Optimizing a Bi-objective Preemptive Multi-mode Resource-Constrained Project Scheduling Problem: NSGA-II and MOICA Algorithms

Javad Hassanpour[a,*], Mohammad Ghodoosi[b], Zahra Sadat Hosseini[c]

[a] *Msc, Department of industrial Engineering, Quchan University of Advanced Technology, Quchan, Iran*
[b] *Msc ,Department of industrial Engineering, university of Torbat-e Heydarieh, Torbat-e Heydarieh, Iran*
[c] *PhD student of Industrial Engineering, Yazd university ,Yazd, Iran*

Received 24 April 2015; Revised 16 November 2015; Accepted 28 November 2015

Abstract

The aim of a multi-mode resource-constrained project scheduling problem (MRCPSP) is to assign resource(s) with the restricted capacity to an execution mode of activities by considering relationship constraints to achieve pre-determined objective(s). These goals vary with managers or decision makers of any organization who should determine suitable objective(s) considering organization strategies. Also, we introduce the preemptive extension of the problem which allows for activity splitting. In this paper, the preemptive multi-mode resource-constrained project scheduling problem (P-MMRCPSP) with Minimum makespan and the maximization of net present value (NPV) has been considered. Since the considered model is NP-Hard, the performance of our proposed model is evaluated by comparison with two well-known algorithms: non-dominated sorting genetic algorithm (NSGA II) and multi-objective imperialist competitive algorithm (MOICA). These metaheuristics have been compared on the basis of a computational experiment performed on a set of instances obtained from standard test problems constructed by the ProGen project generator, where, additionally, cash flows were generated randomly with the uniform distribution. Since the effectiveness of most meta-heuristic algorithms significantly depends on choosing the proper parameters. A Taguchi experimental design method (DOE) was applied to set and estimate the proper values of GAs parameters for improving their performances. The computational results show that the proposed MOICA outperforms the NSGA-II.

Keywords: Multi-objective project scheduling, Resource constraint, Preemptive, Net present value, Meta-heuristic algorithm.

1. Introduction

Project scheduling is an important task of project management. In recent decades, resource-constrained project scheduling has become a standard problem in literature review. The standard resource-constraint project scheduling problem (RCPSP) assumes that an activity can only be executed in a single way, which is determined by a fixed duration and fixed resource requirements. The RCPSP along with some of its extensions has been widely studied in the literature. The multi-mode RCPSP (MMRCPSP) is a generalized version of the RCPSP, in which each activity can be performed in one of a set of modes with a specific activity duration and resource requirements.

The assumptions for the multi-mode RCPSP can be summarized as follows. We consider two additional activities $j = 0$ and $j = n + 1$, representing the start and the completion of the project, respectively. Both are "dummy" activities with durations of 0 and no resource requests. Activity j must be performed in one of its modes which are labeled $1 \ldots M_j$ with M_j being the number of modes. Once started in one of its modes, an activity must be completed in that mode; mode changes and preemption are not permitted. The duration of activity j being executed in mode m is given by d_{jm}. The request of activity j being executed in mode m for resource k is r_{kmj}. In addition to renewable resources, also non-renewable resources are often considered in multi-mode models. A schedule for multi-mode RSPSP assigns a start time S_j and a mode m_j to each activity j.

The first method to solve the multi-mode problems was presented by Slowinski (1980), proposing one-stage and two-stage linear programming. Speranza and Vercellis (1993) presented branch and bound (B&B) algorithm; however, Hartmann and Sprecher (1996) revealed that this algorithm is not able to solve the problem with one and two renewable constraints. After a while, Sprecher et al (1997), Hartmann and Drexl (1998), and Sprecher and Drexl (1998) proposed a branch and bound algorithm. However, several algorithms capable of solving the MMRCPSP have been proposed in recent years by the following researchers: knots et al (2000), Nonobe et al (2001), Alcaraz et al (2003), Heilmann (2003), Zhu et al

[*] Corresponding author Email address: j.hasanpour@qiet.ac.ir

(2006), Ranjbar et al (2008), Ranjbar (2011), Afshar-Nadjafi et al (2013).

Preemptive multi-mode resource-constrained project scheduling problem (P-MMRCPSP) is referred to a generalization of multi-mode resource-constrained project scheduling problem which allows activities to be preempted at any time instance and restarted later on at no additional cost. For the single-mode case, Kaplan (1988) and Demeulemeester and Herroelen (1996) presented an exact algorithm; Ballestin et al. (2008) and Vanhoucke and Debels (2008) used heuristic algorithms. However, Damay et al. (2007) proposed a linear programming method. Then, in multi-mode models, Buddhakulsomsiri and Kim (2006) proved that preemption is very effective to improve the optimal project makespan.

Although some studies concerning preemption have been conducted on the P-RCPSP and P-MMRCPSP, none of them could have been used to solve large-scale problems and provide the optimal solution within a reasonable period of time. Therefore, a number of heuristic and meta-heuristic algorithms have been introduced to solve such problems. Boctor (1996) tested the heuristic programming and proposed five instances with the highest probability of achieving the best solution. Drexl and Grünewald (1993) emphasized random sampling, and then Boctor (1996) proposed a computational heuristic algorithm based on critical method. Kolisch and Drexl (1997) studied a local search method. Lova et al. (2006) presented multi-stage heuristic algorithm based on priority for solving MMRCPSP.

To solve the MMRCPSP, the following methods were used: Mori and Tseng (1997), Hartmann (2001), Alcaraz et al. (2003), and Lova et al. (2009) used genetic algorithm; Bouleimen and Lecocq (2003) used the simulated algorithm; Nonobe and Ibaraki (2001) used tabu search algorithm; Zhang et al. (2006) used particle swarm optimization algorithm, and Ranjbar et al. (2008) used scatter search. However, several algorithms by heuristic and meta-heuristic algorithms have been proposed in recent years: Debels et al. (2006), Van peteghem and Vanoucke (2010), Coelho and Vanhoucke (2011), Afshar-Nadjafi et al. (2013), and Zahra sadat Hosseini et al. (2014).

