}^N OBJ(j)}{N} \tag{5}$$

$$T_0 = Dev = \left(\frac{\sum_{j=1}^N (OBJ(j) - Mean)^2}{N-1} \right)^{1/2} \tag{6}$$

The relationship between temperature reduction and movement towards cooling the system is considered to be in the form of the following fiction.

$$T_i T_i = \alpha T_{i-1} \tag{7}$$

It must be noted that, given the dimensions of the problem, the coefficient α is considered to be 0.9 in this problem. The other matter is the definition of the stopping criterion. As it was mentioned before, given the differences of the two algorithms and the fact that it is illogical to compare them, they are executed at the same time in order to create comparability. Therefore, the stopping criterion is not defined in this algorithm. And, the algorithm continues as long as the defined time doesn't come to end.

In this method, the starting solution to the problem is also considered to be the minimum number of resources needed per day. And, with the process of finding neighborhoods, which will be discussed later, after each week of executing the project is selected, a unit is added to the necessary resources of each one of these week.

In order to reach the new points of the problem, a new neighborhood must be found based on the existing points. To find the new neighborhoods in solving the problem using Simulated Annealing, the difference between the numbers of resources in two consecutive weeks is calculated first. From among the resulting numbers, the two weeks which have the largest difference in the number of resources are then defined. And, in order to decrease this difference, a unit of resource is added to the one with smaller number of resources. By continuing this method, the goal is to reduce the difference between the number of resources in consecutive weeks and thus in the entirety of the project.

5. Exploring the Previous Calculations with the Help of a Real-life Example

The case under study in this article is Samix, a company which executes software solutions. And, the selected project is the project of installing and executing Enterprise Project Management (EPM) software. The process of implementing this software and creating the organization’s desired outputs takes approximately 11 weeks. And, the stages of doing so include the following phases.

- Analysis and Needs Assessment
- Installing and Executing the Software
- Software Configuration
- Creating the Necessary Reports
- Carrying Out a Pilot Project in the Client’s
- Workplace with the Supervision of the Contractor Company

Additionally, given the project plan, the number of resources needed per week are evident based on the implementation of the project. The important issues, however, is the type of the company that the client has. Given the fact that the organization wherein the software is implemented is in fact a holding company, this software must be separately implemented in each one of the holdings. And, the integration process is carried out eventually. As a result of his particular feature of this project, this implementation is not limited to a single project and consists of a number of subprojects. And, this is what satisfies the condition of multi-project nature of the problem.

It must be noted that, in order to compare the proposed algorithms, an optimal solution method by LINGO software is also used. Another important fact to mention is that the implementation of all the mentioned codes is done by a computer with 2.54 MHz CPU under C# software.

A. Optimal Solution

As it was mentioned before, LINGO 8 software is used. However, given the moderately sized dimensions of the problem, it was possible to calculate all the possible permutations for the number of resources. But, given the fact that the model and the problem were practical and functional enough for solving bigger problems, the model's code was implemented in this software as result of which the optimal solution was achieved. It must be noted that it took almost 3 hours for the software to reach the optimal solution. (The next parts of the article include more precise information about the amount of execution time needed for reaching the optimal solution.)

B. Genetic Algorithm

To solve the problem using Genetic Algorithm, the starting answer was considered to be equal to the minimum number of resources needed per two weeks. In each stage, one of the genes was randomly selected; and, a number was added to it. The objective function was then calculated. And, based on Roulette Wheel process, the new population was created.

C. Simulated Annealing Algorithm

To solve the problem using this method, similar to the previous method, the minimum number of resources needed was first considered as the starting answer. Next, the difference between each two consecutive weeks was calculated and allocated to each pair as a penalty. Afterwards, the largest amount of penalty was found. And, with respect to the values of resources in two consecutive weeks, one unit was added to the week which had the lower value. By doing so, it was tried to minimize the amount of resource fluctuation in different weeks, similar to the way in which employing extra forces was minimized. Moreover, in all three methods, the cost of each unit of extra resource is 100, the cost of each new employment is 80, and the cost of each firing is 70.

In the following table, the number of resources needed in each week, which is calculated by each algorithm, is included.

Table 1. The Number of Allocated Resources per Week by Each Algorithm

Week No.	1	2	3	4	5	6	7	8	9	10	11
Minimum Needed Resource	25	27	37	28	33	15	26	26	20	28	28
LINGO	25	27	37	30	33	21	26	26	24	28	28
Genetic Algorithm	25	27	37	28	33	23	26	26	20	28	28
Simulated Annealing Algorithm	26	27	37	28	33	22	26	26	22	28	28

The following table is a comparison between the three algorithms with respect to time and precision of the solution.

