Journal of Optimization in Industrial Engineering, Vol. 11, Issue 1, Winter and Issue 2018, 169-183
DOI: 10.22094/JOIE.2017.592.1381

Modeling and Solution Procedure for a Preemptive Multi-Objective Multi-Mode Project Scheduling Model in Resource Investment Problems

Mostafa Salimi [a], Amir Abbas Najafi [b,*]

[a] *M.Sc, Department of Industrial Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b] *Associate Professor, Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran*
Received 20 February 2016; Revised 14 December 2016; Accepted 20 November 2017

Abstract

In this paper, a preemptive multi-objective multi-mode project scheduling model for resource investment problem is proposed. The first objective function is to minimize the completion time of project (makespan); the second objective function is to minimize the cost of using renewable resources. Non-renewable resources are also considered as parameters in this model. The preemption of activities is allowed at any integer time units, and for each activity, the best execution mode is selected according to the duration and resource. Since this bi-objective problem is the extension of the resource-constrained project scheduling problem (RCPSP), it is NP-hard problem, and therefore, heuristic and metaheuristic methods are required to solve it. In this study, Non-dominated Sorting Genetic AlgorithmII (NSGA-II) and Non-dominated Ranking Genetic Algorithm (NRGA) are used based on results of Pareto solution set.We also present a heuristic method for two approaches of serial schedule generation scheme (S-SGS) and parallel schedule generation scheme (P-SGS) in the developed algorithm in order to optimize the scheduling of the activities. The input parameters of the algorithm are tuned with Response Surface Methodology (RSM). Finally, the algorithms are implemented on some numerical test problems, and their effectiveness is evaluated.

Keywords: Resource investment problem (RIP), Preemption, Serial and parallel schedule generation scheme (SGS), NSGA-II, NRGA, Response surface methodology (RSM).

1. Introduction

The resource-constrained project scheduling problem (RCPSP) includes activities that must be scheduled, which are subject to finish-to-start type precedence relations and renewable resource constraint in order to minimize the makespan.This is a known combinatorial optimization problem, which is NP-hard Blazewicz et al. (1983). The objective of RCPSP is to minimize the makespan of the project while availabilities of the renewable resources are considered given. In the literature, there are several exact methods and heuristics via whichRCPSP can be solvedKolisch et al. (2006) Zhang et al. (2006) Montoya-Torres et al. (2010) Hartmann & Briskorn (2010); Agarwal et al. (2011) Fang & Wang (2012) Kone, (2012) Paraskevopoulos (2012).

In multi-mode version of the RCPSP (MRCPSP), each activity can be performed in one out of a set of modes with a specific activity duration and resource requirements. The problem involves the selection of a mode for each activity and the determination of the activity's start or finish times, such that project makespan is minimized. However, the precedence and resource constraints should be satisfied. MRCPSP,as generalization of the RCPSP, is also NP-hard. In recent years, several algorithms have been proposed to solve MRCPSP Zhu et al. (2006) Zhang et al.(2006); Lova et al.(2006) Jarboui et al. (2008) Ranjbar et al. (2009); Coelho & Vanhoucke, (2011) Ranjbar, (2011) Barrios et al., (2011) Afshar-Nadjafi et al .(2013) Afruzi et al. (2013).

The resource investment problem (RIP) is a close variant of RCPSP. This problem consists of determination of the activities' start times and renewable resources' availabilities, such that the total cost of the resources is minimized subject to a given project deadline. In RIP, project's makespan is not forgotten, but it is controlled with a predetermined deadline as a constraint. This problem was introduced Mohring (1984). He showed that the problem is NP-hard. Rangaswamy [Rangaswamy, 1998] proposed a B&B for the RIP and applied it to the same instance set used Demeulemeester (1995). Drexl and Kimms (2001) proposed two lower bound procedures for the RIP based on Lagrangian relaxation and column generation methods. Other exact procedures wereproposedDemeulemeester, (1995) Rodrigues et al.(2010) Yamashita, et al.(2006) proposed a multi-start heuristic based on the scatter search. Shadrokh and Kianfar (2007) presented a genetic algorithm (GA) to solve the RIP when tardiness of the project is permitted with penalty. Ranjbar et al. developed a path relinking procedure and a genetic algorithm (GA) Ranjbar et al. (2008). Van Peteghem and Vanhoucke (2013) presented an artificial

*Corresponding author Email address: aanajafi@kntu.ac.ir

immune system algorithm for the problem.
The basic project scheduling problems assume that each activity, once started, will be executed until its completion. Preemptive project scheduling problem refers to the scheduling problem which allows activities to be preempted at any time instance and restarted later on at no additional cost or time. In caseswithpreemptiveactivities, modeling and solving such a problem as a classical non-preemptive RCPSP or RIP may lead to poor solutions. Such situations where preemption of an activity is beneficial or necessary are typical, for example, in industry processes where processing units, like reactors or filters, have to be cleaned after the completion of certain subactivities. In the textile industry, preemptions are inevitable. When the fabric type is changed on a machine, the wrap chain must be replaced which indicates the necessity of preemption. The literature on solution methods for the preemptive project scheduling problems is relatively scant. For the preemptive RCPSP, we refer to Kaplan, (1998) Demeulemeester & Herroelen, (1996) Ballestin et al. (2008) Vanhoucke & Debels, (2008) Damay, (2007). For the preemptive MRCPSP, Buddhakulsomsiri and Kim (2006) proved that preemption is very effective to improve the optimal project makespan in the presence of resource vacations and temporary resource unavailability. Van Peteghem and Vanhoucke (2010) have proposed a genetic algorithm for the MRCPSP and its extension to the preempted case. AfsharNajafi and Arani (2014) proposed a model which addresses the preemptive multi-mode resource investment problem with the objective of minimizing the total renewable/nonrenewable resource costs and earliness-tardiness costs by a given project deadline and due dates for activities in order to reach a reasonable procedure time. Genetic algorithm is used to solve the model, and some pre-processingsareused to increase the efficiency of the algorithm and quality of solution. For instance, this approach is used in order to remove inefficient execution modes and reduce the search space. Majid Tavana et al. (2014) proposed a multi-mode model in which a discrete time-cost-quality trade-off problem with activity preemption is extensively investigated. In their model, optimization and time-cost-quality trade-off is done considering generalized precedence relations.
There are some shortcomings in the basic RIP which is considered in this paper simultaneously. First, activities in the basic RIP are assumed non-preemptive, while this assumption is not true in practice. Second, in the basic RIP, the determination of availabilities of only renewable resources is investigated. Third, the basic RIP supposes single execution mode for activities. At last, minimizing the time and cost as two objectives simultaneously is not considered.Therefore, the contribution of this paper is fourfold: first, a mixed integer programming formulation is developed for the multi-objective multi-mode preemptive RIP. We call this problem P-MMRIPSP. This model is not considered in the past literature. Second, minimizing the time and cost is considered as two objectives simultaneously. Third, a new efficient parameter-tuned NSGA-II is developed for the problem due to NP-hardness of the problem. Finally, the effectiveness of the proposed method to solve the P-MMRIPSP is analyzed statistically.
The reminder of the paper is organized as follows. Section 2 describes the problem. Section 3 explains the steps of the proposed Non-dominated Sorting Genetic Algorithm (NSGA-II) to solve the problem. Section 4 describes comparison metrics. Section 5 contains the computational results and performance evaluation of the proposed NSGA-II. Finally, Section 6 concludes the paper.