Ballestin and Blanco (2011) claimed that a few research studies have investigated the multi-objective resourced-constrained project scheduling problem. They tackled such a shortcoming and applied Non-dominated storing genetic algorithm (NSGAII), Strength pareto evolutionary algorithm (SPEAII), and Pareto simulated annealing (PSA). Deblaere et al. (2010) presented multi-mode resource-constrained problem with the objective of minimizing project makespan. Many of the recent research studies in project scheduling focus on maximizing the NPV of the project using the sum of positive and negative discounted cash flows throughout the life cycle of the project. The concept of maximizing cash flow net present value in project scheduling with constrained resource was firstly introduced by Russell (1970). Doersch and Patterson (1997) added binary programming into RCPSP problem maximizing net present value. Goto et al. (2001) proposed a meta-heuristic algorithm containing two-step taboo search to maximize net present value. Ulusoy et al (2001) introduced four payment methods in project scheduling problems, and then considered resource-constrained balance time-resource scheduling problem in AOA networks with the objective of maximizing net present value with respect to payment methods using genetic algorithm. Using payment methods proposed by Ulusoy et al (2001), Mika et al. (2005) presented simulated annealing and taboo search algorithms in activity-on-the-node networks with the objective of maximizing net present value. The preemptive multi-mode RCPSP is categorized in NP-hard problems. However, For the purpose of solving such a hard problem in a reasonable time and finding a good quality solution, it is necessary to choose a powerful multi-objective meta-heuristic algorithm. Nowadays, evolutionary algorithms are used in many fields.

In the literature, little attention has been paid to problems where preemption is allowed. In this paper, we investigate a new bi-objective preemptive multi-mode resource-constrained project scheduling problem based on minimizing makespan and maximizing net present value. The study has been labeled as new one due to bringing in simultaneously the factors of preemption, completing the activity within their earliest and latest finishing periods, NPV maximization and minimizing makespan into the multi-objective multi-mode project scheduling problem. The aim of this paper is to investigate the performance of the proposed MOICA algorithm to solve a bi-objective preemptive multi-mode resource-constrained project scheduling problem.

The reminder of this paper is organized as follows. In section 2, a multi-objective formulation for PMMRCPSP is presented. In section 3, a solution methodology for applied algorithm is presented. In section 4, the structure of the proposed algorithm is explained. The experimental results of comparison between the proposed MOICA algorithm and NSGA-II are shown in section 5. Finally, conclusions are drawn in section 6.

2. Mathematical Formulation

2.1. Constraints

Due to limited resource and precedence relations, there are two types of constraints. Therefore, at a time, an activity can start when all precedence and renewable and nonrenewable resource constraints are satisfied.

2.2. Assumptions

In this paper, a multi-objective project scheduling problem with multi-mode and pre-emptible activities under resource-constrained condition and the following assumptions is considered.

- An activity can start when all precedence as well as this activity do not need set-up time.
- The project is represented as an activity-on-the-oriented node network with the set of activities and the set of pairs of activities: between relationship of start to finish precedence.
- The project contains the dummy activity 1 and dummy activity *N*.
- Each activity is performed in one mode.
- Activities are performed within their earliest start time and latest finish time.
- Resource capacity is pre-specified and limited.
- Each activity will use a given resource unit of the total available resources.
- Each activity will use one or some resources.
- Resources used in project are renewable.
- The duration of each activity is pre-specified and constant throughout the project horizon.
- Cost payment time is determined in the beginning of the project.
- There is no due date for project.
- Financial payment time is determined at the end of each activity.
- There is no limited budget.

2.3. Variables and parameters

Index

N : Number of activities (*i, j* = 1, 2,…, N)
T : Number of periods (*t* = 1, 2,…, T)
K : Renewable resource (k = 1, 2,…, K)

Parameters

M_j: Number of modes of activity j
d_{jm} : Duration of activity j executed in mode *m*
a_{Kt} : Total available units of renewable resources k at time t
r_{kmj} : Number of renewable resources k used by activity j done in mode m
R^{ρ} : Set of renewable resources
T : Maximum project horizon time
M: Big positive number
R^n *:* Set of non-renewable resource
TH *:* Maximum project horizon
α_i : Decrease rate per unit time
CF_{jm}^- : Negative cash flow assigned to mode *m* of activity j
CF_j^+ : Positive cash flow assigned to activity j

Variables

ES_j *:* Earliest start time of activity j
EF_j *:* Earliest finish time of activity j
LF_j *:* Latest finish time of activity j
LS_j *:* Latest start time of activity j

$$X_{jmt} = \begin{cases} 1 & \text{If activity } j \text{ executed in mode } m \\ & is\ performed\ at\ time\ t \\ 0 & \text{Otherwise} \end{cases}$$

$$y_{jm} = \begin{cases} 1 & \text{If activity } j \text{ executed in mode } m \\ 0 & \text{Otherwise} \end{cases}$$

The MINLP formulation for the PMMRCPSP problem is given as follows:

$$max Z_1 = \sum_{j=1}^{n+1} \frac{CF_j^+}{(1+\alpha_i)^{FF_j}} - \sum_{j=1}^{n+1} \sum_{m=1}^{M_j} \sum_{t=ES_j}^{LF_j} \frac{CF_{jm}^-}{(1+\alpha_i)^t} X_{jmt} \tag{1}$$

$$min\, Z_2 = max_t \left\{ \sum_{j=1}^{n+1} \sum_{m=1}^{M_j} t.X_{jmt} \right\}$$

S.t:

$$ES_1 = 0 \tag{2}$$

$$EF_j = ES_j + \sum_{m=1}^{M_j} d_{jm}\, y_{jm} \quad , \ \forall j \tag{3}$$

$$ES_j = max\{EF_i\} \quad , \ \forall j \ \ , i \neq j \tag{4}$$

$$LS_j = LF_j - \sum_{m=1}^{M_j} d_{jm}\, y_{jm} \quad , \ \forall j \tag{5}$$

$$LF_j = min\{LS_j\} \quad , \ \forall j \ \ , i \neq j \ \ , \ \ j \neq 1, n \tag{6}$$

$$LS_n = T \tag{7}$$

$$ES_j\, y_{jm} < t.X_{jmt} + (1 - X_{jmt})M \quad \forall j, j \neq 1, n \ , \quad \forall m, t \tag{8}$$

$$t.X_{jmt} \leq LF_j \tag{9}$$

$$\sum_{t=ES_j}^{LF_j} X_{jmt} = d_{jm} y_{jm} \ , \quad \forall\, j, m \tag{10}$$

$$\sum_{m=1}^{M_j} y_{jm} = 1 \ , \quad \forall j \tag{11}$$

$$\sum_{j=1}^{n}\sum_{m=1}^{M_j} r_{kmj}X_{jmt} \leq a_{Kt} \quad , \ \forall\, k \in R^{\rho}\ ,t = 1,2,..,T \tag{12}$$

$$FF_j = max_t \left\{ t.\sum_{m=1}^{M_j} X_{jmt} \right\} \ , \quad \forall\, j \tag{13}$$

