

A genetic algorithm-based method for solving multi-mode resource-constrained project scheduling problem in uncertain environment

Mohammad Khalilzadeh^{a*}, Seyyed Hajir Hosseini^b and Reza Ghaeli^b

^a*CENTRUM Católica Graduate Business School, Pontificia Universidad Católica del Perú, Lima, Peru*

^b*Computing Studies and Information Systems Douglas College, Canada*

CHRONICLE

ABSTRACT

Article history:

Received: August 12 2019

Received in revised format:

December 27 2019

Accepted: January 31 2020

Available online:

January 31 2020

Keywords:

MRCPSP

Simulation

Uncertainty

Genetic Algorithm

Project scheduling models with resource constraints and multi-mode activities aims to create a schedule for carrying out activities considering precedence constraints and available resources in order to minimize the project duration. In the real world, we face uncertainty related to projects, where there are no historical data, hence, we should rely on the experts' judgements to estimate activity durations. For this purpose, in this paper, the 99-simulation method is used to deal with uncertainty. The exact mathematical programming model is presented in this paper and the hybrid algorithm based on Genetic Algorithm is used to solve this type of project scheduling problem which finds the near-optimal solution in a short computational time. Finally, the effectiveness of the proposed model is examined with a numerical example.

© 2020 by the authors; licensee Growing Science, Canada.

1. Introduction

Project scheduling problem with resource constraints (RCPSP) is among the widely used problems in the field of project management which involves finding a schedule for the project activities so that resource constraints are met. Two types of constraints exist in the RCPSP; the first type is associated with the relationships between activities which presents the sequential order of performing two activities, and the second type is related to resource constraints. In recent decades, many studies on project scheduling problems considering resources constraints have been conducted, taking the certain parameters into account. However, practical projects always are under uncertainty. Dealing with this uncertainty is usually accomplished with various approaches such as fuzzy numbers, probability, scenarios, etc. However, each project is normally unique and it is possible that a project includes activities which have not been accomplished so far in any other projects. In other words, there might be no historical data for these types of activities. Consequently, some certain approaches have low efficiency, as they often need historical data. In this study, the research gap is in such a way to consider the time periods as uncertain and resources as renewable in the resource-constrained project scheduling problems. In this paper, the activities can be performed in different modes, which have different durations and resource usage levels. Usually,

* Corresponding author.

E-mail address: khalilzadeh@pucep.edu.pe (M. Khalilzadeh)

an inverse relationship exists between resources consumption and activity duration. In other words, the more renewable resource usage per time, the less activity duration.

2. Literature review

The resource-constrained project scheduling problems has a broad scope and were examined by a many studies. Beşikci et al. (2015) proposed a multi-project scheduling problem where multiple resources were considered. In their model, resource sharing was not permitted during the project. Also, resources were considered as renewable and there was a capacity constraint for resources storage. Genetic Algorithm (GA) was used to solve this bi-level problem. Singh (2014) used a hybrid algorithm to solve the resource-constrained multi-project scheduling problem. The objective of the model was to minimize project duration as well as tardiness penalty. The penalty was due to different priorities for selecting projects. In some of the studies on these problems, multiple resources have been considered and different techniques have been used for solving them. Afshar-Najafi et al. (2013) considered the project scheduling problem with multiple and limited resources. They minimized the overall project duration by detecting the type of project and the resources constraint. A meta-heuristic method based on GA was implemented to solve the problem and it was compared with a branch and bound exact method. Moreover, Afshar-Nadjafi (2014) proposed a 0-1 integer programming to solve the resource-constrained project scheduling problem and Simulated Annealing (SA) algorithm was used to find near-optimal solution. His research showed that changing the type of resources to improve near-optimal solution was effective [4]. Bianco et al. (2016) developed a programming model for resource leveling problem with the limited timeframe and different activity durations. Their model consisted of a mixed binary linear programming model, a lower bound based on Lagrangian relaxation and the branch and bound methods. Aziz (2013) introduced strategic optimizer software for project scheduling problems with repetitive activities and multiple resources. He tried to find optimal resource consumption with minimization of project duration and cost as well as maximization of the net present value of the project. Cheng et al. (2015) incorporated the renewable and non-renewable resources into their proposed model. They studied a resource-constrained project scheduling problem considering renewable and non-renewable resources as well as the categorization of activities in both permitted and non-permitted modes. In their model, renewable resources were limited at any time. Also, a simple heuristic and an exact method were presented.