2. Notation and Problem Description

The preemptive multi-objective multi-mode resource investment project scheduling problem (P-MMRIPSP) involves the scheduling of project activities on a set K of renewable resource types and a set W of nonrenewable resource types. Each activity i is performed in mode m_i, which is chosen out of a set of M_i different execution modes, that is, with different durations and resource requirements. The duration of activity i, when executed in mode m_i, is d_{im_i}. Each activity i in mode m_i requires $r^{\rho}_{im_ik}$ units of renewable resource type k (k = 1 ,..., K) during each time unit of its execution. For each renewable resource k, the availability r^{ρ}_{k} is constant throughout the project horizon. Activity i, executed in mode m_i, will also use $r^{\nu}_{im_iw}$ nonrenewable resource units (w = 1 ,..., W) of the total available nonrenewable resource R^{ν}_{w}.
In sequent, assume a project represented in AON format by a directed graph $G = \{N,A\}$ where the set of nodes, N, represents activities, and the set of arcs, A, represents finish-start precedence constraints with a time lag of zero. The pre-emptible activities are numbered from the dummy start activity 0 to the dummy end activity n+1 and are topologically ordered, that is, each successor of an activity has a larger activity number than the activity itself.
Also, we consider discrete time points for the preemption. As a rule, a preemptive problem is characterized by a complicated structure of its optimal solutions. When preemption is allowed at arbitrary times, the problem turns out to be intractable Bulbul (2007). In this situation, when the overall number of preemptions is unlimited, and the set of admissible points for preemptions has continuous cardinality, we cannot utilize exact enumerative algorithms, unless a nontrivial preliminary analysis of properties of optimal solutions is performed. Such an analysis reduces the original infinite set of possible points for preemption to a finite set. This reduction allows us to solve the problem by direct enumeration. This analysis is performed for some scheduling problems. Baptiste et al. proved that a wide class of preemptive scheduling models, including both machine and project scheduling models, have the "integer preemption property"; for any problem instance with integral input data, there exists an optimal schedule where all preemptions (as well as starting and completion times of jobs/activities) occur at integer time points. This conclusion is held for objective functions such as total weighted earliness-tardiness and total weighted number of late jobs (Baptiste et al., 2009; Baptiste et al., 2011).
The objectives of the P-MMRIPSP are to schedule a

number of activities in order to minimize the makespan and total cost of the renewable resources. Schedule S is defined by a vector of activity start times and is said to be feasible if all precedence relations and renewable and nonrenewable resource constraints are satisfied. However, execution modes m_i, available resource capacities, and start times for activities have to be determined. The resulting schedule may be transferred into a schedule for the original problem by removing the dummy start and end activity.
that the corresponding precedence relations are adjusted appropriately. The resulting schedule may be transferred into a schedule for the original problem by removing the dummy start and end activity, and one-time unit left shifting.
The P-MMRIPSP model concurrently minimizes the makespan and renewable resource cost of the project. The notations used to formulate the P-MMRIPSP model are

2.1. Mathematical formulation:

It is clear that an activity with duration of 0 is never in progress, and thus does not have a corresponding decision variable which is set to 1. This problem, however, can be easily overcome; the dummy start and end activity are assigned a dummy mode with duration of 1. Also, the other parameters for dummy modes are assumed 0. All other activities with zero duration can be eliminated, provided presented in Table 1.
The following multi-objective mixed integer non-linear mathematical programming formulation is proposed for the P-MMRIPSP model:

$$\text{Min Makespan} = S^{Max}{}_{(n+1)} \quad (1)$$

$$\text{Min Resources Cost} = \sum_{k=1}^{K} C^{\rho}{}_{k} R^{\rho}{}_{k} \quad (2)$$

Table 1
The parameters, notations, and decision variables.

	Parameters and Notation
A	Set of arcs of acyclic digraph representing the project
N	Set of nodes of acyclic digraph representing the project, $\|N\| = n$
n	Number of non-dummy activities, index by i
K	Number of renewable resource(s), index by k
W	Number of no-nrenewable resource(s), index by w
Mi	Set of execution modes for activity i, $i \in N$
$\|Mi\|$	Number of execution modes for activity i, index by m
d_{im_i}	Duration of activity i in mode m_i, $i \in N, m_i \in M_i$
$r^{\rho}{}_{im_ik}$	Resource requirement of activity i in mode m_i for renewable resource type k, $k = 1,\ldots, K$, $i \in N$, $m_i \in M_i$
$r^{\nu}{}_{im_iw}$	Resource requirement of activity i in mode mi for nonrenewable resource type k, $w=1,\ldots,W$, $i \in N$, $m_i \in M_i$
λ	A big positive number
$C^{\rho}{}_{k}$	Unitcostofrenewableresourcetype$k$,$k=1,\ldots, K$
$R^{\nu}{}_{w}$	Total availability of nonrenewable resource type w, $k= 1,\ldots,W$
	Variables
$R^{\rho}{}_{k}$	Total availability of renewable resource type k, $k=1,\ldots, K$
$S_{i\,Min}$	Start time of the first time unit of activity i (integer decision variable)
$S_{i\,Max}$	Start time of the last time unit of activity i (integer decision variable)
x_{im_it}	1, if activity i in mode m_i is in progress at time interval $[t, t+1]$, 0, otherwise (binary decision variable).
y_{im_ii}	1, if activity i is executed in mode m_i, 0, otherwise (binary decision variable).