$$SS_j = min_t \left\{ t.X_{jmt} + \left(1 - X_{jmt}\right)M \right\}\ , \qquad \forall\, j, m \tag{14}$$

$$SS_j > FF_i\ a_{ji} \quad , \ \forall\, j \quad \&\ \ i \neq j \tag{15}$$

The first objective function (1) computes the maximizing project net present value and the second objective function computes the minimizing project makespan. Constraint (2) ensures that the earliest start time of muddy activity 1 is equal to 0. Constraints (3) and (4) represent the earliest start and finish time of each activity with respect to preemptive and precedence relations, respectively. Constraints (5) and (6) express the latest start and the latest finish time of each activity with respect to preemptive and precedence relations. Constraint (7) demonstrates that the latest start time of muddy activity *N* is equal to project horizon. Constraints (8) and (9) ensure that each activity is executed within its earliest start time and latest finish time. Constraint set (10) ensures that if mode *m is* selected for activity *j*, the total processing time must equal the corresponding duration. Constraint set (11) ensures exactly one mode is selected for each activity. Constraint (12) specifies resource availability for renewable resources. Constraints (13), (14), and (15) represent the start and finish times of each activity with respect to preemption.

3. Solution Methodology

3.1. Chromosome Evaluation

One of the most significant ways to reach a suitable algorithm is to design the appropriate chromosomes. Then, extraction of problem solution from this chromosome is highly important. In this study, to show the problem answers, two one-row chromosomes are presented that could be encrypted and decrypted easily. Each chromosome is a random vector. Chromosomes represent activities and priorities whose priorities are identified in using resources. Each answer in the considered algorithm is illustrated by two arrays. The first part consists of a linear array with the dimension of 1× (the number of activity), in which each element (gene) is the representative of respective activity, and the value of the element (gene) shows the mode which is used for the related activity. The second part of coding shows the sequence of doing the parts of each activity that includes a linear array with the following dimensions.

1 × (Total duration of activities based on the first part modes)

In presenting the second part of coding, the gene is assigned to each activity based on the time for performing that activity in the specified mode in the first part. In this case, $P_1m_{i_1}$ is the first gene (the time for performing the first activity in a mode i_1) related to the first activity, $P_2m_{i_2}$is the second gene related to the second activity, and $P_nm_{i_n}$is the last gene related to the nth activity. The values in these genes determine the sequence of performing the related part of activity. For instance, for a project with 2 activities, an illustration of chromosomes is presented in figure 1.

Activity	A_1		A_2		
M_i	1	2	1	2	3
R_1	1	1	1	1	-
R_2	2	1	1	1	1
R_3	-	-	1	-	-
P_{im}	3	5	2	4	7

Part 1:

2	4

Part 2:

0.46	0.97	0.34	0.52	0.41	0.22	0.8	0.74

Fig. 1. Illustration of chromosome

As you can see in the above figure, the first part is composed of an array (1×2) in which the value of each gene provides the considered number of modes. In this array, the first gene is related to the first activity that is performed by 2nd mode, and the second gene is related to the second activity that is performed by 2nd mode.

In the second part, since the time for performing the first activity with mode 2 is equal to 5 time units, the first 5 genes are assigned to the first activity, and duration for performing the second activity with mode 2 is equal to 4 time units. Therefore, 4 second genes are assigned to the second activity. The numbers of each gene show the priority of each section of activity. Activity scheduling is carried out based on these priorities, prerequisite, and resource capacity.

3.2. Crossover operator

Intersection operator functions over two parent chromosomes that are the output of selection operator and functions with the possibility of P_c and creates a new child by combining two parents. Currently, intersection operators are employed extensively for project scheduling problems in genetic algorithms.

In this study, the random intersection operator is used, and it functions in such a way that for the first part of chromosome, a random vector [0,1] is created with the dimensions of the first part of a chromosome. If the number of random vector is larger than 0.5, the location of two chromosomes is changed. Otherwise, the location of two chromosome genes remains constant. Then, according to this random vector, the first part of child array is created as illustrated in the table below. Table 1 is created based on the zero and one random numbers, and it presents how intersection action is performed.

Table1
Crossover operator

Parent 1	1	4	6	8
Parent 2	2	3	5	9
Random number	0.64	0.27	0.41	0.79
	>	<	<	>
Offspring 1	2	4	6	9
Offspring 2	1	3	5	8

In addition, for the second part, we do the same processes with regard to the first part's results. In this regard, due to the fact that the employed mode for performing each activity is from every parent, we directly transfer the related genes from the second part to the child. For example, for a project with 2 activities, an illustration of chromosomes is presented in Table 2.

Table2
Illustration of mutation

	Part 1		Part 2								
Parent 1	**2**	**4**	**0.46**	**0.11**	**0.97**	**0.34**	**0.52**	**0.41**	**0.24**	**0.8**	**0.74**
Parent 2	**1**	**3**	**0.19**	**0.71**	**0.56**	**0.83**	**0.29**				
Offspring 1	**2**	**3**	**0.46**	**0.11**	**0.97**	**0.34**	**0.52**	**0.83**	**0.29**		
Offspring 2	**1**	**4**	**0.19**	**0.71**	**0.56**	**0.41**	**0.24**	**0.8**	**0.74**	**0.41**	

3.3. Mutation operator

The number of chromosomes to which mutation operator is applied could be obtained based on the following equation:

The number of mutation application = [(mutation rate) × (Total population)]

To apply mutation randomly, two of the second-part chromosome genes are selected and their values are exchanged. In other words, the priority of performing two activities that are selected randomly is exchanged. Table 3 illustrates how mutation operator works.

Table3
Illustration of mutation

0.5	**0.2**	**0.6**	**0.9**	**0.8**
0.5	**0.8**	**0.6**	**0.9**	**0.2**

4. The Proposed Algorithms

4.1. Non-dominated sorting genetic algorithm II (NSGA-II)

Non-dominated sorting genetic algorithm II (NSGA-II) is one of the most well-known and efficient multi-objective evolutionary algorithms introduced by Deb et al (2000,2002). Ranking and selecting the population fronts are performed by non-dominance technique and a crowding distance. Also, the algorithm to generate new solutions, as for generated offspring, uses crossover operator and mutation operator. Then, the current population and generated offspring are combined together. Finally, the best solutions in terms of non-dominance and crowding distance are selected from combined population as the new population. The non-dominated technique, the calculation of crowding distance, and crowding selection operator are explained below.