To obtain the near-optimal solution for these problems, different approaches were provided, as these problems are known as NP-hard problems (Blazewicz et al., 1983). Chen and Nie (2014) presented an optimization method for project scheduling problem with resource constraint. This model was based on Critical Differential Evaluation algorithm (CDE). This method showed that it was possible to better solve the objective function with the crisis map and loss function. Rahmani et al. (2015) applied a meta-heuristic approach to solve resource-constrained project scheduling problem. This method was based on Differential Evaluation meta-heuristic method. The results showed that this algorithm has been one of the best algorithms to solve RCPSP. Bakhshi et al. (2014) proposed a model to solve project scheduling problem when little information is available. Then, its performance was evaluated by alternative algorithms and its superiority was proven in an environment with little information. Peteghem and Vanhoucke (2014) considered project scheduling problem with limited resources for multiple simultaneous projects as well as non-deterministic environment. In this model, fuzzy numbers were used for uncertain data; the proposed model dynamically scheduled the project. Chen and Zhang (2016) solved resource-constrained project scheduling problem considering the durations of activities and the availabilities of resource as fuzzy numbers. The objective of their model was to maximize total free floats with two assumptions: the start times of some activities can be delayed and the lack of resources was permitted. Finally, the model robustness was shown by a numerical example. Tavana et al. (2014) used a multi-objective model to solve the project scheduling problem. In their model, three constraints of time, cost and quality were considered at the same time. Their model contained a multi-objective algorithm and a multi-objective mathematical model. Rezaeian et al. (2015) solved

the resource-constrained project scheduling problem by using a sub-population Genetic Algorithm. In this model, a two-objective function was used for robustness. Then, the model was implemented on several project scheduling problems of the PSPLIB and its performance was evaluated.

According to the authors' knowledge, there was no research on multi-mode resource-constrained project scheduling problems with uncertain activity durations. Since this model is much closer to the real world problems than the certain models, the importance of this study is noticeable. In this study, a hybrid Genetic Algorithm is used to solve the proposed model. To cope with uncertain activity durations, a simulation method based on the 99-method is applied. At the end, the performance of the model is examined with a numerical example.

3. Research methodology

3.1. Problem definition

In the first phase, the parameters and assumptions of the model are presented, which include a list of activities, the availability of renewable resources. The list of activities contains the number of activities, precedence relationships, distribution functions of activity durations, different modes of performing each activity and the resource consumption of each activity in any mode. At first, two dummy activities are defined as $i = 1$ and $i = N + 2$ with zero durations that denote the beginning and ending of the project, respectively. In this problem, the available amount of each resource k is constant. Each activity can be performed in several modes; more resource usage per time unit yields less activity duration and vice versa. In this paper, activity preemption is not allowed. In other words, if an activity starts in any mode, it cannot be interrupted until it is completed. The following symbols are used throughout this paper.

Table 1

Symbols used in the article

F_i Time of completing the activity i	S_i time of starting the activity i
λ_i the status of activity i in the activity list	λ list of activities
P_t Ongoing activities	Uncertain duration of each activity \tilde{d}
$r_{im_i k}$ the consumption of the activity i in the state of m_i from the resource k	k renewable resources
The project duration F_t	The number of activities N

3.2. The Model based on the mathematical expectation

This mathematical model is calculated based on the mathematical expectation. The goal of this model is to minimize the total duration of the project (F_t) so that the project completion time is equal to the start of the dummy activity $N + 2$, which is displayed as follows.

$$F_t = S_{N+2} \quad (1)$$

The mathematical model of the multimode resource-constrained project scheduling problem (MRCPSPP) under uncertain environment is as follows:

$$\min E[F_t] \quad (2)$$

subject to

$$s_i + d_{im_i} \leq s_j \quad \forall (i, j) \in A, \forall m_i \in M \quad (3)$$

$$\sum_{i \in P_t} r_{im_i k} \leq R_k, k = 1, 2, \dots, K, P_t = \{i | s_i \leq t \leq s_i + d_{im_i}\} \quad (4)$$

$$s_0 = 0 \quad (5)$$

$$s_i \in \text{int}^+ \quad \forall i \in N \quad (6)$$

The objective function (2) minimizes the project duration. Constraint (3) shows the precedence relationship between activities due to the lack of delay between them. Constraint (4) shows the capacity of resources utilization with respect to the activities (mi) mode. Constraints (5) forces the project to start at zero, and the last constraint (6) considers the activity starting time as natural number. Keep in mind that Pt represents that the activity is running. Because of uncertainty of activities duration, the project ending time is not fixed and it is changeable. In this case, we use 99-simulation method to estimate the expected time of completion. The mathematical model is as follows. In the model, ψ represents the reversed uncertain distribution of the activity duration. To solve the model, an algorithm based on genetic algorithm is used because according to the Blazewich et al. (1983), exact solution of such models requires a lot of time for large models.