Subject to:

$$\sum_{m=1}^{|M|} x_{im_it} \le 1 \quad , i = 1,2,\ldots,n;\ t = EST_i, \ldots, LFT_i - 1 \quad (3)$$

$$\sum_{t=EST_i}^{LFT_i-1} x_{im_it} = d_{im_i} \times y_{im_i} \quad , i = 1,2,\ldots,n;\ m = 1,2,\ldots,|M| \quad (4)$$

$$\sum_{m=1}^{|M|} y_{im_i} = 1 \quad , i = 1,2,\ldots,n \quad (5)$$

$$\sum_{i=1}^{n}\sum_{m=1}^{|M|} r^{\rho}{}_{im_ik} \times x_{im_it} \le R^{\rho}{}_{k} \quad t = EST_i, \ldots, LFT_i - 1;\ k = 1,2,\ldots,K \quad (6)$$

$$\sum_{i=1}^{n}\sum_{m=1}^{|M|} r^{\nu}{}_{im_iw} \times y_{im_i} \le R^{\nu}{}_{w} \quad , w = 1,2,\ldots,W \quad (7)$$

$$S^{Min}{}_{i} \le t \times x_{im_it} + \lambda(1 - x_{im_it}) \quad i = 1,2,\ldots,n;\ m = 1,2,\ldots,|M|;\ t = EST_i, \ldots, LFT_i - 1 \quad (8)$$

$$S^{Max}{}_{i} \ge t \times x_{im_it} \quad i = 1,2,\ldots,n;\ m = 1,2,\ldots,|M|;\ t = EST_i, \ldots, LFT_i - 1 \quad (9)$$

$$S^{Max}{}_{i} + 1 \le S^{Min}{}_{j}, \forall (i,j) \in P \quad (10)$$

$$x_{im_it}, \quad y_{im_i} \in \{0,1\} \qquad \forall i, m_i, t \quad (11)$$

$$S^{Min}{}_{i}, S^{Max}{}_{i}, R^{\rho}{}_{k} \ge 0 \qquad \forall i, m_i, k \quad (12)$$

The first objective function in (1) is to minimize the

makespan of the project or end of the last dummy activity. The second objective in (2) is to minimize the total cost of the project renewable resource. Equation (3) guarantees that each activity *i* can be in progress with at most one mode in each time unit. Equation (4) ensures that each activity *i* should be in progress for d_{im_i} time units in assigned mode *mi*. Equation (5) specifies that only one execution mode is allowed for each activity *i.* Constraint set in (6) takes care of the renewable resources' limitations. Constraint set in (7) takes care of the nonrenewable resources limitations. Constraint set in (8) computes the start time of the first time unit of activity *i*. Constraint set in (9) computes the start time of the last time unit of activity *i*. Equation (10) denotes the constraints of finish-to-start zero-time lag precedence relations. Equation (11) specifies that decision variables x_{im_it} and $y_{im_i}$are binary. Equation (12) specifies that decision variables $S^{Min}{}_i$, $S^{Max}{}_i$, $R^{\rho}{}_k$ are integers.

2.2.Model validation

To ensure the validation of the mathematical model, several problem instances with less than 5 activities, small amounts of parameters, and simple networks are solved by Lingo software; the validity of the model logic, defined variables, and parameters of the problem are ensured. Since the model of the current study is a multi-objective model, it is not possible to solve it using this software. For this purpose, we use ε-Constraint method in which objective function is transformed into constraints. The first objective, which minimizes project makespan, is transformed into a constraint in the model and is considered as the maximum time of project delivery in constraints.

$$S^{Max}{}_{(n+1)} \leq \varepsilon \qquad (13)$$

In the following, Table 2 presents a problem with 5 activities and two dummy activities (start and end activities) with a network, which is shown in Figure 1, and arbitrary parameters.

Table 2
The parameters of 5 activities example

	d_im_i			$r_im_ik_1$			$r_im_ik_2$			$r_im_iw_1$			$r_im_iw_2$		
i \ m_i	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
1	*3*	*2*	*1*	*1*	*2*	*2*	*1*	*2*	*2*	*2*	*1*	*1*	*1*	*2*	*2*
2	*2*	*4*	*3*	*2*	*2*	*0*	*2*	*1*	*2*	*1*	*2*	*2*	*0*	*0*	*1*
3	*3*	*1*	*2*	*0*	*1*	*2*	*1*	*2*	*2*	*1*	*1*	*2*	*2*	*0*	*2*
4	*2*	*3*	*1*	*1*	*1*	*2*	*2*	*0*	*2*	*2*	*2*	*0*	*0*	*1*	*1*
5	*1*	*2*	*3*	*0*	*1*	*2*	*1*	*1*	*0*	*2*	*1*	*1*	*1*	*2*	*1*

The unit costs of the first and second renewable sources are 10 and 15, respectively.

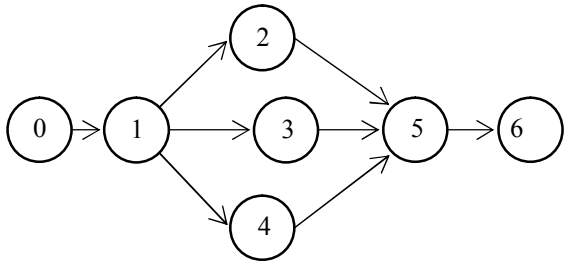

Fig. 1.An example network by 5 actual activities

First, we solve the model using Lingo software by considering the value 4 for ε. Then, we add one unit to ε and run the software again. This process will continue up to ε = 10, and the optimal value of the second objective function and CPU Time isrecorded. Results obtained from these consequent runs are shown in Table 3.

Table 3
CPU time and F2 variable of 5 activities example

Run No.	1	2	3	4	5	6	7
ε	*4*	*5*	*6*	*7*	*8*	*9*	*10*
F_2	*175*	*145*	*130*	*120*	*115*	*115*	*115*
CPU Time	*00:00:03*	*00:05:14*	*00:27:03*	*02:14:55*	*09:02:59*	*20:51:17*	*33:19:34*

As shown in the table, by increasing the value of ε, the value of the second objective function decreases and CPU Time increases extremely in a way that when we set ε equal to 10, Lingo took 19 minutes,34 seconds and 33 hours to reach optimal solution. In this section, the validation of the model is achieved by solving small problems. By increasing the scale of the problem, Lingo and other exact methods will not definitely be able to solve the problem within a reasonable time. Therefore, to solve the proposed model, we should take advantage of other methods that are explained in the following sections.