4.1.1. Non-dominated technique

Suppose that there are *r* objective functions. When the following conditions are satisfied, the solution x_1 dominates another solution x_2. If x_1 and x_2 do not dominate each other, they are placed in the same front.
(1) For all the objective functions, solution x_1 is not worse than another solution x_2.
(2) For at least one of r objective functions, x_1 is exactly better than x_2.
Front number 1 is made by all solutions that are not dominated by any other solutions.
Also, front number 2 is made by all solutions that are only dominated by solutions in front number 1.

4.1.2. Crowding distance

The crowding distance is a measure of the density of solutions. The value of the crowding distance presents an estimate of the solutions density surrounding a particular solution. The crowding distance measure used in NSGA-II is shown in equation (16). Having a lower value of the crowding distance, the solutions are preferred over solutions with a higher value of the crowding distance.

$$crowding\ Distance_i = \sum_{k=1}^{r} \frac{f_{k,i+1}^{p} - f_{k,i-1}^{p}}{f_{k,total}^{p,max} - f_{k,total}^{p,min}} \qquad (16)$$

Where *r* is the number of objective functions, $f_{k,i+1}^{p}$ is the k-th objective function of the (i+1)-th solution, and $f_{k,i-1}^{p}$ is the *k*-th objective function of the (i-1)-th solution after sorting the population according to crowding distance of the k-th objective function. Also, $f_{k,total}^{p,max}$ and $f_{k,total}^{p,min}$ are the maximum and minimum values of objective function k, respectively.

4.1.3. Tournament selection operator

A binary tournament selection procedure has been applied to select solutions for both the crossover and mutation operators. In the following, we describe this selection operator. First, select two solutions of the population size, and then the lowest front number is selected if the two populations are from different fronts. If they are from the same front, the solution with the highest crowding distance is selected.

4.2. Multi-objective imperialist competitive algorithm (MOICA)

4.2.1. Generating initial empires

Each solution in the imperialist competitive algorithm (ICA) is in a form of an array. Each array consists of variable values to be optimized. In GA terminology, this array is called chromosome; however, in this paper, we use the term "country" for this array. In an N-dimensional optimization problem, a country is a $1 \times N$ array. This array is defined by $P = [p_1, p_2, p_3, \ldots, p_N]$, where p_i is the variable to be optimized. Each variable in a country denotes a socio-political characteristic of a country. From this point of view, the algorithm searches for the best country, that is, the country with the best combination of socio-political characteristics, such as cultural, linguistic, and economical policy (Atashpaz-Gargari and Lucas 2007). After generating countries, a non-dominance technique and a crowding distance are used to rank and select the population fronts, and the members of the front one are saved in archive. Then, the best solutions in terms of the non-dominance and crowding distance are selected from population as the imperialists and the remaining countries are colonies.
To calculate the cost value of each imperialist, the value of each objective function is obtained for each imperialist. Then, the cost value of each objective function is calculated by:

$$C_{i,n} = \frac{\left|f_{i,n}^{p} - f_{i}^{p,best}\right|}{f_{i,total}^{p,max} - f_{i,total}^{p,min}} \qquad (17)$$

Where $C_{i,n}$ is the normalized value of objective function *i* for imperialist *n*, and $f_{i,n}^{p}$ is the value of the objective function *i* for imperialist *n*. Also,$f_{i}^{p,best}$,$f_{i,total}^{p,max}$, and $f_{i,total}^{p,min}$ are the best, maximum, and minimum values of objective function *i* at each iteration, respectively. Finally, the normalized cost value of each imperialist (C_n) is calculated by:

$$C_n = \sum_{i=1}^{r} C_{i,n} \qquad (18)$$

Where *r* is the number of objective function, the power of each imperialist is calculated after obtaining the normalized cost as described below, and the colonies are distributed among the imperialist according to power of each imperialist country.

$$power_n = \left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \qquad (19)$$

Then, the initial number of colonies of an empire will be as follows:

$$NCOL_n = round\{power_n \times N_{col}\} \qquad (20)$$

Where $NCOL_n$ is the initial number of colonies of the *n*-th imperialist, and N_{col} is the number of all colonies. We select $NCOL_n$ of colonies randomly, and all of them are allocated to each imperialist. Imperialist with greater power has a bigger number of colonies, while imperialist with weaker power has fewer.

4.2.2. Moving the colonies of an empire toward the imperialist (assimilating)

After dividing colonies between imperialists, colonies move toward their related imperialist. This movement is illustrated in Fig. 1, in which X is the distance between colony and imperialist. α is a random variable with a uniform distribution between 0 and $\beta \times X$, and β is a number greater than 1. Direction of the movement is shown by θ, which is a uniform distribution between $-\gamma$ and γ .

4.2.3. Crossover between colonies

Moreover, colonies share their information together by crossover to improve their fitness. The best colonies have more chance than others to share their information, because colonies are selected in this section by tournament selection. The population percent sharing information is shown by p Crossover.

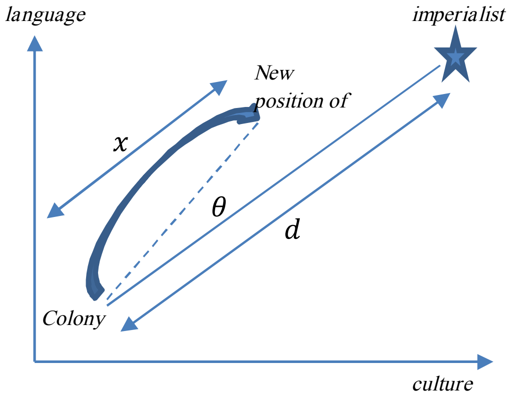

Fig. 2. Moving colonies toward the imperialist with a random angle θ

4.2.4. Exchanging positions of the imperialist and a colony

First, each imperialist with the best colony in terms of the crowding distance in the front one of related colonies is compared together. If the imperialist is not dominated by this colony, this comparison is continued to the last colony in front one. If any colony does not find in front one to dominate imperialist, the imperialist is added to the front one. Front one is sorted in terms of the crowding distance, and solution with a high crowding distance is selected as imperialist.

4.2.5. Total power of an empire

The total power of an empire is mainly affected by the power of the imperialist country; but, the power of the colonies of an empire has an indigent effect on the total power of that empire. Therefore, the equation of the total cost is shown below. (Karimi et al. 2010, Shokrollahpour et al. 2011).

$$TC_n = (Cost\ (imperialist_n) + \xi\ mean\ \{Cost(colonies\ of\ empire_n)\}) \tag{21}$$

Where TC_n is the total cost of the nth empire, and (ξ) is a positive number considered to be less than 1. Also, the costs of imperialists and colonies are calculated by Eqs.17 and 18. The total power of the empire is determined by just the imperialist when the value of (ξ) is small, and increasing it will increase the role of the colonies in determining the total power of an empire.