3.3. GA-based Algorithm

A heuristic GA-based algorithm combined with 99-simulation method is used to solve the MRCPSP. In this paper, non-negative vector of $x = (x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_m)$ represents a chromosome where n is the number of activities, which is shown in Figure 1. In the vector x , the first n numbers show the sequence of activities and numbers from $n + 1$ to the last one depicts the modes of n activities (activities from 1 to n).

2	3	1	4	3	1	1	2
---	---	---	---	---	---	---	---

Fig. 1. Representation of a typical chromosome (a solution)

In Fig. 1, a sample chromosome is shown. In this chromosome, the first four boxes show the sequence of activities as 2.3.1.4 and the next four boxes denote the mode of each activity in which activity 2 is executed in mode 3, activities 3 and 1 are performed in mode 1, and activity 4 is accomplished in mode 2, respectively. In the following, we explain the two methods used to modify the new chromosome in order to satisfy the precedence relationships and uncertain simulation method based on 99-method to deal with the uncertain durations of the activities. It should be noted that in each of the operations that chromosomes enter, the chromosome is broken into two sub-chromosomes; the first one includes the sequence of the activities, and the second one contains the modes of the activities, each of the operations are performed separately for each chromosome and at the end of each operation these two chromosomes are joint to each other. In the chromosome modification method, an empty matrix is defined first. We start with the first chromosome, the first activity is selected if all its predecessors are in the new matrix, otherwise, the next activity is examined. This operation continues until the previous chromosome become empty. Thus, for each initial solution, we have a new answer which is modified based on precedence relationships. This modification operation of precedence relationships is conducted at all stages of forming a new chromosome, including initialization, crossover and mutation operations. Now, we perform uncertain simulation based on the 99-method to calculate the activity durations used for each activity at each iteration of Genetic Algorithm. This method examines the objective value of the new solution based on the fitness function after modifying all the precedence relationships of that activity. The method used in this article is based on the approach proposed by Jie et al. (2011) which was combined with the 99-method. We describe this method in the following.

– 99 – method based algorithm –

Given an activity list;

$r1 = \text{rand}(1,99);$

$$r2 = \text{rand}(1,99);$$

$$as = (a - d) + 2 * d * r1;$$

$$bs = (a + b) - as;$$

$$T = as + (bs - as) * r2;$$

In the 99-method algorithm, a , b , and d represent optimistic, most likely and pessimistic durations of each uncertain activity, respectively. This operation is repeated for all activities in all modes and the values obtained from the algorithm are used to deal with uncertain activity durations in projects. The main steps in the proposed GA-based algorithm are explained as follows.

Step 1: initialization

The first step involves the random generation of population size with a certain number. In this paper, different initial populations are used to get an understanding of the impact of changes in the population sizes on the problem solutions. To generate the initial population, an initial list of activities is randomly generated. Then, the precedence relationships are modified based on the aforementioned modification method.

Step 2: crossover operation

The second step involves the development of the population by generating new chromosome. In this paper, we use single point crossover operation. The crossover point is randomly selected. At each time of this crossover operation, two chromosomes are selected as parents by the roulette wheel selection method that provides more chance for the parents with better fitness to be selected. In this operation, two new children are generated for each pair of parents. The first part of the first new chromosome is inherited from mother chromosome and the second part of the first new chromosome is inherited from father chromosome. Also, the first part of the second new chromosome is inherited from father chromosome and second part of second new chromosome is inherited from mother chromosome. If the number of activities is odd, the middle activity sticks to the right side. In these new chromosomes, we review activities and remove duplicate activities. Then, the activities that are not exist in the chromosomes are added to the end of chromosome according to the number of activities. Now we have two solutions; first, the precedence relationships and then, the 99-method are examined to calculate the fitness value of two new solutions. In this paper, different parents are selected for crossover operation and their impacts on the solutions and computational times are shown.