3. TheProposed NSGA-II for Solving P-MMRIPSP

In this section, we describe the non-dominated sorting genetic algorithm (NSGA-II) which is the well-known metaheuristic that has been successfully applied to a noticeable number of project scheduling problems to solve P-MMRIP. Solution representation, selection, and reproduction are the basic elements of GA. These elements must be well defined and adapted to a specific problem. Before the execution of the proposed NSGA-II, all non-executable and inefficient modes can be omitted in order to reduce the search space Sprecher (1997).Execution mode m_i is called non-executable if its execution would violate the renewable resource constraints in any schedule. Also, a mode is considered inefficient if there is another mode of the same activity with the same or higher duration and no more requirements for all resources.

Also, chromosome structure of the proposed NSGA-II algorithm and its decoding process will be explained. In general, the first essential step in applying genetic

algorithm and its implementation is displaying solutions to problem in form of chromosome. In fact, this step is a key concept in the genetic algorithm that has a great impact on the success and implementation of algorithm. There are various methods for coding solutions of the problem and designing chromosome. Assuming a project with *n*activities, Proposal Displaying in chromosome is done by three separate parts as follows.

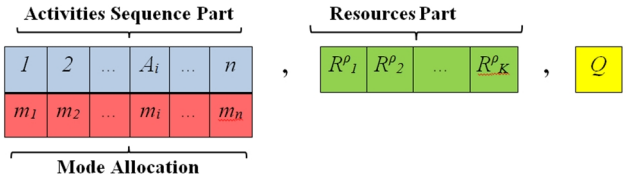

Fig. 2.Chromosome structure

Designed chromosome is composed of three parts. Description of these parts and the number of genes related to each part areas follows.

I. The first part of the chromosome is a matrix with dimension 2×*n*, where n is the number of project activities. Each of the genes in this part of the chromosome takes 1 to *n* values in the first row according to precedence relations; the second row of allocated takes the numbers to execution modes of the related activity in the corresponding entry in the top row, which is a number between one and threein this study.

II. The second part of the chromosome is a matrix with 1×*m*dimension where m is the number of renewable resource.

III. The third part of the proposed chromosome is binary variable *Q* and is used for scheduling generation scheme (SGS) when decoding chromosome. This occursin this way that when *Q* is zero, series-scheduling generation schemeor S-SGS is used, and when *Q* equals one, parallel scheduling generation scheme P-SGS is used in order to schedule activities after cutting activities into a single unit of execution times, which is more fully explained.

$$Q:\begin{cases}1 & \text{Activities Scheduling after conversion to a single time unit by P}-\text{SGS}\\ 0 & \text{Activities Scheduling after conversion to a single time unit by S}-\text{SGS}\end{cases}$$

3.1. Creating an initial population

After determining, a technique which is used to convert any solution to a chromosome is the creation of an initial population of chromosomes. In this stage, initial solution is randomly generated according to the logic of the genetic algorithm, which is dedicated then to each section of the chromosome, that is to say, how creating value and complementing each gene will be explained in detail. Then, how creating value for each part of the chromosome and complementing each gene will be explained in detail.

Feasible sequence and related mode allocation to activity section. In this section, according to the sequence of the generated problem which is explained in the next chapter, we generate a sequence of non-repetitive numbers which indicate the activities, and then put them in the first row of this section. Note that in this stage, no preemption is considered for activities, and they will be executed completely and without preemption or turning into single time units. This part is numbered, so that the first activity, or the number 1, is assigned, and then due to precedence relation of the given problem, the next number, whose predecessor has been coded in the algorithm and will be shown in the attachment related to the algorithm, is checked and randomly placed.This process continues till the last gene (n-th activity). In this section, as previously mentioned, the second row of numbers related to allocation of activity's executive modes pertaining to upper entry will be placed. Since three modes for each activity are considered in this study, then it would simply be carried out. That is, for every gene, a random number between one and three is generated.

Renewable resources. In the second part, in order to give value to genes related to sources, we consider the two concepts of minimum and maximum source levels to achieve a feasible space and better quality solutions, numbers assigned to these genes are also a random number between the range of minimum and maximum levels of resources. For more explanation, it is explained that if the given random number is small in a way that at least one activity could not be executed in any of its execution modes, then the algorithm falls into the loop that will repeat forever, resulting in nothingwhich is equivalent to getting far from the solution space.

Also, if a given number is too large, the quality of the solution is reduced, or it takes longer time to reach a desired solution. To address this fault, a number of procedures and determining the minimum and maximum resource levels redone as follows.

In order to reach the minimum source level, the algorithm scans the execution modes of each activity, and then keeps the smallest need for each resource. This step is performed for each resource and all activity. Afterwards, the values obtained for a resource, e.g., m-th resource, consider the largest number as the minimum number for numbering. This process is done for all resources, and the minimum numbers for each resource are obtained. To find the maximum level of resources in this research, we have selected the floor and ceiling value of the half of the project activities at maximum resource level as the maximum of this range. When the beginning and end of the range of each source are obtained, we set a random number between these ranges of each source in its corresponding genes.

Serial or parallel scheduling generation scheme. In this section, for a simple initialization, we randomly assign a number between zero and one to the related gene. Zero represents the serial schedule-generating plan, and one represents the parallel schedule-generating scheme.

3.2. Chromosome evaluation

As stated in the section of chromosome evaluation in single-objective genetic algorithms, by passing from this state to another in the solution space of the problem, it gradually gets closer to the global optimality point.

Passing from one state to another occurs based on the cost of each state. In genetic algorithm, the cost function could be directly used to estimate the optimality of a state. In fact, the fitness evaluation determines the objective function value considering the constraints of the problem. According to the fact that chromosomes have been correctly designed and generated to satisfy all constraints, no penalty function is required to ensure or satisfy the constraints mentioned in section 2. At this stage the value of a chromosome of the population is obtained by determining criteria. Each chromosome is decoded by using a function which is called fitness function, then a fitness value is considered for each chromosome. Afterwards, some concepts, such as serial scheduling generation scheme(S-SGS) and parallel scheduling generation scheme (P-SGS) are applied in order to obtain an evaluation value for a chromosome. In this study, activities are scheduled in single time unit with regard to the third part of the chromosome, which is previously described, and scheduling scheme of the related chromosome, which is determined by binary values with a feasible sequence of activities in the considered chromosomes. In order to describe the two schemes of S-SGSand P-SGS, an example is presented as follows.

S-SGS. In the literature, one of the decoding processes is S-SGS, presented by Kelly. Assume the following part of a network in which the predecessors of activities 4, 5, and 6 are executed, and all of the predecessors are executed until the 6th day, and activities must be scheduled from the start of the 7th day. Consider the feasible sequence 4,5,6,7 which is obtained by a chromosome. The required resource for each activity is shown on nodes related to each activity in figure 3.

