Journal of Industrial and Systems Engineering
Vol. 10, Special Issue on Scheduling, pp. 92-118
Autumn (November) 2017

A multi-objective optimization model for project scheduling with time-varying resource requirements and capacities

Farhad Habibi[1], Farnaz Barzinpour[1]*, Seyed Jafar Sadjadi[1]

[1] *School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

farhadhabibi1993@gmail.com, barzinpour@iust.ac.ir, sjsadjadi@iust.ac.ir

Abstract

Proper and realistic scheduling is an important factor of success for every project. In reality, project scheduling often involves several objectives that must be realized simultaneously, and faces numerous uncertainties that may undermine the integrity of the devised schedule. Thus, the manner of dealing with such uncertainties is of particular importance for effective planning. A realistic schedule must also take account of the time-based variations in the capacity of renewable resources and the amount of resources needed to undertake the activities and the overall effect of such variations on the schedule. In this study, we propose a multi-objective project scheduling optimization model with time-varying resource requirements and capacities. This model, with the objectives of minimizing the project makespan, maximizing the schedule robustness, and maximizing the net present value, considers the interests of both project owner and contractor simultaneously. Two multi-objective solution algorithms, NSGA-II and MOPSO, are modified and adjusted with Taguchi method to be used for determination of the set of Pareto optimal solutions for the proposed problem. The proposed solution methods are evaluated by the use of fifteen problems of different sizes derived from Project Scheduling Problem Library (PSPLIB). Finally, solutions of the algorithms are evaluated in terms of five evaluation criteria. The comparisons show that NSGA-II yields better results than MOPSO algorithm. Also, we show that ignoring the time-based variations in consumption and availability of resources may lead to underestimation of project makespan and significant deviation from the optimal activity sequence.
Keywords: Resource-constrained project scheduling, Net Present Value (NPV), robust scheduling, resource variation, multi-objective optimization.

1- Introduction

In a Resource-Constrained Project Scheduling Problem (RCPSP), a series of activities to be undertaken continuously, according to their precedence relations, and using a limited set of renewable resources (such as human resources, machinery and equipment) must be scheduled with the purpose of achieving one or several particular objectives of several types of RCPSP, the one with the objective of minimizing the project completion time (or makespan) has attracted the most attention (see, e.g., Creemers, 2015, Delgoshaei et al., 2015, Wu et al., 2011, Chtourou and Haouari, 2008, Shi et al., 2010, Ghassemi-Tari and Olfat, 2007, Kumar and Arunagiri, 2010).

*Corresponding author
ISSN: 1735-8272,

But in addition, literature of RCPSP contains valuable publications on not only this objective but also other types of objective as well as alternative assumptions for activities, their precedence relations, and resources.
A summary of these works have been compiled and reviewed by Hartmann and Briskorn (2010). Another primary objective usually adopted for RCPSPs is the project cost minimization. In this approach, studies such as Liu and Zheng (2008) have focused entirely on minimizing the total cost of the project while others such as Berthaut et al. (2014) and Kang and Choi (2015) have attempted to establish a tradeoff between time and cost. Cost-based objectives can be expressed in terms of optimization of Net Present Value (NPV); an approach that was first introduced by(Russell, 1970) and later followed in works such as Sobel et al. (2009),Wiesemann et al. (2010) and Leyman and Vanhoucke (2016). Also, other articles such as Hsu and Kim (2005),Song et al. (2015) and Yuan et al. (2015) have studied the resource investment problem, where objective functions are based on renewable resources, while others such as Akkan et al. (2005), Demeulemeester et al. (1998) and Vanhoucke et al. (2002)have focused on the discrete time-cost tradeoff problem, where objective functions are based on nonrenewable resources. Real world conditions often compel the project managers to make their decision in line with multiple objectives. Thus, the overall objective is to optimize not only the makespan, but also revenue, cost, and resource leveling and even control the uncertainties involved in the project. Consequently, multi-objective project scheduling problem have been introduced to aid the project managers in making better decisions by taking multiple aspects of the project into consideration.

A major issue associated with scheduling problems is the presence of uncertainties and occurrence of unexpected events, which may disrupt and delay the work schedule. These disruptions may occur for various reasons like misestimating of duration of activities, lack of expected access to resources, addition or omission of an activity in the project network, or unexpected adverse weather conditions. Goldratt (1997)points out that a disrupted schedule increases the project expenses by causing the resource to remain idle, increasing the work-in-progress inventory, and intensifying the system atmosphere. In the project scheduling literature, lack of certainty has been addressed by approaches such as reactive scheduling, stochastic scheduling and fuzzy scheduling(see, e.g., Soltani and Haji, 2007). In addition to the above approaches, proactive (robust) scheduling has also proven useful in minimizing the effect of unexpected events on primary performance criteria such as project makespan. Such approach to scheduling has been utilized effectively by Lambrechts et al. (2011),Lamas and Demeulemeester (2016) and Palacio and Larrea (2017).

In the project scheduling literature, standard RCPSP has been the subject of many developments and modifications, for example, introduction of multiple operating modes for activities, generalized precedence relations, preempted activities, and also other approaches for generalizing the resource constraints. Another assumption of standard RCPSP is the uniformity of resource requirements and capacities over time, which undermines the practical applicability of the solutions; because resource availability is subject to variations caused by labor time offs and planned maintenance operations while demand for resources may also vary with the progress of activity. It is therefore important to incorporate such assumptions into project scheduling problems. However, Hartmann (2015) reports that the project scheduling literature contains too few works on time-varying resource requirements and capacities, and the concept is only mentioned in articles such as Bartusch et al. (1988), Sprecher (2012) and De Reyck et al. (1999).Table 1 lists some of the previous works on robust project scheduling.

Table 1. Review of RCPSP models with robustness considerations

Article	Model			Objective functions					Resource varying with time	Considering interests of owner and contractor Simultaneously	Solution method
	Single-objective	Multi-objective	Multi-stage	Makespan	Robustness	Cost	NPV	Quality			
Haouari and Al-Fawzan (2002)		X		X	X						MOTS
Al-Fawzan and Haouari (2005)		X		X	X						MOTS
Abbasi et al. (2006)		X		X	X						SA
Chtourou and Haouari (2008)			X	X	X						Two-stage-priority-rule-based algorithm
Lambrechts et al. (2008)	X				X						TS
Fallah et al. (2010)	X				X						Heuristic methods
Xiong et al. (2011)		X		X	X	X				X	MOGA
Lambrechts et al. (2011)	X				X						Heuristic methods
Artigues et al. (2013)	X				X						Heuristic methods
Gomes et al. (2014)		X		X	X						MOG, MOVNS, GMOVNS, MOVNS_I & PILS
Wang et al. (2014)		X		X	X	X		X		X	CGA
Xiong et al. (2014)		X		X	X	X				X	K-MOEA
Hao et al. (2014)		X		X	X						moEDA
Rezaeian et al. (2015)		X		X	X						SPGA
Lamas and Demeulemeester (2016)	X				X						Branch-and-cut method
Mogaadi and Chaar (2016)	X				X						Improved GA
Afshar-Nadjafi (2016)	X				X						A recursive heuristic
Proposed model		X		X	X		X		X	X	NSGA-II & MOPSO

As can be seen, existing articles on robust project scheduling, like other works on project scheduling, have assumed both the resources availability over a period and the resource consumption over the progress of activity to be uniform and time-invariant. Also, despite numerous studies in the field of multi-objective project scheduling, few studies have attempted to minimize the makespan and maximize the schedule robustness and Net Present Value (NPV) simultaneously. However, these objectives encompass the three most important decisions of project managers. Giving due attention to the cost-based objectives, which constitute one of the primary goals of the contractor, beside the makespan minimization as well as robustness maximization objectives, which constitute the primary goals of the owner, allows the interests of both parties to be incorporated into scheduling. Also, considering the time-varying resource requirements and capacities along with these objectives allows the resulting schedule to be more realistic.

In an attempt to make the project scheduling more realistic and enable the project managers to make better decisions in regard to project activities, this paper introduces a robust multi-objective optimization model for Resource-Constrained Project Scheduling Problem (RCPSP) based on discounted cash flows and time-varying resource requirements and capacities. It is assumed that activities are carried out by consuming renewable resources (with variable requirements and capacities).The objective functions of our proposed model are the minimization of makespan, maximization of schedule robustness, and maximization of Net Present Value (NPV).Given that RCPSP inherently belongs to the class of NP-hard problems and that solving multi-objective mathematical optimization models with metaheuristic algorithms result in more effective determination of Pareto optimal solutions, two multi-objective metaheuristic algorithms, namely NSGA-II and MOPSO, are used to solve this model. The proposed model and the solution methods are evaluated by fifteen problems of different sizes derived from standard data of Project Scheduling Problem Library (PSPLIB). After tuning the parameters of both algorithms with Taguchi method, solution methods are compared in terms of five different evaluation criteria.

In the rest of this paper: in section 2, problem formulations, notations, and description are provided; in section 3, solution approaches are discussed; in section 4, the effects of anticipated variations in resources on the schedule are explained, problem parameters are discussed, algorithm parameters are tuned by Taguchi method, the criteria to be used for evaluation of the algorithms are explained, and the results of evaluations are presented. Finally, Section 5 presents the conclusions.

2- Problem formulation

In the project scheduling problem, project consists of n activities that must be performed without preemption. Project structure is represented by an Activity-On-Node (AON)diagram in the form of the graph $G(V, E)$, where V is the set of vertices (or nodes) and E is the set of edges (or arcs), which represent respectively the activities and the associated precedence relations. These precedence relations are of finish-to-start type with zero time-lag. Graph nodes are named on a topological basis, in other words, the number with which an activity is labeled is greater than the label number given to all of its preceding activities. The nodes 1 and n of graph G are dummy nodes representing the start and end of project, meaning that they have a zero-long duration and need zero resources to be finished. All activities can be performed in only one way and each activity has a time-varying demand for resources over its progress. It is also assumed that resources necessary for the progress of activities are renewable resources with time-varying capacities (availabilities).