4.2.6. Imperialistic competition

The power of a weaker emprise will reduce, and the power of more powerful ones will increase in the imperialistic competition. All empires compete with each other to take the possession of the weakest colony of the weakest empire. In other words, first, some (usually one) of the weakest colonies of the weakest empire are chosen, and then the possession of these colonies (or this colony) is given to the winner imperialist among all empires in the imperialistic competition. In this competition, the most powerful empires will not definitely possess these colonies; but, these empires will be more likely to possess them. This competition is modeled by just selecting one of the weakest colonies of the weakest empires, and then for calculating the probability of possession of each empire, first, the normalized total cost is obtained as follows:

$$NTC_n = \max_i\{TC_i\} - TC_n \tag{22}$$

Where NTC_n is the normalized total cost of nth empire, and TC_n is the total cost of nth empire. Having the normalized total cost, the probability of possession of each empire is calculated by:

$$P_{P_n} = \left|\frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i}\right| \tag{23}$$

Then, the roulette wheel method is used for assigning the mentioned colony to one of empires.

4.2.7. Revolution

In each decade, revolutions are performed in some of the colonies and all of the imperialists. The revolution rate in this paper is shown by PRevolution.

4.2.8. Archive adaption

Ranking and sorting is done by the non-dominated and crowding distance for each empire. Then, the members of front one of each empire are selected in order to be added to the archive. Finally, the members of front one are kept, and others are deleted after ranking and sorting solutions in the archive.

4.2.9. Eliminating the powerless empires

Powerless empires will collapse and their colonies will be distributed among other empires in the imperialistic competition. In this paper, when an empire loses all of its colonies, we consider that it collapses.

4.2.10. Stopping criteria

In this paper, the stopping criteria or end of imperialistic competition is considered when there is only one empire between all of countries.

5. Experimental Results

The performance of the proposed MOICA is compared with the well-known multi-objective evolutionary algorithm, namely NSGA-II. Two algorithms studied in this paper are coded using MATLAB 7.13 and run on a personal computer with a 2.26 GHz CPU and 3 GB main memory.

As already mentioned, the test problems were comprised of small- and large-sized problems as in Table 4. In this paper, the Taguchi method is applied to both scales for parameter tuning. Parameter tuning by the Taguchi method is explained in detail by representing the step-by-step results for the large-sized problems, while the obtained results from small-sized problems are also reported. In each size, five test problems are chosen randomly.

Table 4
Description of the test problem

Problem number	Project file name at (PSPLIB)	Number of activities	Size of problem	Number of resources	Number of modes
1	j1227-8.mm	12	Small	3	2
2	j1227-9.mm	12	Small	3	2
3	j1227-10.mm	12	Small	3	2
4	j1228-1.mm	12	Small	3	2
5	j1228-2.mm	12	Small	3	2
6	j189-1.mm	18	Large	3	2
7	j189_2.mm	18	Large	3	2
8	j189_3.mm	18	Large	3	2
9	j189_4.mm	18	Large	3	2
10	j189_5.mm	18	Large	3	2

5.1. Experimental design

5.1.1. Taguchi parameter design

To tune the algorithm parameters, there exist some statistical approaches for designing examination. Taguchi improved a group of slight factorial experiments matrices in such a way that after several trials, the number of experiments for a given problem can be reduced. Among several experimental design techniques, the Taguchi method has been successfully applied for a systematic approach for optimization. The Taguchi method uses an orthogonal array to organize the experimental results. In this method, orthogonal arrays are used to study several variables using a few experiments. Taguchi classified factors into two main classes: controllable factors and noise factors, which cannot be controlled directly. When eliminating noise factors is not possible, Taguchi method minimizes noise effects and optimizes the level of substantial controllable factors. Taguchi converts iterative data to values considered as a criterion for changes in results. This conversion is the ratio of *S/N*. *S* represents desirable values, *N* represents undesirable values, and the goal is to maximize this ratio. In other words, Taguchi analyzes changes using the desirable ratio of *S/N*. The desirable ratio of *S/N* in this paper is as follows:

$$S/N_T = 10\ log\left(\frac{\bar{y}^2}{s^2}\right) \qquad (24)$$

The goal of this study is to determine the appropriate parameters of MOICA and NSGA-II as inner variables for obtaining optimal solution. Taguchi technique is used to tune the population size, cross over rate, mutation rate, and number of generation in NSGA-II and to tune the number of countries, the number of imperialists,
revolution rate, and assimilation rate in MOICA. This method is used for four factors and three levels, such that factors are algorithms' parameters. Tables [5],[6] show the value of factors in each level for MOICA and NSGA-II in which numbers 1, 2, and 3 are the levels of each factor. All values shown in Table [5],[6] have been determined by some research proposals and some experiments. On the other hand, to select the appropriate orthogonal array, it is necessary to calculate the total degree of freedom for the total mean and two degrees of freedom for each factor with three levels ($2 \times 4 = 8$). Thus, the sum of the required degrees of freedom is $1 + 2 \times 4 = 9$. Therefore, the appropriate array must have at least 9 rows. Table 7 shows orthogonal array with four factors and three levels. So, this array is L9.

Table 5
parameters and their levels for MOICA Problems

Factor	Symbol	Level 1	Level 2	Level 3
$\boldsymbol{Num.cou}$	A	100	150	200
$\boldsymbol{Num.imp}$	B	10	15	20
$\boldsymbol{p_m}$	C	0.25	0.3	0.35
$\boldsymbol{assimilation\ rate}$	D	0.25	0.3	0.35

Table 6
parameters and their levels for NSGA-II Problems

Factor	Symbol	Level 1	Level 2	Level 3
$\boldsymbol{npop}$	A	100	150	200
$\boldsymbol{Max\ gen}$	B	50	75	100
$\boldsymbol{p_c}$	C	0.95	0.85	0.8
$\boldsymbol{p_m}$	D	0.1	0.2	0.3

Table 7
The orthogonal array L9

Experiment number	*A*	*B*	*C*	*D*
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

As it is mentioned before, problems considered in this study are classified into large and small groups in which five problems are selected randomly. To obtain more reasonable results, each problem is examined for four times. So, we can obtain 20 results for each Taguchi examination. It should be noted that MID approach is used to determine the most suitable examination in tuning the parameters of MOICA and NSGA-II. The results obtained from Taguchi are converted to *S/N* ratio. The optimal level of each factor in NSGA-II is illustrated in figure 3. Clearly, the most suitable levels of this factor are A(1), B(2), C(3), and D(3), respectively.