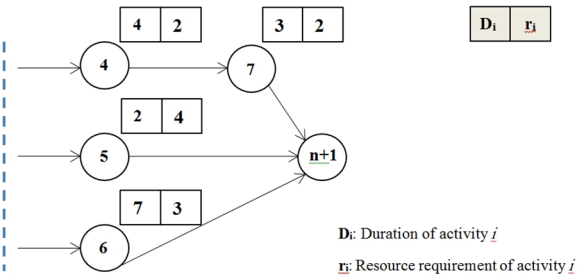

Fig.3.Part of an example network for presented P-SGS and S-SGS

As shown in the above example, activity 4 is set in the earliest possible time, i.e., start of the 7th day. Then, the next activity in the sequence, which is activity 5, is taken into consideration and is set according to the resource level in the earliest possible time, the start of the 11th day. Similarly, the next activity in the sequence, the activity 6, is set in the earliest possible time which is the 13th day according to the available resource level. And finally, the last activity of this example (Activity 7) is set in the earliest possible time, i.e., the13th day, according to the available resources level. This type of scheduling scheme is S-SGS whose Gant chart is presented in figure 4.

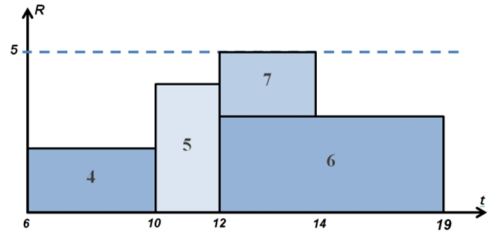

Fig. 4. Activity scheduling by S-SGS scheme

It can be observed that the obtained makespan in this example equals 19.

Scheduling of the above network by applying the proposed algorithm is as follows:

The first activity durations are divided into single time units, and then they are scheduled by using S-SGS. The related Gant chart is illustrated in figure 5.

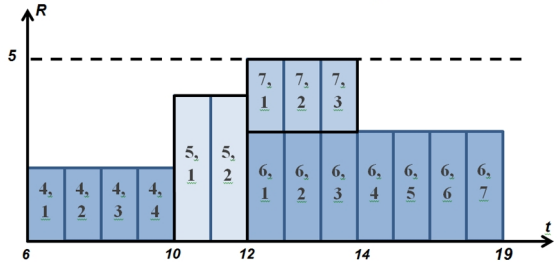

Fig.5. Activities scheduling by applying the proposed S-SGS

As can be seen, the obtained makespanalso equals 19by using this approach.

P-SGS. This scheduling scheme was first presented by Bedworth& Bailey. The difference between this scheme with the previous one is that when we confront an activity of the sequence which we are not allowed to set it in parallel due to resource constraint, we start to scan other activities to the end of the sequence, and each activity which does not violate resource and precedence constraints is set in parallel. For more description of this section, P-SGS is applied tothe previous example.

First, activity 4 placed at the start of 7th day, after the finish of the 10th day available resource equal to three. The next activity, i.e., activity 5, is considered due to four resources which cannot be put at the start of the 7th day, and it is in parallel with Activity 4. This is where the difference of this scheme with the S-SGS is revealed. The rest of the activities, which do not violate the sequence, are scanned by the procedure in this approach. Particularly in this example, activity 6 requiring three resources can be found to be set at the start time of the 7th day due to the resource availability. Afterwards, activity 5 and then activity 7 are set with respect to the considered sequence. Therefore, scheduling the activities is in parallel form. Gant chart of the example is presented in Figure 6.

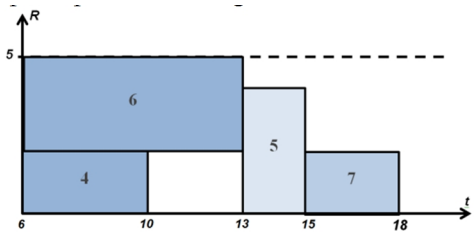

Fig.6. Activities scheduling by P-SGS

It can be observed that the makespan equals 18 by using P-SGS in this example. In order to represent the scheduling approach for the above example by algorithm procedure, the following stepsare as follows: the first activity durations should be divided into single time units, and then scheduled by applying P-SGS. In this example, when the single time units of activity 4 are placed, depending on how the procedure continues, single time units of activity 6 are also placed, and this process continues until the finish time of the 10th day.

Then, Based on the fact that after placement of activities of single time units, the scanning process continues again at the start time of the 11th day due to the availability of resource, and it goes back to activity 5 again, and single time units of this activity are placed. Afterwards, single time units of activities 6 and 7 are successively placed. The related Gant chart is shown in figure 7.

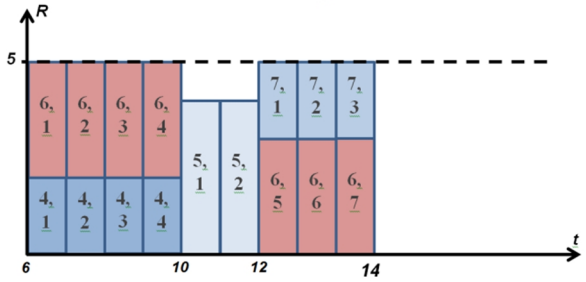

Fig.7. Activities scheduling by applying proposed P-SGS

It could be seen that by using this approach, the obtained makespan is 14. There is a significant difference between this example and the problem whose preemption of activities is not allowed, and that is the shorter makespan with the same resource level. Naturally, it is expected that the last procedure in the proposed algorithm would be preferably selected, because in most cases, it results in a better solution compared to other schemes when the preemption is allowed. This will be discussed in the next chapter.

3.3. Crossover operator

This process is simulated based on the combination of chromosomes during reproduction of living creatures. The two chromosomes are selected according to the strategy explained in the previous section, and the crossover operator used in the proposed NSGA-II algorithm is done separately for each section.

Activities sequence. In this section, single-point and two-point crossover operators are used,suchthatone of the methods israndomly selected first in the proposed algorithm, and then the crossover operator is selected. Each operator is described separately below.

a. Single-point crossover: In this section, when two chromosomes are determined for crossover operation, one point is randomly selected, and both parent chromosomes are cut from that part and divided into two parts. Then, children are produced in such a way that the first child is produced from the first part of the first parent and the second part of the second parent, and the second child is produced from the first part of the second parent and the second part of the first parent as follows.

b.two-point crossover: In this section, after selecting the two parents, two points are randomly selected and both of the parent chromosomes are cut, then the cut between two points on the parent's chromosomes is substituted, and chromosome child is produced. This procedure is depicted in figure 8.