In the proposed model, the above assumptions are implemented by the following notations and definitions:

2-1- Sets

j	Set of activities
k	Set of resources
t	Set of time periods
G	Set of nodes and arcs on graph
V	Set of nodes

E	Set of arcs
P_j	Set of direct predecessors of activity j
S_j	Set of direct successors of activity j

2-2- Parameters

n	Number of activities
K	Number of type's renewable resources
d_j	Duration of activity j
R_{kt}	Availability of resource type k in time period t
r_{jkt}	Request for resource type k by activity j in process time t
CF_j^+	Positive cash flows for activity j
CF_j^-	Negative cash flows for activity j
NDS_j	Number of direct successors of activity j
α	Discounted rate
T	Project time window

2-3- Decision variables

ES_j	Earliest start time of activity j
EF_j	Earliest finish time of activity j
LS_j	Latest start time of activity j
LF_j	Latest finish time of activity j
FS_j	Free slack of activity j
C_j	Completion time of activity j
$C_{\max}$	Maximum completion time
NPV	Net present value
Ro	Scheduling robustness

$$x_{jt} = \begin{cases} 1 & \text{If activity } j \text{ is completed in time period } t \\ 0 & \text{Otherwise} \end{cases}$$

Among the decision variables of the model, the values of ES_j, EF_j, LS_j and LF_j depend on precedence relations, activities durations, and project time window, and since these values are calculated by the model, they are listed as decision variables. Also, FS_j and C_j are secondary decision variables which are defined for objective function calculations. Also, $C_{\max}, NPV$, and Ro are decision variables expressing the values of objective functions. The primary decision variable of this problem is x_{jt}, which expresses the completion time of project activities.

2-4- Objective functions

The objective functions of our proposed model are the minimization of makespan, maximization of schedule robustness, and maximization of Net Present Value (NPV). In the first objective function, the goal is to minimize the project makespan, or in other words, hasten the completion of activity n (like a standard RCPSP). This objective is expressed by equation 1:

$$C_{\max} = C_n = \max(\sum_{t=EF_j}^{LF_j} t.x_{jt}) \quad (1)$$

One method to increase the robustness of a schedule against disruptions is to maximize the free float of activities. With adequate float considered for the schedule, if, for any reason, some activities take more time than initially estimated, project can be finished on time without any need for addition funding. The probability of occurring disruptions for an activity is directly related to its duration and the amount of resources required, and the effect of disruption grows with the number of activities directly succeeding that activity; so in the objective function, r_{jrt}, d_j and NDS_j are used as weights to maximize the summation of free float of activities. The second objective function is expressed by Equation 2:

$$Ro = \sum_{j=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{d_j}(r_{jrt}.d_j.NDS_j)\,FS_j \quad (2)$$

In the real world, every project involves at least two parties: i) the client or the project owner, and ii) the contractor, who undertakes the project in practice. From the contractor's view, payments made by client act as revenue and payments the contractor make to procure materials and labor are the expenses. There are several different ways a client can pay its contractor, and method of payment may affect the project's NPV from the contractor's perspective. Ulusoy et al. (2001) have outlined several types of payment structures as follows:

- **Lump-Sum Payment (LSP) model:** In this model, which is one of the most common payment structures, contractor receives the total amount specified in contract when the project is finished. Assuming that the contractor pays the expenses of all activities within their respective earliest and latest start dates, NPV of the LSP model will be in the form of Equation 3:

$$NPV_{LSP} = \left(\sum_{j=1}^{n} CF_j^{+}\right)(1+\alpha)^{-C_{\max}} - \sum_{j=1}^{n}\sum_{t=ES_j}^{LS_j}\frac{CF_j^{-}.x_{jt}}{(1+\alpha)^{t}} \quad (3)$$

- **Payments at Event Occurrences (PEO) model:** In this method, payments will be made after completion of previously agreed-upon activities.

- **Payments of Activities (PAC) model:** In this method of payment, contractor receives the amount corresponding to each activity once the activity is finished. Assuming that the contractor pays the expenses of all activities within their respective earliest and latest start dates, NPV of the PAC model will be in the form of Equation 4:

$$NPV_{PAC} = \sum_{i=1}^{n}\frac{CF_i}{(1+\alpha)^{FT_i}} = \sum_{j=1}^{n} CF_j^{+}(1+\alpha)^{-C_j} - \sum_{j=1}^{n}\sum_{t=ES_j}^{LS_j}\frac{CF_j^{-}.x_{jt}}{(1+\alpha)^{t}} \quad (4)$$

- **Equal Time Intervals (ETI) model:** In this model, contractor receives H-1 payments at equal time intervals over the course of project and receives the H-th (final) payment once the project is finished. The NPV of the ETI model will be in the form of Equation 5:

$$NPV_{ETI} = \sum_{i=1}^{H-1}\frac{CF_i}{(1+\alpha)^{t_i}} + \frac{CF_H}{(1+\alpha)^{C_{\max}}} = \sum_{i=1}^{H-1}\frac{\left(\sum_{j=1}^{n} CF_j^{+}\right)(1+\alpha)^{-t_i}}{H} + \frac{CF_H}{(1+\alpha)^{C_{\max}}} - \sum_{j=1}^{n}\sum_{t=ES_j}^{LS_j}\frac{CF_j^{-}.x_{jt}}{(1+\alpha)^{t}} \quad (5)$$

- **Progress Payment (PP) model:** In the final payment method known as the Progress Payment model, contractor receives regular payments at certain time intervals over the course of the project. For example, payments may be made at the end of each month based on the work carried out over that duration plus a previously agreed-upon rate acting as the contractor's profit. The difference between the ETI and PP models is that in the PP model, the number of payments is not known in advance.

In the LSP model, maximization of NPV is equivalent to minimization of $C_{\max}$. In the PEO model, the set of nodes at which payments will be made is known, so this has no significant effect on the schedule of activities. In the ETI model, payments will be made in H installments and H is known. So with the reduction of $C_{\max}$, value of the third objective function will increase. Thus, given that the first objective function of the model seeks to minimize the makespan ($C_{\max}$), we use the PAC model in this objective function to maximize the NPV. As a result, the first objective function is expressed with equation 4.

2-5-Proposed model

Hence, our proposed mixed-integer optimization model is as follows:

$$\text{Minimize } C_{\max} \tag{6}$$

$$\text{Maximize } Ro=\sum_{j=1}^{n}\sum_{k=1}^{K}\sum_{t=1}^{d_j}(r_{jrt}.d_j.NDS_j)\,FS_j \tag{7}$$

$$\text{Maximize } NPV_{PAC}=\sum_{i=1}^{n}\frac{CF_i}{\left(1+\alpha\right)^{FT_i}}=\sum_{j=1}^{n}CF_j^{+}\,(1+\alpha)^{-C_j}-\sum_{j=1}^{n}\sum_{t=ES_j}^{LS_j}\frac{CF_j^{-}.x_{jt}}{\left(1+\alpha\right)^{t}} \tag{8}$$

Subject to:

$$ES_1=0 \tag{9}$$

$$EF_j=ES_j+d_j \qquad \forall j=1,2,...,n \tag{10}$$

$$ES_j=Max\left\{EF_i\right\} \qquad \forall i\in P_j\,;\ \forall j=1,2,...,n \tag{11}$$

$$LS_j=LF_j-d_j \qquad \forall j=1,2,...,n \tag{12}$$

$$LF_j=Min\left\{LS_i\right\} \qquad \forall i\in S_j\,;\ \forall j=1,2,...,n \tag{13}$$

$$LF_n=T \tag{14}$$

$$FS_j=LF_j-EF_j \qquad \forall j=1,2,...,n \tag{15}$$

$$\sum_{t=EF_i}^{LF_i}t.x_{it}\le\sum_{t=EF_j}^{LF_j}\left(t-d_j\right)x_{jt} \qquad \forall j=1,2,...,n\ ;\ \forall i\in P_j \tag{16}$$

$$\sum_{t=EF_j}^{LF_j}x_{jt}=1 \qquad \forall j=1,2,...,n \tag{17}$$

$$C_j=\sum_{t=EF_j}^{LF_j}t.x_{jt} \qquad \forall j=1,2,...,n \tag{18}$$

$$C_{\max}\ge C_j \qquad \forall j=1,2,...,n \tag{19}$$

$$T\le\sum_{j=1}^{n}d_j \tag{20}$$

$$C_{\max}\le T \tag{21}$$

$$\sum_{j=1}^{n} r_{jkt} \sum_{b=t}^{t+d_j-1} x_{jb} \leq R_{kt} \qquad \forall k = 1,2,...,K \; ; \; \forall t = 1,2,...,T - d_j + 1 \qquad (22)$$

$$x_{jt} = \{0,1\} \qquad \forall j = 1,2,...,n \; ; \; \forall t = 1,2,...,T \qquad (23)$$

The above model utilizes the objective functions described in the previous section as equations 6, 7 and 8. Constraints 9 to 14 are the formulations tasked with calculation of earliest and latest finishing time of all activities. Constraint 15 calculates the free floating time of activity j. Constraint 16 expresses the precedence relations between the project activities. Constraint 17 states that each activity must only have one start and one finish time, and once started must progress without any preemption until it is finished. Constraint 18 calculates the completion time of activity j. Constraint 19 calculates the value of $C_{\max}$. Constraint 20 determines the project's time window and Constraint 21 states that T (time window) is an upper bound for $C_{\max}$. Constraint 22 ensures that in the presence of sufficient resources in a period, activity j starts at the time b. Finally, constraint 23 defines the domain of decision variables.