From figure 3 and the value of delta in table 8, we can conclude that $npop$ has the greatest effect on NSGA-II; p_c , $Max\ gen$, and p_m are other impressive factors, respectively. The optimum level of factors above is mentioned in table 10.

To tune the parameters of MOICA, the Taguchi results are also converted into *S/N* ratio. Figure 4 shows the optimal level of each factor for this algorithm. From these graphs, it can be concluded that the most suitable levels of this factor are A(2), B(3), C(1), and D(2), respectively.

From figure 4 and the value of delta in table 9, we can conclude that p_m has the greatest effect on MOICA; nimp ‹npop, and as rate are other impressive factors, respectively. Optimum level of factors above is mentioned in table 10.

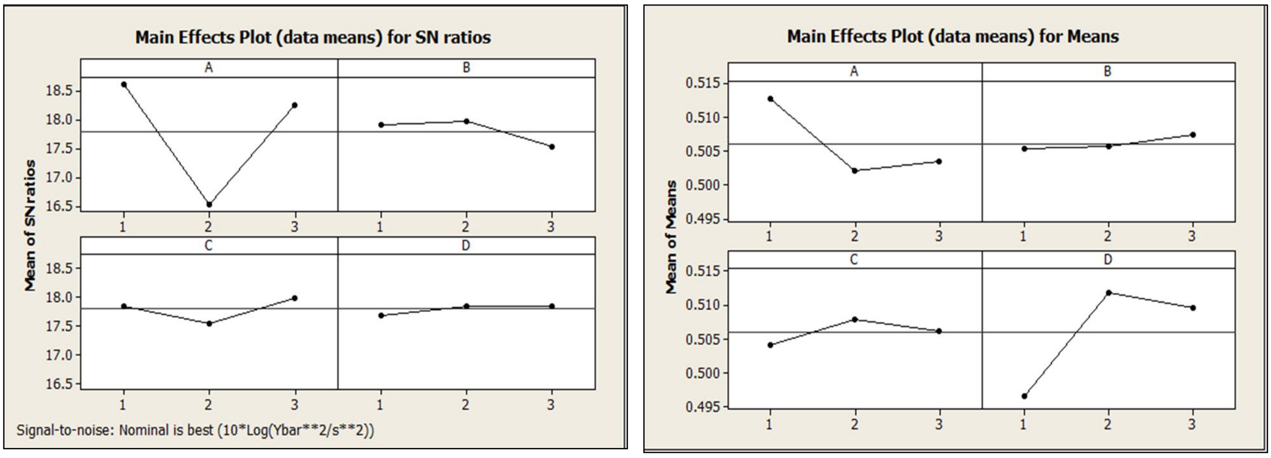

Fig. 3. Mean S/N ratio for level of each factor NSGA-II

Table 8
S/N NSGAII

Response table for signal to noise ratios (nominal is better)				
Level	*A*	*B*	*C*	*D*
1	18.62	17.90	17.86	17.70
2	16.53	17.96	17.55	17.85
3	18.25	17.53	17.99	17.85
Delta	2.09	0.43	0.44	0.16
Rank	1	3	2	4

Table 9
S/N MOICA

Response table for signal to noise ratios (nominal is better)				
Level	*A*	*B*	*C*	*D*
1	17.15	16.73	18.21	16.88
2	17.36	16.28	16.60	17.47
3	16.51	18.01	16.21	16.66
Delta	0.85	1.73	2.00	0.81
Rank	3	2	1	4

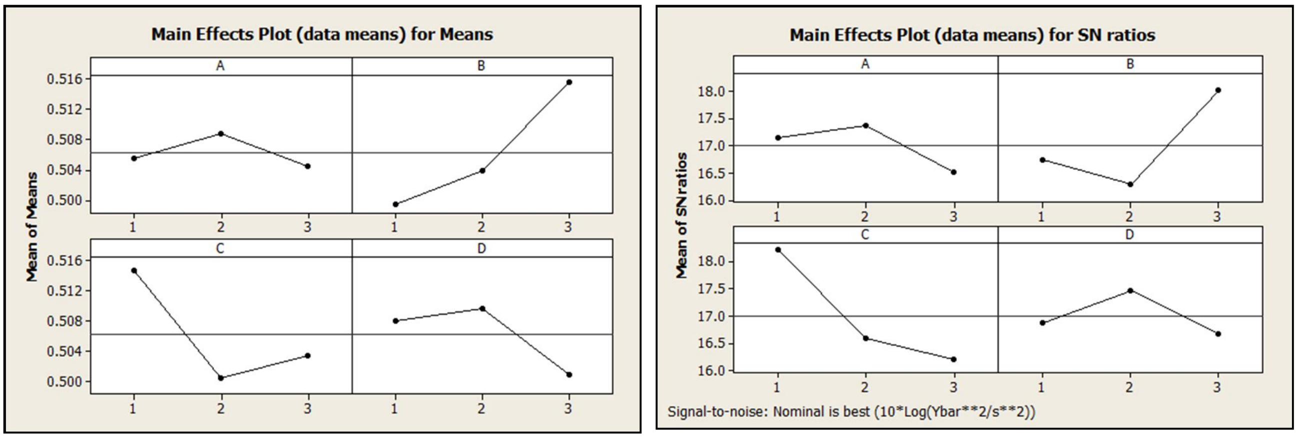

Fig. 4. mean S/N ratio for level of each factor MOICA

Table 10
Parameter setting values

Factor	Symbol	Optimal level for each problem	Factor	Symbol	Optimal level for each problem
	NSGA-II			MOICA	
$npop$	A	100	$Num.cou$	A	150
$Max\ gen$	B	75	$Num.imp$	B	20
p_c	C	0.8	p_m	C	0.25
p_m	D	0.3	$assimilation\ rate$	D	0.3

5.2. Comparison metrics

To validate the reliability of the proposed MOICA, three comparison metrics are taken into account.

- **Mean ideal distance (MID):** The closeness between Pareto solutions and ideal point (f_1^{best}, f_2^{best}) is determined by using MID. The equation of MID is defined by:

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(\frac{f_{1i}-f_1^{best}}{f_{1,total}^{max}-f_{1,total}^{min}}\right)^2+\left(\frac{f_{2i}-f_2^{best}}{f_{2,total}^{max}-f_{2,total}^{min}}\right)^2}}{n} \qquad (25)$$

Where n is the number of non-dominated solution, and $f_{i,total}^{max}$ and $f_{i,total}^{min}$ are the maximum and minimum values of each fitness functions among all the non-dominated solutions obtained by the algorithms, respectively. Regarding this definition, the algorithm with a lower value of MID has a better performance.