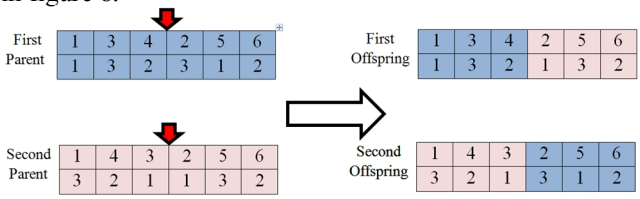

Fig.8. One-point crossover

The problem that arises in this context is that, during this process, an infeasible sequence might be generated, and this infeasibility may include sequence violation in network of the problem or repeating one or two activities in the genes of this part. To address this shortcoming, after chromosome generation, precedence relations in appendix B related to NSGA-II algorithm code in the section of fitness function should be checked.According to "unique" order, if two activities are repeated in a chromosome, one of them is removed and the activity absent in chromosome is randomly replaced by a execution mode. Afterwards, the process of scanning precedence relations of activities is started, and then a feasible sequence would be achieved.

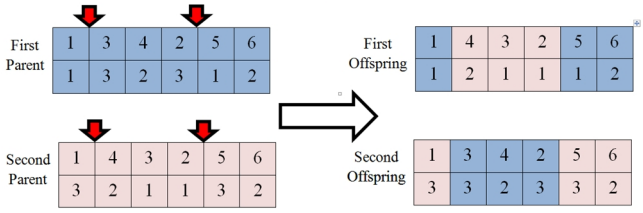

Fig.9.Two-point crossover

*Renewable resources.*In this section, one-point and two-point crossovers described in the previous section are also used. The only problem which we may confront in this section is that when the crossover operation is finished, the obtained values may be out of the specified range for each resource. In order to overcome this shortcoming, a code called CB mentioned in appendix B is used. The performance of this code is described as follows. Aftercrossover operation, the value which is obtained for the resource 2 equals to x. Thus, if the value of x is an infeasible value to the range which has been set for renewable source 2 (i.e., more than upper bound or less than lower bound), by entering equations (14) and (15), we would have a feasible value of x in the considered interval as output.

x=CB(x,lb,ub)

x=max(x,lb) (14)

x=min(x,ub) (15)

According to the procedures described in this section, feasibility of the solution space is guaranteed.

Schedule generation scheme. In this section, since there is only one gene, the process occurs by considering the

probability of 0.5 for selecting crossover operator. In other words, after the selection of the two parents, with probability of 0.5, these two replaced genes and children are produced. The feasibility of the chromosomes remains unchanged in this section.

3.4. Mutation operator

The mutation operator used in the proposed NSGA-II algorithm is applied to three sections of activities sequences, renewable resources, and schedule-generating scheme, which are separately described as follows:

Part of activity sequences. In this section, swap and inversion mutations are applied. In order to describe swap mutation, it should be noted that the two genes are randomly selected first, and then their values are substituted with each other. To better explain the process, figure 10 represents an example of a chromosome in the section of sequence activities and the allocation of the execution modes. For the inversion operator, the two genes in the selected chromosome are chosen, and then the field is reversed between two genes. To better explain the process, an example of a chromosome in the section of sequence activities and the allocation of the execution modes is illustrated in figure 11.

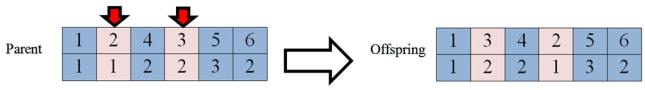

Figure 10: Swap Mutation

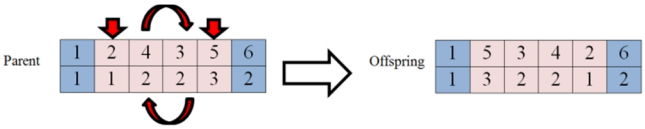

Fig. 11. Inversion Mutation

In this section, sequence in the network of the problem may be violated and leads to infeasible sequence. Hence, by using the same procedure explained for the crossover, this shortcoming are solved, and chromosome children are feasibly applied for evaluation.

Part of renewable resources. In this section, swap and inversion mutation operators are applied. Moreover, to avoid infeasible solution space, the same procedure explained for crossover operator is also used here, and it is made sure that the solution space is feasible after the mentioned mutation operations.

Part of scheduling generation scheme. In this section, because of the existence ofonly one gene, two values of zero and one are replaced, and the chromosome feasibility remains unchanged.

3.5. Children evaluation and combination with parents

In this section, children created through the crossover and mutation operators are evaluated, and a fitness value is assigned to each child. In this part of the algorithm, population of children and parents iscombined, and a population twice the size of the initial population size is created. The combination of solutions avoids losing the best answer among the population of parents and children. Since there are many objective functions in multi-objective optimization problems, elitism problem becomes ambiguous. In such cases, a non-domination ranking is used in a way that each solution can be valued based on non-domination.

3.6. Validation of the proposed algorithm

Finally, in this section, in order to test the validation of the proposed algorithm, the mentioned problem instance in section two is solved using the developed NSGA-II algorithm and compared to solution obtained using Lingo software. The results are presented transparently in table 4 and figure 12.

Table 4
Results of NSGA-II and Lingo for 5 activities

Lingo	F1	4	5	6	7	8	9	10
	F2	175	145	130	120	115	115	115
	CPU TIME	00:00:03	00:05:14	00:27:03	02:14:55	09:02:59	20:51:17	33:19:34
NSGA-II	F1	4	5	7	8	8	8	8
	F2	215	165	130	130	120	120	120
	CPU TIME	00:00:58	00:01:11	00:01:22	00:01:47	00:01:19	00:01:52	00:01:34

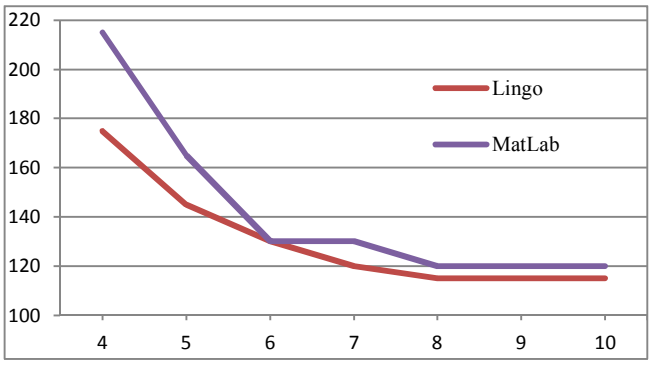

Fig.12.The chart for results of NSGA-II and Lingo

As can be seen, solutions obtained from the proposed algorithm are in much shorter times and very close to the solution obtained from exact methods, and the solution trend indicates the validity of the developed algorithm. In this section, we describe the proposed algorithm and examine its validity.