Step 3: permutation operation

The mutation operation operates as follows: first, an activity is randomly selected for the mutation. This operation is used because GA does not get stuck in a local optimum and the algorithm can also cover other solution space. The permutation operation used in this study is two-point replacement operation. In this operation, a parent is randomly selected. Then, two points of it are randomly chosen, and the places of these two points are changed in a chromosome. The generated solution may violate the precedence relationships; hence, it is modified by the afore-mentioned precedence modification method. Finally, its fitness value is calculated and recorded by the 99-method.

Step 4: selection process

This stage is the last stage of each iteration. At this stage, all initial solutions and the solutions generated at each iteration are sorted in non-ascending order. Then, the first chromosomes are selected as the top ones and replaced with the other chromosomes. The selected chromosomes are used as parents for the next generation.

4. Numerical example

In this section, a numerical example is used to show the efficiency of the model. This problem consists of 30 activities and four types of renewable energy resources. Activity durations are shown as Beta distribution functions. In this paper, activities can be performed in different modes, instead of one mode. The durations of these new modes are about 10% more than the durations stated in the afore-mentioned example. Also, the resource usage in new modes is one unit less than the initial modes. The parameters of the example are shown in Table 2.

Table 2
The problem input parameters

Activity	Renewable resources required for each activity in each period										Prerequisite
	Mode 1	Mode 2	R11	R21	R31	R41	R12	R22	R32	R42	
1	0	0	0	0	0	0	0	0	0	0	-
2	(5.8.10)	(6.9.11)	4	0	0	0	3	0	0	0	1
3	(3.4.5)	(4.5.6)	10	0	0	0	9	0	0	0	1
4	(4.6.7)	(5.7.8)	0	0	0	3	0	0	0	2	1
5	(2.3.4)	(3.4.5)	3	0	0	0	2	0	0	0	4
6	(6.8.9)	(7.9.10)	0	0	0	8	0	0	0	7	2
7	(3.5.6)	(4.6.7)	4	0	0	0	3	0	0	0	3
8	(7.9.13)	(8.10.15)	0	2	0	0	0	1	0	0	3
9	(1.2.3)	(2.3.4)	6	0	0	0	5	0	0	0	4
10	(6.7.9)	(7.8.10)	0	0	0	2	0	0	0	1	4
11	(5.9.11)	(6.10.13)	0	5	0	0	0	4	0	0	2
12	(7.8.9)	(8.9.10)	0	7	0	0	0	6	0	0	8
13	(4.6.7)	(5.7.8)	4	0	0	0	3	0	0	0	3
14	(2.3.6)	(3.4.7)	0	8	0	0	0	7	0	0	9.12
15	(6.10.12)	(7.11.14)	3	0	0	0	2	0	0	0	2
16	(7.11.12)	(8.13.14)	0	0	0	5	0	0	0	4	10
17	(3.6.7)	(4.7.8)	0	0	0	8	0	0	0	7	13.14
18	(3.5.6)	(4.6.7)	0	0	0	7	0	0	0	6	13
19	(1.3.4)	(2.4.5)	0	2	0	0	0	1	0	0	8
20	(5.7.8)	(6.8.9)	0	10	0	0	0	9	0	0	5.11.18
21	(1.2.4)	(2.3.5)	0	0	0	6	0	0	0	5	16
22	(5.7.8)	(6.8.9)	2	0	0	0	1	0	0	0	16.17.18
23	(2.3.5)	(3.4.6)	3	0	0	0	2	0	0	0	20.22
24	(11,13,14)	(13,15,16)	0	9	0	0	0	8	0	0	19.23
25	(1.3.4)	(2.4.5)	4	0	0	0	3	0	0	0	10.15.20
26	(5.7.8)	(6.8.9)	0	0	4	0	0	0	3	0	11
27	(6.8.11)	(7.9.13)	0	0	0	7	0	0	0	6	8
28	(1.3.4)	(2.4.5)	0	8	0	0	0	7	0	0	21.27
29	(5.7.8)	(6.8.9)	0	7	0	0	0	6	0	0	19
30	(10.12.13)	(11.14.15)	0	7	0	0	0	6	0	0	6.24.25
31	(10.12.13)	(11.14.15)	0	0	2	0	0	0	1	0	26.28
32	0	0	0	0	0	0	0	0	0	0	29.30.31

The proposed algorithm has been coded with MATLAB version R2015a on a laptop with 2.20 GHz CPU and 8GB RAM. The model is solved with different parameters as follows (Table 3). In Table 3, pm and pc denote the percentage of chromosome selected from the parent chromosomes for the mutation and crossover operations respectively. The solutions obtained in the Table 3 are calculated by 50 runs of the Genetic Algorithm. According to the table, it is concluded that the population size has a significant impact on the computational time, but no considerable impact on the final solution. It can also be derived that higher mutation rate usually yields better solutions due to exploring more space. Subsequently, the quality of the solutions is examined in the following table.