- **The rate of achievement of two objectives simultaneously (RAS):** at first, the ideal point is calculated, and then RAS is obtained by:

$$RAS = \frac{\sum_{i=1}^{n}\left(\frac{\left|f_{1i}-f_1^{best}\right|}{f_{1,total}^{max}-f_{1,total}^{min}}+\frac{\left|f_{2i}-f_2^{best}\right|}{f_{2,total}^{max}-f_{2,total}^{min}}\right)}{n} \qquad (26)$$

Regarding this definition, the algorithm with a lower value of RAS has a better performance.

- **Spacing metric (SM):** This metric measures the uniformity of the spread of the non-dominated solution, and then SM is obtained by:

$$SM = \frac{\sum_{i=1}^{n}\left|\bar{d}-d_i\right|}{(n-1)\bar{d}} \qquad (27)$$

Where d_i is the Euclidean distance between consecutive solutions in the obtained non-dominated set of solutions, and $\bar{d}$ is the average of these distances. Regarding this definition, the algorithm with a lower value of SM has a better performance.

5.3. Comparative results

The proposed MOICA is applied to solve the test problems and its performance is compared with the NSGA-II.

Table 11 lists the values of the above-mentioned comparison metrics, and shows that the proposed MOICA is superior to NSGA-II in each test problem.

As shown in Table 11, the results of two algorithms are very close. However, the MOICA optimal solutions are better in terms of three metrics in comparison with the other algorithm. For example, the obtained result for the 10th problem shows that the MOICA outperforms NSGA-II according to mean ideal distance (MID) metric with a value of 0.4890, while MID is equal to 0.4941 for the NSGA-II (the lower the MID, the better). MOICA also excels based on the rate of achievement of simultaneous objectives' (RAS) metric with a corresponding value of 0.4880 (the lower is better). Finally, it is superior, according to the spacing metric, SM=0.4508 (the lower, the better). Similarly, the outcomes of MOICA in the other test problems show that the metrics for the algorithm are more favorable compared to the NSGA-II.

Table 11
Computational results of metrics for the algorithms

Problem/metric	MID metric		RAS metric		Spacing metric	
	NSGA-II	MOICA	NSGA-II	M0ICA	NSGA-II	MOICA
Problem 1	0.4555	0.5097	0.4555	0.5098	0.2683	0.6010
Problem 2	0.4948	0.4948	0.4948	0.4948	0.0322	0.0322
Problem 3	0.4628	0.4170	0.4629	0.4170	0.2052	0.3336
Problem 4	0.5673	0.5673	0.5074	0.5674	0.4000	0.4000
Problem 5	0.6066	0.6066	0.6067	0.6067	0.7404	0.7404
Problem 6	0.5410	0.5382	0.5410	0.5282	0.4446	0.5422
Problem 7	0.4773	0.4875	0.4773	0.4875	0.4332	0.5633
Problem 8	0.4997	0.5365	0.4997	0.5365	0.6338	0.6550
Problem 9	0.3092	0.3144	0.3092	0.3144	0.9418	0.7897
Problem 10	0.5270	0.4186	0.5270	0.4186	0.4085	0.4085

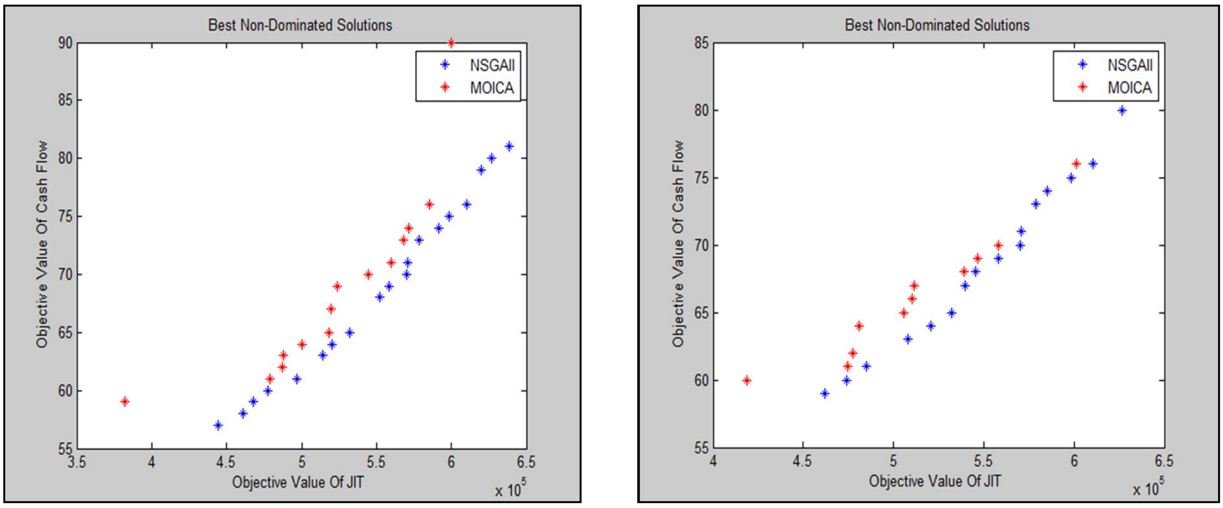

Fig. 5. Non-dominated solution for each algorithm

6. Conclusion

This paper presented a new multi-objective imperialist algorithm (MOICA) for solving a bi-objective preemptive multi-mode resource-constrained project scheduling problem based on minimizing makespan and maximizing net present value. To validate the proposed MOICA, a number of test problems were designed to evaluate the performance and reliability of the proposed algorithm in comparison with one well-known multi-objective evolutionary algorithm, called NSGA-II. In addition, some useful comparison metrics (i.e., spacing, MID, and RAS metrics) were applied. The results of this study indicate that the proposed MOICA outperforms NSGA-II and is able to improve the quality of the obtained solutions.

For future research, at least three issues are worth investigating. Firstly, it would be interesting to generalize the present model to include nonrenewable resources as well as multiple execution modes for each activity. Secondly, it appears that the number of efficient solutions increases with the number of activities. Finally, it is recommended that the problem be solved using other heuristic and meta-heuristic algorithms. The results can be compared with the ones of this study.

Acknowledgements

I would like to acknowledge the support received from Quchan University of Advanced Technology, Quchan, Iran.

References

Alcaraz, J., Maroto, C. & Ruiz, R., (2003).Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, 54, 614–626.

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

Atashpaz-gargari, E., & lucas, C, (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialist competitive. IEEE Congress on evolutionary computation, Singapore.