4. Comparison Metrics

To tune the parameters of multi-objective algorithms,somecriteria should be introduced for algorithms. Generally, unlike single-objective optimization criteria,"diversityof Pareto solutions" and "convergence to the Pareto solution"should be considered in multi-objective optimization criteria,asdescribed (Deb, 2000).In this section, we express quantitative and qualitative comparisons' metrics often used for comparing

metaheuristic algorithms. Fivecomparative criteria to evaluate multi-objective optimization algorithms are presented as follows:

*Maximum Diversity Criteria (DVR).*DVR was presented (Zitzler, 1999). This criterion measures the space cube diameter which is used by the end values of objectives for the set of nan-dominated solution. Equation (16) represents the computational procedure of this index:

$$D = \sqrt{\sum_{j=1}^{m} (\max_i f_i^j - \min_i f_i^j)^2} \qquad (16)$$

For example, the two objective criteria are equal to Euclidean distance between the two border solutions in objective space. Whatever the criteria, the larger, the better.

Spacing criteria (SPC). This criterion, presented by Schott, J (1999), computes the relative distance of consecutive solutions using equation (17) (Schott, 1995).

$$S = \sqrt{\frac{1}{|n|}\sum_{i=1}^{n} (d_i - \bar{d})^2} \qquad (17)$$

where in:

$$d_i = \min_{k \in n \wedge k \neq 1} \sum_{m=1}^{2} |f_m^i - f_m^k| \qquad \bar{d} = \sum_{i=1}^{n} \frac{d_i}{|n|}$$

Measured distance equals the minimum sum of absolute difference in objective functions' values between *i*-th solution and solutions in the final non-dominated set. Note that this distance criterion is different from thatof minimum Euclidean distance between solutions.

The above criteria measure the standard deviation of different amounts of di.

When solutions are uniformly located next to each other, then the value of s will be too small, so an algorithm whose final non-dominated solution has a small amount of spacing will be better.

In order to enhance the readability of the two mentioned criteria, the maximum spread and spacing criteria are shown schematically in figure 13.

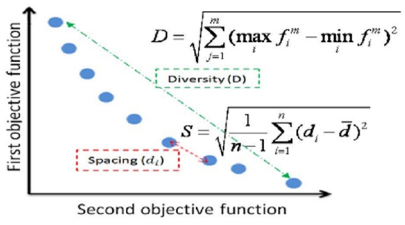

Fig. 13.Maximum Diversity Criteria and Spacing criteria in multi-objective problem

Number of Pareto Solution criteria (NOP).

Value of NOP criteria represents the number of Pareto optimal solutions that can be found in each algorithm. Figure 14 provides an example of calculating the NOP. The larger the criteria are, the better they will be.

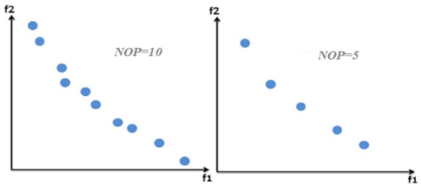

Fig. 14. Number of Pareto Solution in multi-objective problem

Mean Ideal Distance criteria (MID). Figure 15 schematically shows the MID criterion. The smaller the scale is, the better it will be.

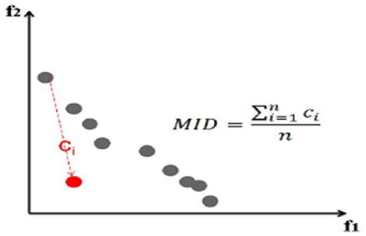

Fig. 15. Mean Ideal Distance in multi-objective problem

CPU Time (CT) criteria.

CPU time is one of the most important performance indicators of each metaheuristic algorithm. The smaller scale is better. As stated before, the
purpose of tuning the input parameters of the proposed depend on the values of input parameters. In this section, the process of tuning input parameters' value of the two NSGA-II and NRGA algorithms is explained.

Parameter tuning of NSGA-II and NRGA

The NSGA-II and NSGA algorithms' input parameters include the maximum number of iterations (MaxIt), population size (nPop), probability of crossover (Pc), and mutation (Pm); each of them has three low, medium, and high levels shown by(1), (0), and (+1), respectively.

Table 5 shows the search range of input parameters' levels of these two algorithms for the presented model in this study.

In this section, in order to implement the procedures of parameter tuning by applying response surface methodology (RSM), central composite design (CCD) with factorial design 2^4, including 8 axial points and 5 central points, is considered. As discussed before, parameters of the multi-objective evolutionary algorithm should be tuned in such a way that the evaluation criteria for these types of algorithms are led to good results.

Therefore, each of the five introduced criteria is considered as a response variable. Tables6 and 7 present the parameter tuning procedure and obtained results for tuning parameters of the proposed NSGA-II and NRGA algorithms relevant to the firs algorithms is to make comparative criteriaof the evaluation of the two algorithms, introduced in this section, resulting ingood solutions; the results ofmetaheuristic algorithm design are normal.

Table 5
Factors and factors levels

Algorithms	Parameter	Range	Low (-1)	Medium (0)	High (+1)
	MaxIt	80-100	80	90	100
	nPop	20-40	20	30	40
	Pc	0.60-0.70	0.60	0.65	0.70
	Pm	0.07-0.13	0.07	0.1	0.13