Table 3
Displaying the near-optimal solutions

stage	population	pc	pm	Near-optimal solution																		Project duration	Time to solution				
1	20	0.8	0.4	1	3	13	18	8	12	27	19	29	7	4	5	10	9	14	17	2	11	26	20	6	63	7.8043	
				15	25	16	22	23	24	30	21	28	31	32	1	1	2	1	2	2	1	1	1	1			
				1	1	1	2	1	1	2	1	2	1	1	1	2	1	1	1	1	1	1	1	1	1		
2	30	0.7	0.4	1	3	8	4	5	19	29	12	13	18	9	14	7	17	27	10	16	22	21	28	2	63	11.75.27	
				15	11	26	20	31	23	24	6	25	30	32	1	1	1	2	1	2	2	2	1	1			
				1	1	1	1	1	1	1	1	2	1	1	1	2	2	2	1	1	2	2	1	2			
3	20	0.8	0.3	1	3	7	13	18	8	19	27	29	12	4	5	10	16	21	9	14	17	28	22	2	62	7.246	
				15	11	26	20	31	23	24	25	6	30	32	1	1	1	2	1	2	1	1	2	2			
				2	2	2	2	2	2	2	2	1	1	1	1	1	2	1	1	1	2	2	1	2			
4	30	0.8	0.4	1	3	4	5	10	16	13	7	18	8	12	19	21	2	15	11	20	25	29	9	14	62	12.1552	
				17	22	23	24	6	30	26	27	28	31	32	2	1	2	1	2	1	2	1	1	2	1		
				1	1	2	1	2	2	1	2	2	1	1	2	2	1	1	1	1	2	1	2	2	2		
5	30	0.7	0.2	1	3	8	19	12	4	5	29	9	14	13	17	18	2	27	7	15	10	16	21	22	63	10.1536	
				28	6	11	20	23	25	24	30	26	31	32	1	1	2	1	2	1	1	1	2	1			
				1	1	2	2	1	2	2	2	2	2	2	1	1	2	2	1	1	1	2	1	1	2		
6	30	0.7	0.3	1	3	2	11	26	15	8	12	19	29	6	13	18	4	5	20	9	14	17	27	10	65	11.0757	
				25	16	21	28	31	7	22	23	24	30	32	2	1	1	1	2	1	1	1	1	2			
				2	2	1	2	2	2	2	2	1	2	2	2	1	2	1	1	2	1	2	1	1	2		
7	30	0.6	0.3	1	3	13	7	18	4	10	9	5	8	27	12	14	17	2	6	11	20	16	21	22	63	9.8858	
				23	28	26	31	15	25	19	24	30	29	32	2	1	1	2	2	2	2	1	1	1			
				1	1	1	2	2	2	1	1	2	2	1	1	1	1	2	1	1	1	1	1	2	2		
8	20	0.8	0.3	1	4	5	3	7	8	12	10	13	18	9	14	17	16	22	21	2	6	11	26	20	61	7.3835	
				23	19	24	15	25	30	27	29	28	31	32	1	2	1	1	2	2	2	1	2	1			
				2	1	1	1	2	1	1	2	1	1	1	1	1	2	1	1	2	1	2	1	1			
9	30	0.6	0.2	1	3	4	5	9	8	19	27	12	14	13	18	17	29	2	15	11	26	20	6	10	65	8.8254	
				16	21	28	31	7	25	22	23	24	30	32	2	1	1	1	2	1	2	2	2	1			
				1	1	1	1	2	1	1	1	2	2	2	2	2	1	2	2	1	2	1	1	2			
10	30	0.6	0.3	1	3	13	18	8	12	7	4	27	5	9	14	17	2	15	11	26	20	6	10	25	65	9.8982	
				16	22	23	21	28	31	19	29	24	30	32	2	2	1	2	1	1	2	2	2	2			
				1	1	1	2	1	2	1	2	1	1	2	2	2	2	2	2	2	2	2	2	1	1	2	