3- Solution approach

3-1- Solution representation

When developing a metaheuristic algorithm, one of the notable issues is how to represent the solution in a way that satisfactory performance in the search space would be achievable. In the case of project scheduling problem, solution representation can vary depending on the involved decision variables. In this paper, solution is represented by a chromosome consisting of a vector of feasible permutation reflecting the sequence of activities. This representation allows the start and finish times of activities to be easily determined based on their precedence relations, required resources, and available resources in each period of planning. The proposed model is to be solved with continuous solution algorithms, so permutation of activities is determined based on the order of obtained numbers, or in other words, based on the use of random key strategy. Figure 2 illustrates some of the feasible activity sequences for the project with precedence network of figure 1.

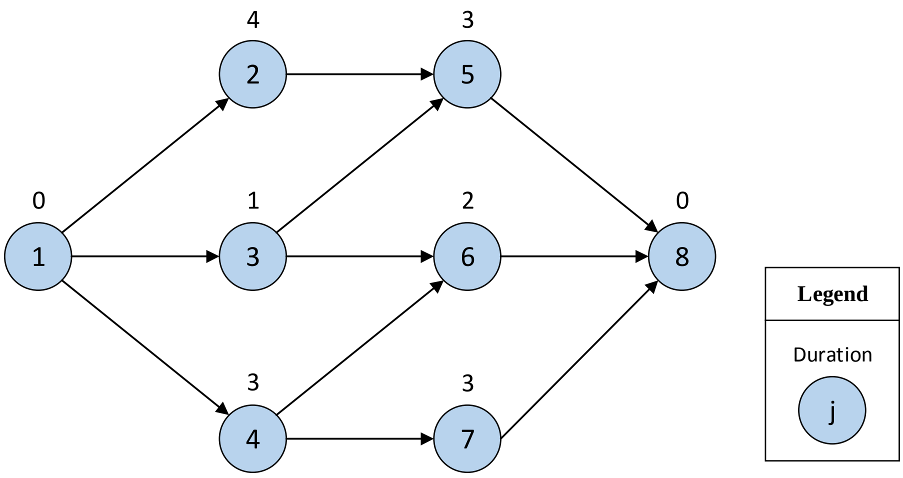

Figure 1. Precedence network of a problem

1	4	7	3	2	5	6	8
1	3	4	7	6	2	5	8
1	4	2	3	5	6	7	8
1	4	3	6	2	5	7	8
1	2	4	7	3	5	6	8

Figure 2. Some of the feasible activity sequences for the project with precedence network of figure 1

3.2. Solving methods

3.2.1. Non-dominated Sorting Genetic Algorithm II

Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a multi-objective evolutionary algorithm introduced by Deb et al. (2002)as an improved version of the original NSGA. NSGA-II sorts the population of parents and offspring with an elitist strategy and improves the diversity of solutions by a mechanism that is based on crowding distance operator instead of niched operators. Thank to these features, NSGA-II has become well-known as a reliable and suitable multi-objective genetic algorithm and has found extensive applications in many fields.

- **The Proposed NSGA-II**

In this paper, we attempt to improve the efficiency of NSGA-II for our purpose by using Arithmetic crossover and Gaussian mutation in the production of new population. In the Arithmetic crossover, the parents $x_1$and $x_2$having an equal number of elements will be randomly selected, and then, the vector α with the same number of elements as the parents will be used to produce offspring using equations 24 and 25.

$$y_{1i} = \alpha_i . x_{1i} + (1 - \alpha_i) x_{2i} \qquad 0 \le \alpha_i \le 1 \qquad (24)$$

$$y_{2i} = \alpha_i . x_{2i} + (1 - \alpha_i) x_{1i} \qquad 0 \le \alpha_i \le 1 \qquad (25)$$

In the Gaussian mutation, a certain number of chromosomes (this number is an adjustable algorithm parameter) will be randomly chosen; then μ percent of genes of the selected chromosomes will be subjected to a mutation with standard deviation δusing equation 26. In cases where mutated gene violates a defined range, its value will be set equal to the corresponding limit of that range.

$$x_i^{'} = x_i + \delta\left(Rand\left[0,1\right]\right) \qquad (26)$$

As shown in equation 27, the value of δ is equaled to a coefficient (β) of the gene's variation range.

$$\delta = \beta\left(Var^{\max}(x) - Var^{\min}(\mathrm{x})\right) \qquad 0 \le \beta \le 1 \qquad (27)$$

Finally, once offspring are produced and mutated, all chromosomes will be sorted in terms of the rank and crowding distance value obtained according to the objectives, and the fittest chromosomes will be selected to form the new generation.

3.2.2. Multi-Objective Particle Swarm Optimization Algorithm

Multi-Objective Particle Swarm Optimization (MOPSO) algorithm was first introduced by Coello et al. (2004) as an extension of PSO for solving multi-objective problems. Unlike PSO, this algorithm utilizes a concept known as "repository" or "hall of fame" for storing non-dominated particles and Pareto front. In MOPSO, each particle moves toward a member of the repository known as the leader. In other words, in MOPSO, the leader chosen from the repository replaces the global best (Gbest) used in PSO.

- **The Proposed MOPSO**

In this paper, MOPSO algorithm is also used to solve the proposed model. To solve the problem, as suggested by Coello et al. (2004), first an initial population will be created at random, then the best individual experience of each particle will be determined, and non-dominated members will be identified and stored in the repository. Each particle should select a member of the repository as the leader and move according to that leader. In multi-objective optimization algorithms, dispersion of the points in the Pareto front represents the strengths of solution, so we partition the objective space into a number of cells divided by gridlines; then utilizing Boltzmann method, we use equation 28 to assign each cell with a selection probability, and then select a cell and eventually a leader by using a roulette wheel mechanism. According to equation 28, cells with fewer Pareto points have a higher chance of being selected, thus their members have higher chance of being selected as a leader. This mechanism ensures satisfactory dispersion in the Pareto front.

$$p_i = \frac{e^{-\beta . n_i}}{\sum_j e^{-\beta . n_j}} \tag{28}$$

In equation 28, p_i is the probability of selecting cell i, n_i is the number of members in cell i, and β is the leader selection pressure parameter. Once the leader is chosen and position and velocity of every particle are updated, the best individual experience of each particle will be updated and the new non-dominated members will be added to the repository. In this step, some of the existing members of the repository may be dominated and thus replaced by the new members. The repository can store a limited number of members, so there may be not enough room to store new members. In this case, algorithm will utilize a mechanism similar to the approach used for selecting the leader to remove some of the existing members of the repository. In this mechanism, equation 29 is used to assign each cell with a selection probability and then a roulette wheel mechanism is used to select a cell and eventually the member to be deleted. In equation 29, cells with higher number of Pareto points have a higher chance of being selected. This mechanism also ensures better dispersion in the Pareto front.

$$q_i = \frac{e^{\gamma . n_i}}{\sum_j e^{\gamma . n_j}} \tag{29}$$

In equation 29, q_i is the probability of selecting cell i, n_i is the number of members in cell i, and γ is the deletion selection pressure parameter. The entire process, from selecting the leaders to updating the repository will be repeated until the desired stop condition is satisfied.

4- Computational results

To demonstrate the effect of time-varying resource requirements and capacities, the main contribution of the proposed model, on the project scheduling, we assume a small project with 6 real (non-dummy) activities, 2 renewable resources, and a precedence network shown in figure 1. Because of certain reasons (e.g. labor time offs and planned maintenance operations), the amount of available renewable resources (R_{kt}) varies over the time period as shown in table 2.The amount of resources needed to carry out activities (r_{jkt}) also varies over processing time as shown in table 3.With makespan minimization serving as the objective, the project duration will be 11 days and the total amount of resources to be consumed in each time period will be as shown in figure 3.

Table 2. Amount of available renewable resources over each time period (R_{kt})

Resource type	Period															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	10	12	12	11	10	6	9	8	8	10	12	11	12	10	10	9
2	9	9	8	7	7	8	9	9	7	8	8	6	8	9	10	6

Table 3. Amount of resources needed for each activity over the course of its progress (r_{jkt})

Activity	Resource type	Period			
		1	**2**	**3**	**4**
1	**1**	--	--	--	--
	2	--	--	--	--
2	**1**	4	4	3	5
	2	3	1	0	4
3	**1**	7	--	--	--
	2	8	--	--	--
4	**1**	5	4	6	--
	2	5	2	2	--
5	**1**	3	2	4	--
	2	7	5	9	--
6	**1**	5	5	--	--
	2	1	1	--	--
7	**1**	6	5	7	--
	2	4	3	5	--
8	**1**	--	--	--	--
	2	--	--	--	--

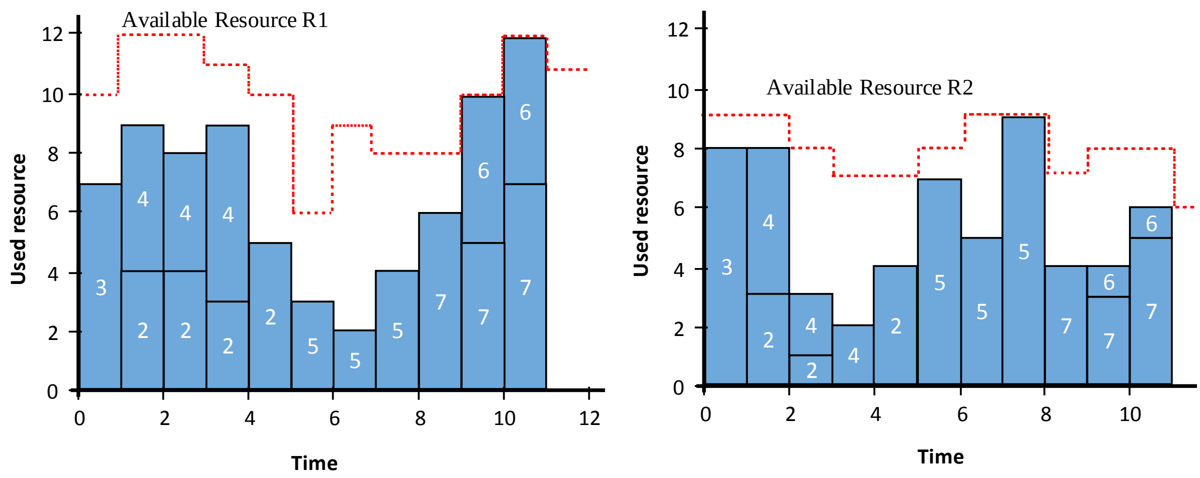

Figure 3. Total amount of consumed resources in each period when resource requirements and availability is variable

If we ignore the time-based variations of resources, or in other words, assume the amount of resource to be consumed (r_{jkt}) and to be available (R_{kt}) during any given time period to be fixed over that period (i.e. to be equal to the average of their time-varying counterparts), the result will be a project with precedence network shown in figure 4. In this case, with makespan minimization considered as the objective, the project duration will be 10 days and the total amount of resources to be consumed in each time period will be as shown in figure 5.