Table 2. A Comparison between Algorithms Used for Achieving the Solution

	Value of Objective Function	Solving Time				Error Percentage
		Millisecond(s)	Second(s)	Minute(s)	Hour(s)	
Optimal Solution	4590	2	44	23	89	0%
Genetic Algorithm	4790	0	0	7	329	4.35%
Simulated Annealing Algorithm	4760	0	0	4	244	3.70%

As it is evident, Simulated Annealing Algorithm and Genetic Algorithm both result in an answer which is close to that of the Optimal Solution. But, the Simulated Annealing Algorithm is faster and more precise. Additionally, it is considerably useful, especially when it comes to calculating the manpower for large number of weeks.

6. Conclusion

Resource Leveling Problem in multi-project mode is known to be a NP-complete problem. And, solving it using normal planning methods is not possible. Therefore, for the first time, the two metaheuristic methods of Genetic Algorithm and Simulated Annealing Algorithm were used.

To assess the quality of the resulting answers, the real-life information of a financial company was used. And, the optimal solution was achieved by means of exploring all of the possible cases and conditions using a software which is coded using C#. A rather long time, approximately 3 hours, is needed in order to achieve the optimal result. However, using the two aforementioned methods, a result is achieved which has small error and is acquired in a considerably smaller amount of time. With the help of the example used here, it can be concluded that, Simulated Annealing Algorithm creates a more precise answer and outcome during a smaller timespan. Therefore, using it is highly preferred, proper and appropriate for problems with larger size.

References

- [1] Nikoofal Sahl Abadi, N., Bagheri, M., & Assadi, M. (2018). Multiobjective model for solving resource-leveling problem with discounted cash flows. *International Transactions in Operational Research*, 25(6), 2009-2030.
- [2] Khazandi, M., Kaveh, A., Alipour, M., & Karimi Aghmiani, H. (2016). Application of CBO and CSS for resource allocation and resource leveling problem. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 40(1), 1-10.
- [3] Bandelloni, M., Tucci, M., & Rinaldi, R. (1994). Optimal resource leveling using non-serial dynamic programming. *European Journal of Operational Research*, 78(2), 162-177.

- [4] Khattab, M. M., & Soyland, K. (1996). Limited-resource allocation in construction projects. *Computers & industrial engineering*, 31(1-2), 229-232. [5] Younis, M. A., & Saad, B. (1996). Optimal resource leveling of multi-resource projects. *Computers & industrial engineering*, 31(1-2), 1-4.
- [5] Garmsiri, M., & Abassi, M. R. (2012). Resource leveling scheduling by an ant colony-based model. *Journal of Industrial Engineering International*, 8(1), 1-9.
- [6] Gordon, J., & Tulip, A. (1997). Resource scheduling. *International Journal of Project Management*, 15(6), 359-370.
- [7] Elwany, M. H., Korish, I. E., Barakat, M. A., & Hafez, S. M. (1998). Resource smoothening in repetitive projects. *Computers & industrial engineering*, 35(3-4), 415-418.
- [8] Neumann, K., Zimmermann, j. (1999), Resource leveling for project with schedule-dependent time windows. *European Journal of Operational Research*, Vol. 117, 1999, PP. 591-605.
- [9] Demeulemeester, E. (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, 41(10), 1590-1598.
- [10] Neumann, K., & Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, 127(2), 425-443.
- [11] Shadkam, E. (2021). The problem of resource leveling in multi-project mode by Cuckoo Optimization Algorithm. *JOURNAL OF CIVIL AND ENVIRONMENTAL ENGINEERING*, (), -. doi: 10.22034/jcee.2021.44761.2015.
- [12] Kurtulus, I. B. R. A. H. I. M. S., & Davis, E. W. (1982). Multi-project scheduling: Categorization of heuristic rules performance. *Management science*, 28(2), 161-172.
- [13] Deckro, R. F., Winkofsky, E. P., Hebert, J. E., & Gagnon, R. (1991). A decomposition approach to multi-project scheduling. *European Journal of Operational Research*, 51(1), 110-118.
- [14] Lawrence, S. R., & Morton, T. E. (1993). Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, 64(2), 168-187.
- [15] Altun, M., Sonmez, R., & Akcamete, A. (2020). A mixed integer programming method for multi-project resource leveling. *Journal of Construction Engineering, Management & Innovation*, 3(2), 131-40.
- [16] Speranza, M. G., & Vercellis, C. (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*, 64(2), 312-325.
- [17] Lova, A., Maroto, C., & Tormos, P. (2000). A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European journal of operational research*, 127(2), 408-424.
- [18] Tsubakitani, S., & Deckro, R. F. (1990). A heuristic for multi-project scheduling with limited resources in the housing industry. *European Journal of Operational Research*, 49(1), 80-91.
- [19] Ghomi, S. F., & Ashjari, B. (2002). A simulation model for multi-project resource allocation. *International Journal of Project Management*, 20(2), 127-130.
- [20] Leu, S. S., Yang, C. H., & Huang, J. C. (2000). Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. *Automation in construction*, 10(1), 27-41.
- [21] Kastor, A., & Sirakoulis, K. (2009). The effectiveness of resource levelling tools for resource constraint project scheduling problem. *International Journal of Project Management*, 27(5), 493-500.