Boctor, F., (1996).An adaption of the simulated annealing for solving resource constrained project scheduling problems. *International Journal of Production Research*, 34, 2335–2351.

Bouleimen, K. & Lecocq, H., (2003).A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode versions. *European Journal of Operational Research*, 149, 268–281.

Buddhakulsomsiri, J. & Kim, D., (2006). Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 175, 279–295.

Ballestin, F., Valls, V.& Quintanilla, S. (2008). Preemptive in resource-constrained project scheduling. *European Journal of Operational Research*,189, 1136–1152.

Ballestin,F., & Blanco, R., (2011). Theoritical and practical fundamentals for multi-objective optimization resource constrained project scheduling problem,Applied Mathematics and computation, 38,51-62.

Coelho,j. & vanhouvke, M., (2011). Multi-mode resource constrained project scheduling problem using RCPSP and SAT solvers, *European journal of Operational Research*, 213, 73-82.

Drexl,A.&Grünewald,J.,(1993).Nonpreemptive multi-mode resource-constrained project scheduling. IIE Transactions, 25, 74–81.

Demeulemeester, E. & Herroelen, W., (1996). An efficient optimal procedure for the preemptive resource-constrained project scheduling problem. *European Journal of Operational Research*, 90, 334–348.

Damay, J.& Quilliot, A. & Sanlaville, E., (2007). Linear programming based algorithms for Preemptive and non-preemptive rcpsp. European Journal of Operational Research, 182, 1012–1022.

Dieter D., Bert. D.r., Roel.L.& Mario.v.,(2006). A hybrid scatters search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, 169, 638-653.

Deblaere,F.,Demeulemwwster, E. & Herroelen,W., ,(2010). Reactive scheduling in the multi-mode RCPSP, *computers & Operation Research*, 380, 63-74.

Doersch RH. & Patterson JH, (1997). Scheduling a project to maximize its present value: a zero-one programming *approach. Management science*, 23(8), 882-890.

Deb,K.,Agrawals., Pratap, A. & Meyarivan,T. (2000).A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.In M.S.et al.(Ed.),Parallel Problem solving from Nature- PPSN VI,Berlin, springer ,849-858.

Deb,k., Pratap,A.,Agarwal,S., & Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.IEEE *Transaction on Evolutionary Computation*, 6(2),182-197.

Goto, E., Joko,T., Fujisawa,K., Katoh,N. & Furusaka,S., (2001). Maximizing Net Peresent value for Generalized Resource Constrained Project Scheduling Problem, Technical Paper,Nomura Research institute, Department of Architecture and Architectural System, Kyoto,Japan ,

Hartmann, S. & Sprecher, A., (1996). A note on hierarchical models for multi-project planning and scheduling, *European Journal of Operational Research* ,94, 377– 383.

Hartmann, S. & Drexl, A., (1998).Project scheduling with multiple modes: A comparison of exact algorithms. Networks, 32, 283–297.

Hartmann, S., (2001). Project scheduling with multiple modes: A genetic algorithm. Annals of Operations Research ,102, 111–135.

Hartmann, S. & Briskorn, D., (2010). A survey of variants and extensions of the resource-constrained project scheduling problem, 207, 1-14.

Kaplan, L., (1988). Resource-Constrained Project Scheduling with Preemptive of Jobs. Ph.D. Thesis, University of Michigan.

Kolisch, R., Drexl, (1997). A Local search for nonpreemptive multi-mode resource constrained project scheduling.IIE Transactions, 29, 987–999.

Karimi, N., Zandieh, M., & Najafi, A.A. (2010). Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism, *International Journal of Production Research,* 49 (16), 4965-4977.

Lova, A., Tormos, P., Cervantes, M. & Barber, F., (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, 117 (2), 302–316,

Lova, A., Tormos, P., Barber, F., (2006). Multi-mode resource constrained project scheduling: Scheduling schemes, priority rules and mode selection rules. *Inteligencia Artificial*, 30, 69–86.

Mori, M. & Tseng, C., (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. *European Journal of Operational Research*, 100,134–141.

Mika M., Waligora G.&Weglarz j. (2005). Simulated annealing and tabu search for multi-mode resource constrained project scheduling problem with positive discounted cash flows and different payment models, *European journal of Operational Research,* 164, 639-667.

Nonobe, K. & Ibaraki, T., (2001). Formulation and Tabu Search Algorithm for the Resource Constrained Project Scheduling Problem (RCPSP).Technical Report, Kyoto University.

Ranjbar, M., De Reyck, B. & Kianfar, F., (2008). A hybrid scatter-search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, 193 (1), 35–48.

Russell, AH. (1970). Cash Flows in networks, *Management Science*, 16(5), 357-73.

Slowinski, R., (1980). Two approaches to problems of resource allocation among project activities – A comparative study. *Journal of Operational Research Society*, 8, 711–723.

Speranza, M. & Vercellis, C., (1993). Hierarchical models for multi-project planning and scheduling. *European Journal of Operational Research*, 64, 312–325.

Sprecher, A., Hartmann, S. & Drexl, A., (1997). An exact algorithm for the project scheduling with multiple modes. OR Spektrum, 19, 195–203.

Sprecher, A. & Drexl, A., (1998). Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*,107,431–450.

Shokrollahpour, E., Zandieh, M., & Dorri, B., (2011). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flow shop problem. *International Journal of Production Research*, 49(11),3087-3103.

Ulusoy G. & Sivikaya-Serifoglu F., (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. & Debels, D., (2008). The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Computers and Industrial Engineering*, 54, 140–154.

Weglarz, J., Jozefowska, J., Mika, M. & Waligora, G., (2011). Project scheduling with finite and infinite number of activity processing modes - A survey, EJOR 208, 177-205.

Zhu, G., Bard, J. & Tu, G., (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *Journal on Computing*, 18 (3), 377–390.

Zhang, H., Tam, C. & Li, H., (2006). Multimode project scheduling based on particle swarm optimization. Computer-*Aided Civil and Infrastructure Engineering*, 21, 93–103.

Zahra sadat, H., Javad , H. & Emad, R., (2014). A Bi-objective Pre-emption Multi-mode Resource Constrained Project Scheduling Problem with due Dates in the Activities. *Journal of optimization in industrial Engineering*, 15 -15-25.

This article can be cited: Hassanpour, J., Ghodoosi, M. & Hosseini, Z.H. (2017). Optimizing a Bi-objective Preemptive Multi-mode Resource-Constrained Project Scheduling Problem: NSGA-II and MOICA Algorithms. *Journal of Optimization in Industrial Engineering*. 10(21), 79-92.

URL: http://qjie.ir/article_263_37.html