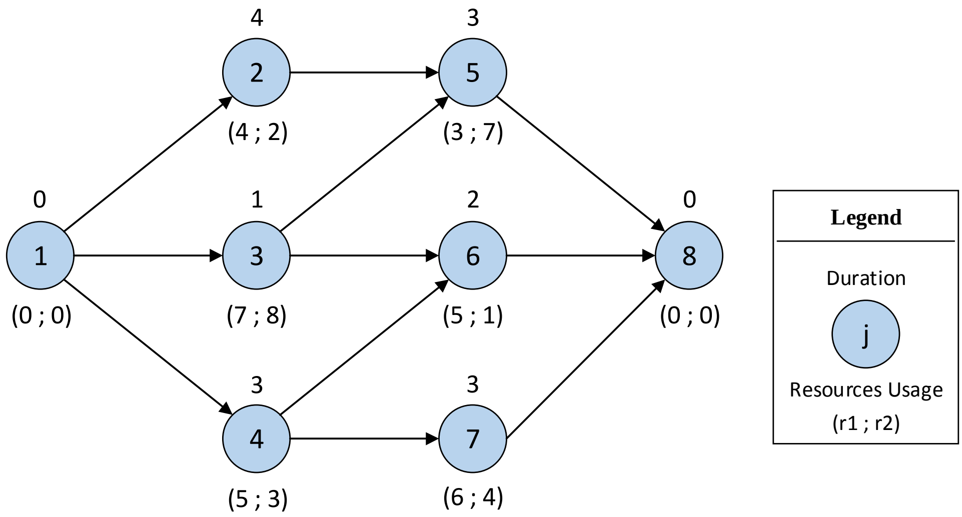

Figure 4. Precedence network of the problem in the case of fixed resources

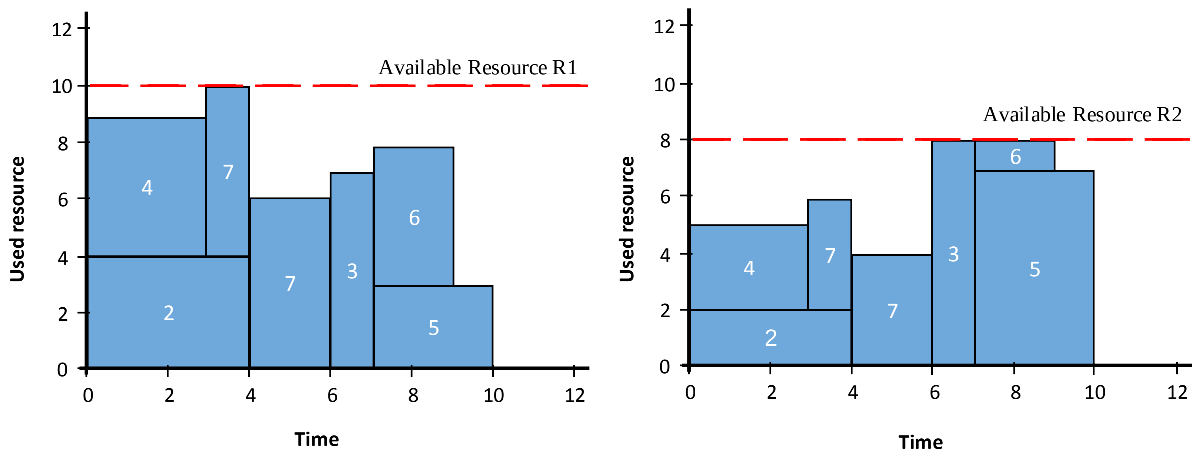

Figure5. Total amount of consumed resources in each period when resource requirements and availability is constant

Table 4 shows a summary of optimal activity start times obtained with and without consideration of time-varying resource requirements and capacities. The difference between these two times is given in the column "Absolute deviation ".

Table 4. Comparing the obtained schedules

Activity	Activity type	Start time		Absolute deviation
		Considering variations in resources	Ignoring variations in resources	
1	Dummy	0	0	0
2	Real	1	0	1
3	Real	0	6	6
4	Real	1	0	1
5	Real	5	7	2
6	Real	9	7	2
7	Real	8	3	5
8	Dummy	11	10	1

As can be seen, ignoring the time-based variations in consumption and availability of resources may lead to underestimation of project makespan and significant deviation from the optimal activity sequence. Therefore, it can also affect the cost of project implementation and increase it. As table 1 shows, the majority of previous models in the context of RCPSP models with robustness considerations ignore the time-based variations in resources.

As explained earlier, in order to solve the proposed model, two metaheuristic algorithms have been proposed.There are two reasons for solving the proposed model with multi-objective metaheuristics NSGA-II and MOPSO. First, since the proposed model is a more general version of RCPSP, which belongs to the class of NP-hard problems, the proposed model is NP-hard as well. Second, solving multi-objective problems with metaheuristics has an advantage over alternative approaches, that is, it allows a set of Pareto optimal solutions to be obtained at once and the solution space to be searched more efficiently. Figure 6 shows the set of Pareto optimal solutions given by NSGA-II for a problem with 12 activities and 4 types of renewable sources.

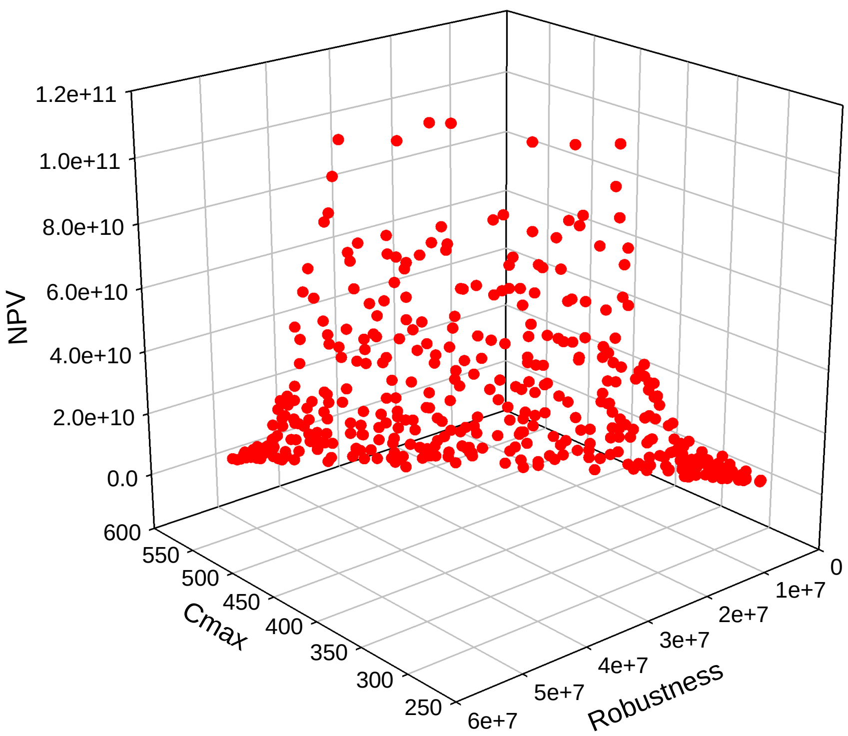

Figure 6. Set of Pareto optimal solutions for the proposed model

As shown in figure 6, the use of completion time minimization, schedule robustness maximization, and cash flow NPV maximization objectives leads to a variety of solutions which, decision maker can choose among at will. There is also a relationship between these objectives, as schedule robustness (the second objective) increases with the increase of project completion time (the first objective). Figure 7 shows the relationship between these objectives in the outputs of NSGA-II for a problem with 12 activities and 4 types of renewable sources. This direct relationship is because a longer project completion time corresponds to longer float times for project activities, and thereby a reinforced schedule robustness or, to put it simply, a lower likelihood of delay in project completion.