Equations should be written with the same word processor used for the text, without hand written symbols. Equations must be centred and numbered sequentially with their numbers in parenthesis and justified to the right. Please refer to the following equation:

$$Error = \frac{(worst - best)}{Best}$$

Table 4
Model Performance in different modes

stage	population	pc	pm	The best	The worst	average	% of Error
1	30	0.6	0.2	62	66	64.4	%6.45
2	30	0.6	0.3	61	66	64.3	%8.19
3	30	0.7	0.4	62	66	64.6	%6.45
4	20	0.7	0.2	64	66	64.9	%3.12
5	20	0.8	0.3	61	65	63.5	%6.55
6	20	0.8	0.4	64	66	64.7	%3.12

Table 4 shows the results of the performance validation of the algorithm where in each case, the algorithm was run 10 times according to its parameters and the results are shown in Table 4. The reason for the relatively high error of the algorithm is to round the numbers of the activities time at the end of each stage of the simulation. However, due to the high speed of the algorithm and considering the uncertain duration of the activities, the model has an acceptable performance.

5. Conclusion

In this paper, a fuzzy multi-mode resource-constrained project scheduling model was proposed to deal with real world problems. The investigation of the model, according to various parameters of the genetic algorithm and activities uncertain time, represents acceptable performance of the model. Data for each activity is in such a way that there is no historical data for them and it is only determined using the expert opinion. In this paper, a heuristic algorithm is presented based on genetic algorithm in which, 99-method is used to deal with the uncertain parameters. Numerical example is also studied according to the data of the similar articles.

For future research, we can consider different distribution functions for the uncertain parameters and used the other distributions as well, which are closer to the real world. One can also consider interruption in the activities.

References

- Afshar-Nadjafi, B., Rahimi, A., & Karimi, H. (2013). A genetic algorithm for mode identity and the resource constrained project scheduling problem. *Scientia Iranica*, 20(3), 824-831.
- Afshar-Nadjafi, B. (2014). A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption. *Applied Computing and Informatics*.
- Aziz, R. F. (2013). Optimizing strategy software for repetitive construction projects within multi-mode resources. *Alexandria Engineering Journal*, 52(3), 373-385.
- Bakhshi, T., Sinharay, A., Sarkar, B., & Sanyal, S. K. (2015). A Semi Definite Programming Based Optimization Algorithm for Project Scheduling Problem. *Journal of The Institution of Engineers (India): Series C*, 96(2), 89-98.
- Beşikci, U., Bilge, Ü., & Ulusoy, G. (2015). Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research*, 240(1), 22-31.
- Bianco, L., Caramia, M., & Giordani, S. (2016). Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities. *OR Spectrum*, 38(2), 405-425.
- Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1), 11-24.
- Cheng, J., Fowler, J., Kempf, K., & Mason, S. (2015). Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. *Computers & Operations Research*, 53, 275-287.
- Chen, W., & Ni, X. (2014). Chaotic differential evolution algorithm for resource constrained project scheduling problem. *International Journal of Computing Science and Mathematics*, 5(1), 81-93.
- Chen, L., & Zhang, Z. (2016). Preemption resource-constrained project scheduling problems with fuzzy random duration and resource availabilities. *Journal of Industrial and Production Engineering*, 33(6), 373-382.
- Jie, H., Huaiyan, C., & Yun, C. (2011, August). Uncertainty evaluation using Monte Carlo method with MATLAB. In *Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on (Vol. 2, pp. 282-286)*. IEEE.
- Rahmani, N., Zeighami, V., & Akbari, R. (2015). A study on the performance of differential search algorithm for single mode resource constrained project scheduling problem. *Decision Science Letters*, 4(4), 537-550.
- Rezaeian, J., Soleimani, F., Mohaselafshary, S., & Arab, A. (2015). Using a meta-heuristic algorithm for solving the multi-mode resource-constrained project scheduling problem. *International Journal of Operational Research*, 24(1), 1-16.
- Singh, A. (2014). Resource Constrained Multi-Project Scheduling with Priority Rules & Analytic Hierarchy Process. *Procedia Engineering*, 69, 725-734.
- Tavana, M., Abtahi, A. R., & Khalili-Damghani, K. (2014). A new multi-objective multi-mode model for solving preemptive time-cost-quality trade-off project scheduling problems. *Expert Systems with Applications*, 41(4), 1830-1846.
- Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235(1), 62-72.