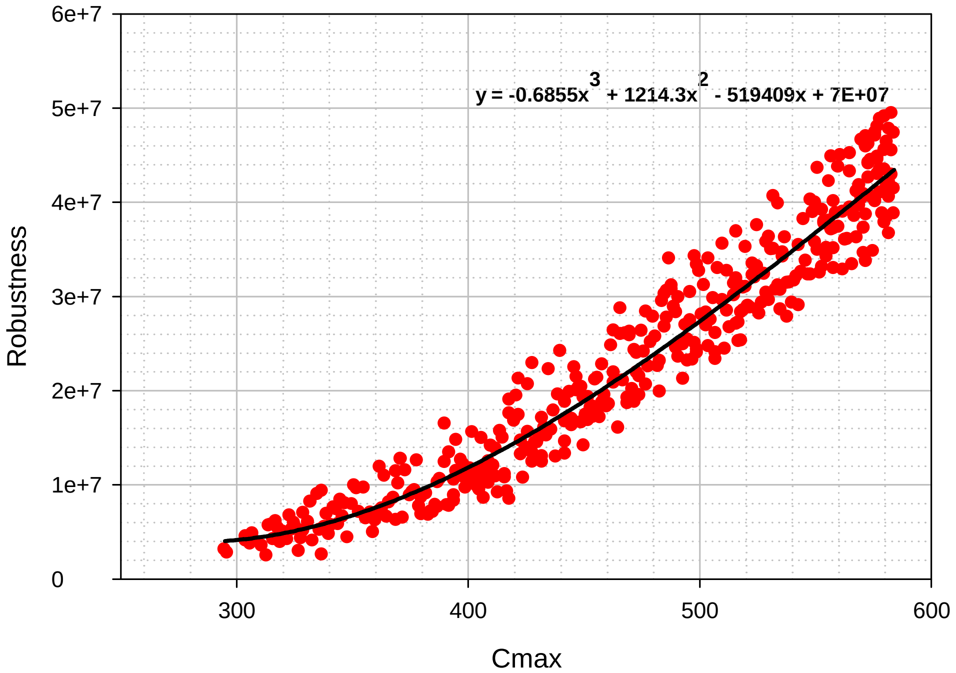

Figure 7. The relationship between project completion time and scheduling robustness

It can also be observed that as the project completion time increases (the first objective), on average, the net present value of cash flows (the third objective) increases as well. Figure 8 highlights this relationship between these two objectives in the outputs of NSGA-II for a problem with 12 activities and 4 types of renewable sources.

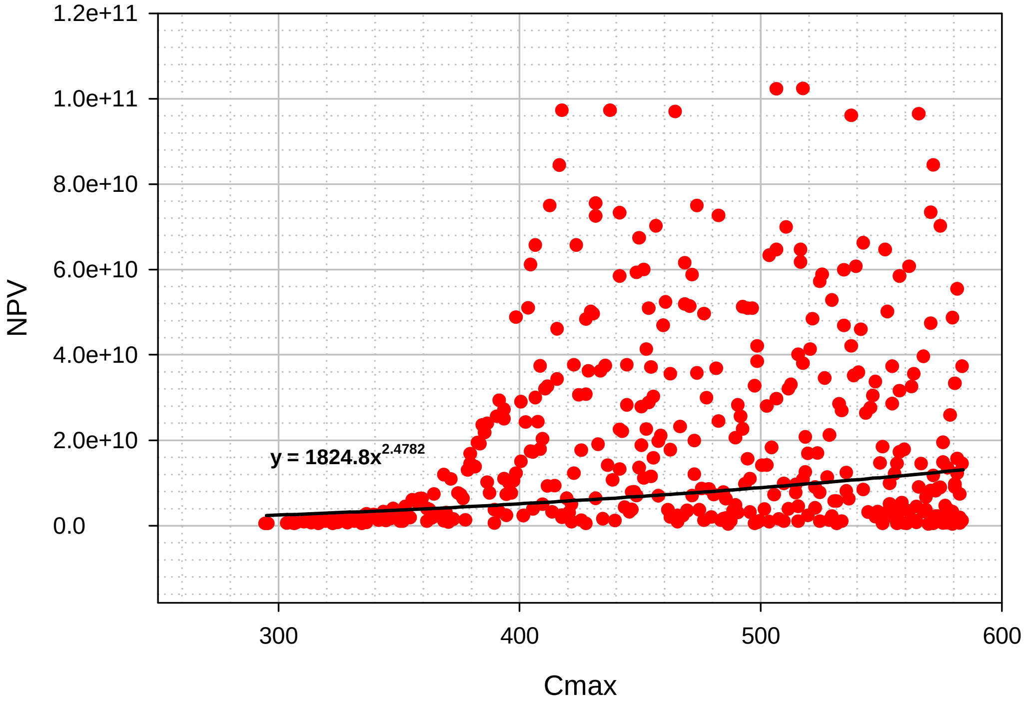

Figure 8. The relationship between project completion time and Net Present Value

To compare the efficiency of the two proposed algorithms, these solution methods are also evaluated by using a group of problem instances with 15, 30, 60, 90 and 120 activities and 2, 3 and 4 renewable resources, which have been derived from the Project Scheduling Problem Library (PSPLIB). The problems of this library lack some of the features required for testing the proposed model, so the amount of resources required for each activity in each period and the available resources in each period (considering time-based variations) as well as the amount of receipts and payments for each activity are generated randomly with uniform distribution. The reason for choosing this particular distribution function and parameters is the presence of data with the same range in other articles as well as PSPLIB. Table 5 shows the parameters used for evaluating the model.

Table 5. Data used for problem parameters

Parameter	Uniform distribution	
	A	B
r_{jkt} *	0	10
R_{kt}	10	40
CF_j^+	18	35
CF_j^-	10	18
Others	Using PSPLIB data	

* Used only for the problems with 15, 60 and 90 activities. For the problems with 30 and 120 activities, we have used the standard data of PSPLIB.

Before solving the problem with NSGA-II and MOPSO, parameters of both algorithms need to be optimized to ensure accurate results and satisfactory performance. One simple and effective method of optimizing the parameters of an algorithm is the use of Taguchi tests. This approach allows our purpose to be achieved easily and via minimum number of trials. In this process, parameters of NSGA-II and MOPSO algorithms are categorized into three levels and then tuned separately for small, medium and large problems. The results of parameter tuning with Taguchi method are shown in tables 6 and 7.

Table 6. Tuned parameters of NSGA-II algorithm

Problem size	Number of activities	NSGA-II Parameters					
		Max Iterations	Population size	Crossover Percentage	Mutation Percentage	Mutation rate	Mutation step size
Small	J_{15}	150	200	0.9	0.2	0.1	0.05
Medium	J_{30}, J_{60}	200	300	0.9	0.2	0.1	0.05
Large	J_{90}, J_{120}	250	400	0.9	0.2	0.1	0.05

Table 7. Tuned parameters of MOPSO algorithm

Problem size	Num of activities	MOPSO Parameters											
		Max Iter	Pop size	Rep size	Inertia Weight	IW damping rate	C_1	C_2	Num of Grids	Inflation Rate for Grids	Leader selection pressure	Deletion selection pressure	Mutation rate
Small	J_{15}	150	150	200	1.5	0.85	1	2	7	0.2	2	6	0.1
Medium	J_{30}, J_{60}	200	175	300	1.5	0.85	1	2	7	0.2	2	6	0.1
Large	J_{90}, J_{120}	250	200	400	1.5	0.85	1	2	7	0.2	2	6	0.1

Performance of the algorithms used for solving the proposed model is evaluated in terms of 5 criteria described in the following.

- **Number of Pareto Solutions (NPS):** One convenient criterion for measuring the performance of an algorithm developed for solving multi-objective problems is the number of non-dominated solutions found by that algorithm. This criterion is particularly more important when the problem has a discrete nature and there is a possibility of producing duplicate solutions for objective functions. Naturally, access to a higher number of Pareto solutions can assist the decision maker to adopt better decisions. Thus, any algorithm that can provide more Pareto solutions will be considered to have a better performance.

- **Quality Metric (QM):** This criterion is one of the most important measures for comparing the quality of Pareto solutions obtained by two different multi-objective algorithms. To calculate this criterion, we compare the Pareto solutions of both algorithms together and remove the solutions dominated by the solution of other algorithm. The QM of each algorithm is defined as the ratio of the number of its remaining non-dominated solutions (after comparison) to the initial number of its solutions (before comparison). Naturally, the algorithm that achieves a higher QM value has a better performance.

- **Mean Ideal Distance (MID):** This criterion measures the proximity of Pareto solutions to the ideal point (f_1^{best}, f_2^{best}, f_3^{best}) and is calculated by equation 30.

$$MID = \frac{\sum_{i=1}^{n} \sqrt{\left(\frac{f_{1i} - f_1^{best}}{f_1^{nadir} - f_1^{best}} \right)^2 + \left(\frac{f_{2i} - f_2^{best}}{f_2^{nadir} - f_2^{best}} \right)^2 + \left(\frac{f_{3i} - f_3^{best}}{f_3^{nadir} - f_3^{best}} \right)^2}}{n} \quad (30)$$

In equation 30, n is the number of non-dominated solutions, and f_i^{best} and f_i^{nadir} are the best and worst values of objective function i subject to existing constraints. In view of this definition, the algorithm with lower MID value has a better performance.

- **Diversification metric (DM):** This criterion represents the dispersion of Pareto solutions and can be calculated by equation 31:

$$DM = \sqrt{\left(\frac{\max\{f_{1i}\} - \min\{f_{1i}\}}{f_1^{nadir} - f_1^{best}} \right)^2 + \left(\frac{\max\{f_{2i}\} - \min\{f_{2i}\}}{f_2^{nadir} - f_2^{best}} \right)^2 + \left(\frac{\max\{f_{3i}\} - \min\{f_{3i}\}}{f_3^{nadir} - f_3^{best}} \right)^2} \quad (31)$$

This criterion in fact measures the diameter of the cube encompassing the space created by the boundaries of objective functions for the set of non-dominated solutions. In view of this definition, a higher DM value signifies the better performance of the algorithm.

- **Spacing metric (SM):** This criterion measures how uniform is the dispersion of the set of non-dominated solutions and is defined by equation 32.

$$SM = \frac{\sum_{i=1}^{n-1} \left| d_i - \bar{d} \right|}{(\mathrm{n}-1)\bar{d}} \quad (32)$$

In equation 32, d_i denotes the Euclidean distance between consecutive solutions in the set of non-dominated solutions obtained by the algorithm, and $\bar{d}$ is the average of these distances.

In this section, the discussed algorithms are evaluated by the use of fifteen problems with 15, 30, 60, 90 and 120 activities and 2, 3 and 4 renewable resources. The values of performance evaluation criteria for the two algorithms are shown in tables 8-12. For better comparison, figures 9-13 illustrate the plots of obtained results in terms of different problem sizes.

Table 8. Results obtained for the test problems with 15 activities

Number of activities	Number of resources	Iteration	Comparison metrics									
			NPS		QM		MID		DM		SM	
			NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO
J_{15}	K_2	1	189	144	0.9350	0.6100	1.4722	1.4842	0.8523	0.7994	0.0044	0.0069
		2	190	156	0.9200	0.5850	1.4997	1.5330	0.7966	0.7556	0.0053	0.0065
		3	185	149	0.9412	0.6435	1.5824	1.5667	0.7414	0.6842	0.0042	0.0067
	Average		188	150	0.9321	0.6128	1.5181	1.5280	0.7968	0.7464	0.0046	0.0067
J_{15}	K_3	1	197	138	0.8900	0.7950	1.0918	1.2123	0.8362	0.7655	0.0038	0.0073
		2	193	164	0.8950	0.8050	1.0822	1.1668	0.8411	0.7875	0.0018	0.0061
		3	198	152	0.8800	0.7300	1.0996	1.1352	0.8264	0.7924	0.0010	0.0066
	Average		196	151	0.8883	0.7767	1.0912	1.1714	0.8346	0.7818	0.0022	0.0067
J_{15}	K_4	1	195	126	0.9250	0.6800	1.4182	1.6298	0.7998	0.7250	0.0051	0.0080
		2	199	141	0.8950	0.6650	1.4194	1.4980	0.7793	0.7552	0.0027	0.0071
		3	195	137	0.9400	0.5600	1.4064	1.5137	0.8071	0.7536	0.0050	0.0054
	Average		196	135	0.9200	0.6350	1.4147	1.5472	0.7954	0.7446	0.0043	0.0068

Table 9. Results obtained for the test problems with 30 activities

Number of activities	Number of resources	Iteration	Comparison metrics									
			NPS		QM		MID		DM		SM	
			NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO
J_{30}	K_2	1	300	232	0.9467	0.6833	0.9956	1.1939	0.8204	0.7148	0.0026	0.00054
		2	300	202	0.9800	0.7200	0.9845	1.0278	0.8283	0.7854	0.0032	0.0238
		3	300	198	0.9400	0.7533	0.9784	1.4215	0.8298	0.6757	0.0025	0.0051
	Average		300	211	0.9556	0.7189	0.9862	1.2144	0.8262	0.7253	0.0028	0.0098
J_{30}	K_3	1	300	209	0.9333	0.7800	0.9315	1.0158	0.7981	0.7420	0.0028	0.0048
		2	298	249	0.9333	0.7167	0.9307	1.0299	0.8040	0.7376	0.0029	0.0033
		3	300	234	0.8867	0.7800	0.9187	1.0031	0.8084	0.7385	0.0031	0.0043
	Average		299	231	0.9178	0.7589	0.9270	1.0163	0.8035	0.7394	0.0029	0.0041
J_{30}	K_4	1	298	219	0.9567	0.3933	0.9003	0.9962	0.8032	0.7018	0.0026	0.0046
		2	300	237	0.9600	0.4433	0.8918	1.1858	0.8072	0.6463	0.0028	0.0042
		3	299	230	0.9467	0.3233	0.8991	0.9749	0.8015	0.7216	0.0022	0.0044
	Average		299	229	0.9545	0.3866	0.8971	1.0523	0.8040	0.6899	0.0025	0.0044

Table 10. Results obtained for the test problems with 60 activities

Number of activities	Number of resources	Iteration	Comparison metrics									
			NPS		QM		MID		DM		SM	
			NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO
J_{60}	K_2	1	300	243	0.9667	0.8233	0.8514	1.0069	0.8074	0.7139	0.0024	0.0028
		2	300	296	0.9833	0.7000	0.8514	1.0363	0.8074	0.7183	0.0024	0.0032
		3	300	296	0.9833	0.7000	0.8514	1.0363	0.8074	0.7183	0.0024	0.0032
	Average		300	278	0.9778	0.7411	0.8514	1.0265	0.8074	0.7168	0.0024	0.0031
J_{60}	K_3	1	300	240	0.9733	0.5600	0.8342	1.0886	0.8100	0.6846	0.0020	0.00056
		2	300	260	0.9600	0.7667	0.7104	1.0572	1.0860	0.6997	0.0012	0.0018
		3	300	244	0.9700	0.5633	0.8239	1.1681	0.8101	0.6875	0.0022	0.0041
	Average		300	248	0.9678	0.6300	0.7895	1.1046	0.9020	0.6906	0.0018	0.0022
J_{60}	K_4	1	300	250	0.9333	0.6267	0.7728	0.8304	0.8566	0.7975	0.0023	0.0040
		2	300	282	0.9267	0.6133	0.7728	1.0433	0.9565	0.6962	0.0023	0.00049
		3	300	219	0.9667	0.6567	0.7873	0.7808	0.9238	0.8478	0.0020	0.0076
	Average		300	250	0.9422	0.6322	0.7776	0.8848	0.9123	0.7805	0.0022	0.0040

Table 11. Results obtained for the test problems with 90 activities

Number of activities	Number of resources	Iteration	Comparison metrics									
			NPS		QM		MID		DM		SM	
			NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO
J_{90}	K_2	1	400	364	0.9925	0.5075	0.7882	1.1714	0.8657	0.7174	0.0010	0.0012
		2	400	388	0.9900	0.5325	0.8255	1.3722	0.8283	0.6797	0.00093	0.0016
		3	400	382	0.9900	0.5525	0.8344	1.2147	0.8160	0.6952	0.00058	0.0022
	Average		400	378	0.9908	0.5308	0.8160	1.2528	0.8367	0.6974	0.0008	0.0017
J_{90}	K_3	1	400	327	0.9950	0.7300	0.6873	1.0535	1.5157	0.7145	0.0014	0.0031
		2	400	326	0.9800	0.7025	0.6661	1.1571	0.7616	0.6892	0.0023	0.0031
		3	400	326	0.9800	0.7025	0.6661	1.1571	1.2233	0.6892	0.0023	0.0031
	Average		400	326	0.9850	0.7117	0.6732	1.1226	1.1669	0.6976	0.0020	0.0031
J_{90}	K_4	1	400	347	0.9925	0.7075	0.7118	0.8857	1.1617	0.7410	0.0025	0.0029
		2	400	312	0.9775	0.8400	0.7149	1.0225	1.1887	0.7180	0.0020	0.0031
		3	400	305	0.9800	0.8175	0.7149	1.3401	1.1887	0.6697	0.0020	0.0033
	Average		400	321	0.9833	0.7883	0.7139	1.0828	1.1797	0.7096	0.0022	0.0031

Table 12. Results obtained for the test problems with 120 activities

Number of activities	Number of resources	Iteration	Comparison metrics									
			NPS		QM		MID		DM		SM	
			NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO	NSGA-II	MOPSO
J_{120}	K_2	1	400	299	0.9925	0.9125	0.7566	1.3622	0.6979	0.4645	0.0025	0.0034
		2	400	367	0.9925	0.9225	0.7566	1.4278	0.6979	0.4619	0.0025	0.0027
		3	400	354	0.9825	0.9475	0.7566	1.3413	0.6979	0.4557	0.0025	0.0025
	Average		400	340	0.9892	0.9275	0.7566	1.3771	0.6979	0.4607	0.0025	0.0029
J_{120}	K_3	1	400	303	0.9750	0.8275	0.6702	1.0645	1.0291	0.4859	0.0020	0.0033
		2	400	318	0.9875	0.8325	0.6702	0.9954	1.0291	0.4912	0.0020	0.0032
		3	400	330	0.9900	0.8650	0.6729	0.8030	1.0069	0.5331	0.0021	0.0030
	Average		400	317	0.9842	0.8417	0.6711	0.9543	1.0217	0.5034	0.0020	0.0032
J_{120}	K_4	1	400	260	0.9875	0.8500	0.6662	0.8200	1.2981	0.5269	0.0025	0.0039
		2	400	235	0.9875	0.9050	0.6462	0.7105	1.3397	0.6267	0.0025	0.0043
		3	400	362	0.9900	0.7300	0.6662	0.8649	1.2981	0.5185	0.0025	0.0028
	Average		400	286	0.9883	0.8283	0.6595	0.7985	1.3120	0.5574	0.0025	0.0037

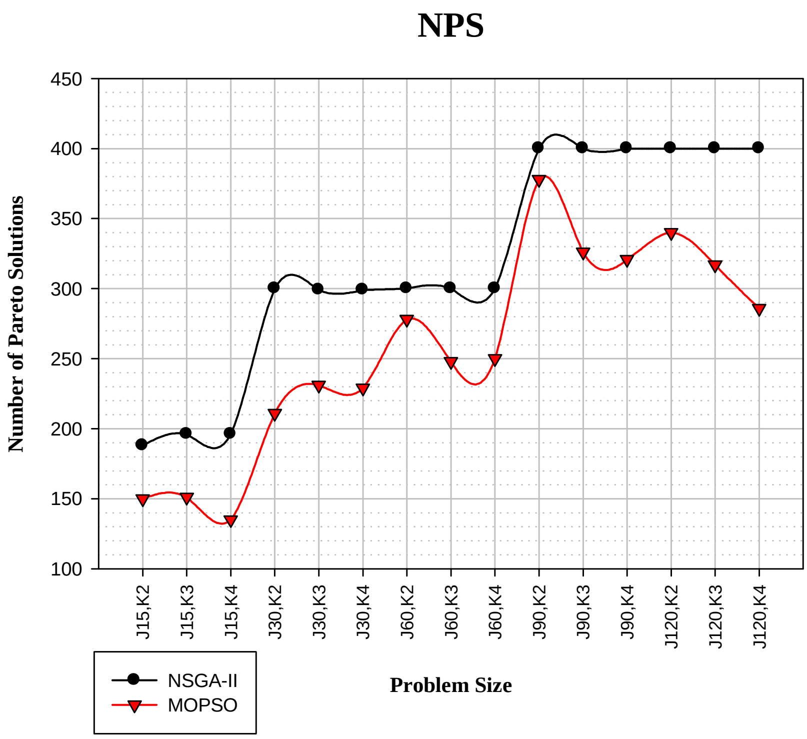

Figure9. The result of comparing the algorithms in terms of NPS

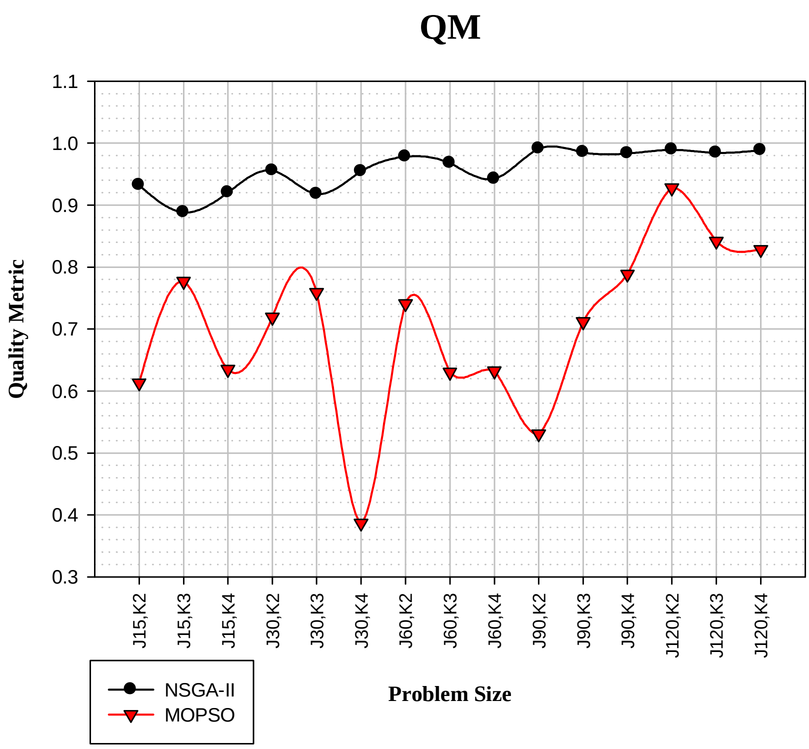

Figure10. The result of comparing the algorithms in terms of QM

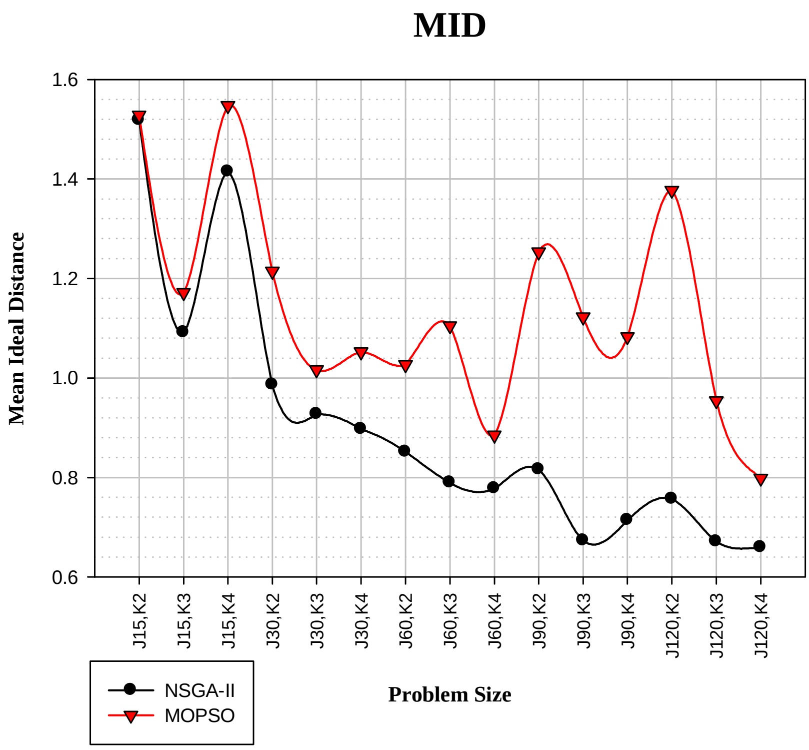

Figure11. The result of comparing the algorithms in terms of MID

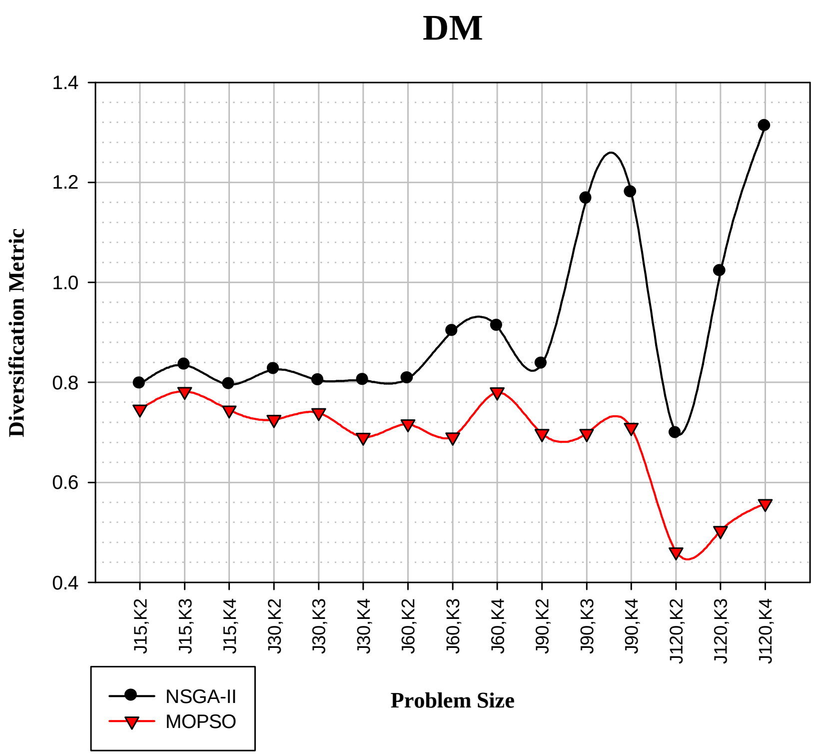

Figure 12. The result of comparing the algorithms in terms of DM

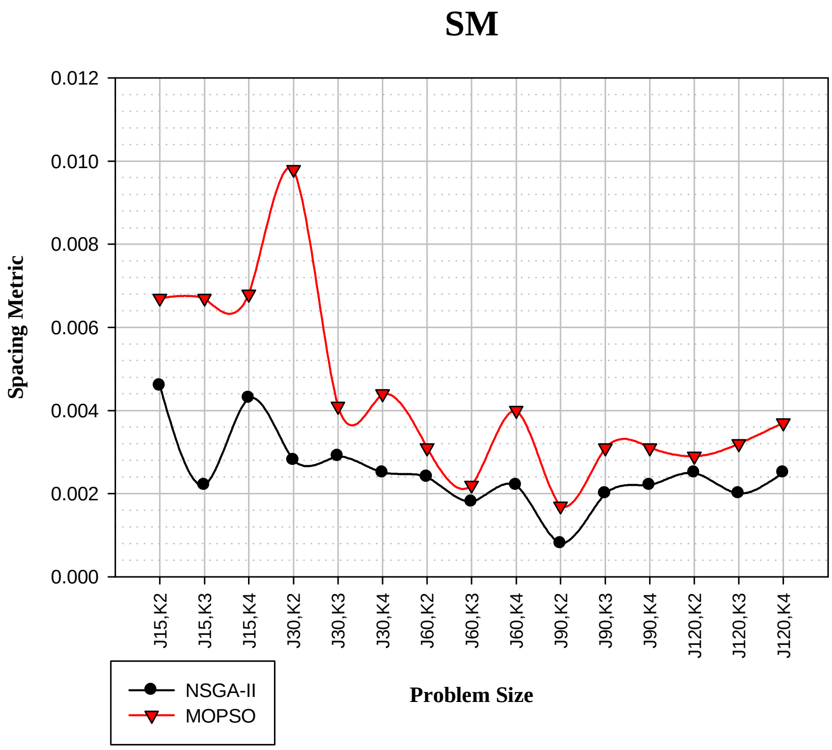

Figure 13. The result of comparing the algorithms in terms of SM

As can be seen, the proposed solution methods is assessed based on fifteen problem instances of different sizes, each of which is solved 3 times with both NSGA-II and MOPSO algorithm. Tables 8-12 show the value of five evaluation criteria for the problems with 15, 30, 60, 90 and 120 activities. The mean values of performance criteria NPS, QM, MID, DM and SM, obtained after solving the problems of different sizes are shown respectively in the graphs of Figures 9-13. According to these graphs, the following results can be concluded.

- For each specific problem, although the population size in NSGA-II is equal to the repository size in MOPSO algorithm, NSGA-II has obtained a higher number of unique Pareto solutions.
- From the QM perspective, regardless of the problem size, NSGA-II has shown better capability than MOPSO in providing Pareto optimal solutions of higher quality.
- NSGA-II also outperforms MOPSO in terms of MID criterion. This advantage of NSGA-II gradually grows with the size of the problem.
- For each specific problem, NSGA-II has a better DM value than MOPSO, which signifies its ability to search for non-dominated solutions more extensively and thus provide the decision-maker(s) with more alternatives.
- The results show that for each specific problem, NSGA-II yields non-dominated solutions with lower SM values; a result that again points to its superiority over MOPSO.

In conclusion, the results show that features of NSGA-II and effectiveness of its mechanism in finding Pareto optimal solutions of the proposed model allow it to exhibit better performance in this application. This superiority of NSGA-II over MOPSO algorithm is evident in all proposed evaluation criteria and for the problems of all sizes.

5- Conclusion

In this paper, we introduced a multi-objective mathematical model for robust resource-constrained project scheduling with discounted cash flows, time-varying resource requirements, and time-varying resource capacities. In the proposed model, the goal is to minimize the project makespan, maximize the schedule robustness, and maximize NPV simultaneously, in order to assist the project managers to make better and more realistic decisions for timely completion of project activities. It was found that ignoring the predicted time-based variations in consumption and availability of resource (assuming them to be constant over time) may lead to inaccurate scheduling; thus to avoid this issue, these variations were incorporated into the proposed model. Since RCPSP belongs to the class of NP-hard problems and solving multi-objective mathematical optimization models with metaheuristic algorithms leads to more effective determination of Pareto optimal solutions, two multi-objective metaheuristic algorithms, NSGA-II and MOPSO, were adjusted and used to solve this model. The proposed solution methods were evaluated by fifteen problems of different sizes, which were derived from the problems of PSPLIB. After tuning the parameters of both algorithms with Taguchi method, solution methods were compared in terms of five different evaluation criteria NPS, QM, MID, DMand SM. The comparisons showed that, on average, NSGA-II yields better results than MOPSO algorithm.

References:

Abbasi, B., Shadrokh, S. & Arkat, J. 2006. Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied mathematics and computation,* 180**,** 146-152.

Afshar-Nadjafi, B. 2016. A new proactive approach to construct a robust baseline schedule considering quality factor. *International Journal of Industrial and Systems Engineering,* 22**,** 63-72.

Akkan, C., Drexl, A. & Kimms, A. 2005. Network decomposition-based benchmark results for the discrete time–cost tradeoff problem. *European Journal of Operational Research,* 165**,** 339-358.

Al-Fawzan, M. A. & Haouari, M. 2005. A bi-objective model for robust resource-constrained project scheduling. *International Journal of production economics,* 96**,** 175-187.

Artigues, C., Leus, R. & Nobibon, F. T. 2013. Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal,* 25**,** 175-205.

Bartusch, M., Mohring, R. H. & Radermacher, F. J. 1988. Scheduling project networks with resource constraints and time windows. *Annals of operations Research,* 16**,** 199-240.

Berthaut, F., Pellerin, R., Perrier, N. & Hajji, A. 2014. Time-cost trade-offs in resource-constraint project scheduling problems with overlapping modes. *International Journal of Project Organisation and Management,* 6**,** 215-236.

Chtourou, H. & Haouari, M. 2008. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & industrial engineering,* 55**,** 183-194.

Coello, C. A. C., Pulido, G. T. & Lechuga, M. S. 2004. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation,* 8**,** 256-279.

Creemers, S. 2015. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *Journal of Scheduling,* 18**,** 263-273.

De Reyck, B., Demeulemeester, E. & Herroelen, W. 1999. Algorithms for scheduling projects with generalized precedence relations. *Project Scheduling.* Springer.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation,* 6**,** 182-197.

Delgoshaei, A., Ariffin, M., Baharudin, B. & Leman, Z. 2015. Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithms. *International Journal of Industrial Engineering Computations,* 6**,** 503-520.

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W. & Vanhoucke 1998. New computational results on the discrete time/cost trade-off problem in project networks. *Journal of the Operational Research Society,* 49**,** 1153-1163.

Fallah, M., Aryanezhad, M. & Ashtiani, B. Preemptive resource constrained project scheduling problem with uncertain resource availabilities: Investigate worth of proactive strategies. Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on, 2010. IEEE, 646-650.

Ghassemi-Tari, F. & Olfat, L. 2007. Development of a set of algorithms for the multi-project scheduling problems. *Journal of Industrial and Systems Engineering,* 1**,** 11-17.
Goldratt, E. M. 1997. *Critical chain: A business novel,* North River Press Great Barrington, MA.

Gomes, H. C., Das Neves, F. D. A. & Souza, M. J. F. 2014. Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations. *Computers & Operations Research,* 44**,** 92-104.

Hao, X., Lin, L. & Gen, M. 2014. An effective multi-objective EDA for robust resource constrained project scheduling with uncertain durations. *Procedia Computer Science,* 36**,** 571-578.

Haouari, M. & Al-Fawzan, M. A. 2002. A bi-objective model for maximizing the quality in project scheduling. DIMACS technical report 2002.

Hartmann, S. 2015. Time-Varying Resource Requirements and Capacities. *Handbook on Project Management and Scheduling Vol. 1.* Springer.

Hartmann, S. & Briskorn, D. 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research,* 207**,** 1-14.

Hsu, C.-C. & Kim, D. S. 2005. A new heuristic for the multi-mode resource investment problem. *Journal of the Operational Research Society,* 56**,** 406-413.

Kang, C. & Choi, B.-C. 2015. An adaptive crashing policy for stochastic time-cost tradeoff problems. *Computers & Operations Research,* 63**,** 1-6.

Kumar, S. & Arunagiri, A. 2010. Learning curve consideration in makespan computation using artificial neural network approach. *Journal of Industrial and Systems Engineering,* 4**,** 183-192.

Lamas, P. & Demeulemeester, E. 2016. A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *Journal of Scheduling,* 19**,** 409-428.

Lambrechts, O., Demeulemeester, E. & Herroelen, W. 2008. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of scheduling,* 11**,** 121-136.

Lambrechts, O., Demeulemeester, E. & Herroelen, W. 2011. Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research,* 186**,** 443-464.

Leyman, P. & Vanhoucke, M. 2016. Payment models and net present value optimization for resource-constrained project scheduling. *Computers & Industrial Engineering,* 91**,** 139-153.

Liu, Z. & Zheng, Y. Resource-constrained multiple projects scheduling with the objective of minimizing activities cost. Control and Decision Conference, 2008. CCDC 2008. Chinese, 2008. IEEE, 1027-1032.

Mogaadi, H. & Chaar, B. F. Scenario-Based Evolutionary Approach for Robust RCPSP. Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015, 2016. Springer, 45-55.

Palacio, J. D. & Larrea, O. L. 2017. A lexicographic approach to the robust resource-constrained project scheduling problem. *International Transactions in Operational Research,* 24**,** 143-157.

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**,** 1-16.

Russell, A. 1970. Cash flows in networks. *Management Science,* 16**,** 357-373.

Shi, Y.-J., Qu, F.-Z., Chen, W. & Li, B. 2010. An artificial bee colony with random key for resource-constrained project scheduling. *Life system modeling and intelligent computing.* Springer.

Sobel, M. J., Szmerekovsky, J. G. & Tilson, V. 2009. Scheduling projects with stochastic activity duration to maximize expected net present value. *European Journal of Operational Research,* 198**,** 697-705.

Soltani, A. & Haji, R. 2007. A project scheduling method based on fuzzy theory. *Journal of Industrial and Systems Engineering,* 1**,** 70-80.

Song, Y., Liu, J., Wimmers, M. O. & Jiang, Z. A differential evolution algorithm with local search for resource investment project scheduling problems. Evolutionary Computation (CEC), 2015 IEEE Congress on, 2015. IEEE, 1725-1731.

Sprecher, A. 2012. *Resource-constrained project scheduling: Exact methods for the multi-mode case,* Springer Science & Business Media.

Ulusoy, G., Sivarikaya-Şerifoglu, F. & Şahin, Ş. 2001. Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. *Annals of Operations research,* 102**,** 237-261.

Vanhoucke, M., Demeulemeester, E. & Herroelen, W. 2002. Discrete time/cost trade-offs in project scheduling with time-switch constraints. *Journal of the Operational Research Society***,** 741-751.

Wang, W.-X., Wang, X., Ge, X.-L. & Deng, L. 2014. Multi-objective optimization model for multi-project scheduling on critical chain. *Advances in Engineering Software,* 68**,** 33-39.

Wiesemann, W., Kuhn, D. & Rustem, B. 2010. Maximizing the net present value of a project under uncertainty. *European Journal of Operational Research,* 202**,** 356-367.

Wu, S., Wan, H.-D., Shukla, S. K. & Li, B. 2011. Chaos-based improved immune algorithm (CBIIA) for resource-constrained project scheduling problems. *Expert Systems with Applications,* 38**,** 3387-3395.

Xiong, J., Chen, Y., Liu, J. & Abbass, H. A. An evolutionary multi-objective scenario-based approach for stochastic resource investment project scheduling. Evolutionary Computation (CEC), 2011 IEEE Congress on, 2011. IEEE, 2767-2774.

Xiong, J., Liu, J., Chen, Y. & Abbass, H. A. 2014. A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems. *IEEE Transactions on Evolutionary Computation*, 18**,** 742-763.

Yuan, X., Liu, J. & Wimmers, M. O. A multi-agent genetic algorithm with variable neighborhood search for resource investment project scheduling problems. Evolutionary Computation (CEC), 2015 IEEE Congress on, 2015. IEEE, 23-30.