Table 6
The results of parameter tuning for algorithm NSGA-II

Run	NSGA-II Parameters				NSGA-II Implementing				
Order	maxIt	nPop	Pc	Pm	DVR +	SPC -	NOP+	MID -	CT -
1	80	30	0.65	0.1	1539.06	23.4939	30	10001437.20	66.30
2	80	20	0.7	0.07	2059.05	35.2929	20	10001503.40	49.17
3	100	40	0.7	0.07	1539.06	11.9800	40	10001586.83	101.56
4	90	30	0.7	0.1	2571.02	51.1493	30	10002228.50	75.43
5	80	20	0.6	0.07	2059.04	190.6992	20	10001356.25	43.61
6	80	40	0.6	0.13	1413.07	11.9800	40	10001366.50	83.61
7	90	40	0.65	0.1	1539.06	11.9800	40	10001439.58	94.33
8	90	30	0.65	0.1	1413.06	12.9755	30	10001197.37	77.57
9	90	30	0.65	0.1	1539.06	10.7166	30	10001428.83	72.02
10	80	40	0.7	0.13	1413.07	136.6057	40	10001147.83	91.41
11	100	30	0.65	0.1	2645.04	15.4050	30	10002062.10	80.38
12	100	40	0.6	0.07	1539.06	13.3895	40	10001455.05	98.02
13	90	30	0.6	0.1	2059.05	22.9549	30	10001479.53	67.09
14	90	30	0.65	0.07	1539.06	22.9549	30	10001379.53	72.96
15	90	30	0.65	0.1	1257.02	33.3730	30	10001622.87	77.43
16	100	20	0.7	0.13	1413.07	16.7519	20	10001370.40	62.99
17	90	30	0.65	0.1	1999.05	11.1991	30	10001308.03	75.18
18	90	30	0.65	0.13	1413.07	157.3412	30	10001051.57	76.45
19	100	20	0.6	0.13	1539.06	28.5333	20	10001335.35	59.37
20	90	30	0.65	0.1	1539.06	13.7857	30	10001473.37	76.92
21	90	20	0.65	0.1	2059.05	6.6819	20	10001677.90	55.20

Table 7
The results of parameter tuning for algorithm NRGA

Run	NRGA Parameters				NRGA Implementing				
Order	maxIt	nPop	Pc	Pm	DVR +	SPC -	NOP+	MID -	CT -
1	90	30	0.65	0.1	2897.02	94.1224	30	10002736.23	75.16
2	100	30	0.65	0.1	2897.03	49.7130	30	10002074.10	75.69
3	90	40	0.65	0.1	2645.06	45.1090	40	10001937.73	97.80
4	80	20	0.6	0.07	2059.04	16.7519	20	10001837.45	58.84
5	100	20	0.6	0.13	1969.01	29.0828	20	10001614.50	60.53
6	90	30	0.65	0.07	1539.06	13.7857	30	10001621.17	82.23
7	80	30	0.65	0.1	2645.04	96.9240	30	10002097.10	84.23
8	90	30	0.65	0.13	1539.05	23.3143	30	10001556.47	87.92
9	80	40	0.6	0.13	2059.04	19.1773	40	10001622.58	102.75
10	80	40	0.7	0.13	1539.05	11.9800	40	10001332.80	100.55
11	90	30	0.65	0.1	1999.04	92.0596	30	10001570.77	75.44
12	100	40	0.6	0.07	2571.02	90.3568	40	10002613.43	95.40
13	90	30	0.65	0.1	1465.03	66.4310	30	10001958.53	85.68
14	100	40	0.7	0.07	1539.05	87.4511	40	10001106.18	94.65
15	90	30	0.65	0.1	2645.04	90.6448	30	10001717.47	76.95
16	80	20	0.7	0.07	2645.04	90.0313	20	10002268.80	51.61
17	90	30	0.7	0.1	2645.04	88.8496	30	10001814.40	74.80
18	90	30	0.6	0.1	2059.05	24.4091	30	10001959.20	75.11
19	100	20	0.7	0.13	2645.04	104.8980	20	10002437.45	65.82
20	90	30	0.65	0.1	1413.06	90.4257	30	10001180.67	72.53
21	90	20	0.65	0.1	2645.03	52.0792	20	10002019.20	59.73

Since evaluation criteria are not of a kind, they must be obtained using equation (18) whose doses of factorial

$$\text{RPD} = \frac{|\text{Method sol} - \text{Best Sol}|}{|\text{Best Sol}|} * 100 \qquad (18)$$

Response column in these tables represents the mean of the normalized value of the criteria which are used as the response variable in the RSM method. Anova table related to NRGA the NSGA-II algorithms and regression functions of the algorithms are obtained by the Design Expert software. Finally, the optimal levels of input parameters in algorithms are derived, as shown in table 8.

Table 8
Optimization of values of parameters for NSGA-II and NRGA

Model	Parameters			
	maxIt*	nPOp*	Pc*	Pm*
NSGA-II	84	40	0.69	0.08
NRGA	95	29	0.62	0.073

5. Computational Experiments

Since the proposed mathematical model is quite novel, no suitable problems were found in the existing literature for testing the performance of algorithms. In this study, in total, 30 scheduling problems havebeen selected including 10 small-sized, 10 medium-sized, and 10 large-sized problems. In addition, some other data were necessary to be inserted into the model according to its requirements, which are described as follow.

Activities' duration is a random number between (1, 10).

- The execution modes of activities at most are three modes.
- All reports of this research are based on two or three renewable and non-renewable sources.
- The activities' requirement of renewable and nonrenewable resources is a random number between zero and five.
- Normal time for each activity in any mode follows the uniform distribution of U(1, 10).

In this section, problems are classified into three groups with small, medium, and largesizes. The first group with 5 activities, the second group with 10 activities, and the third group with 15 activities are considered. However, execution modes and number of resources are slightly changed in order to prevent repeating the experiment;all algorithms presented in this study have been coded using software Matlab R2013b in Windows 7, run on a computer with the specifications provided in Table 9.

Table 9
Computer Specification

Processor	Intel® Core™ i5-2430M CPU @ 2.13GHz
RAM	4.00 GB (2.61 usable)
HDD	500 GB

Computed values of comparative criteria of the two algorithms are presented in tables 10 and 11.Performance comparison of each of these two algorithms on the five evaluation criteria are illustrated in Figures 16 to 19.

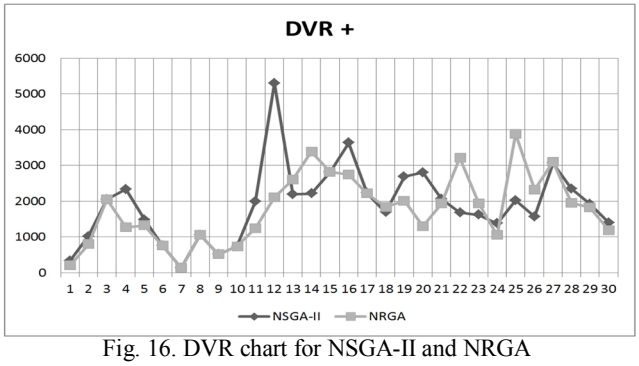

Fig. 16. DVR chart for NSGA-II and NRGA

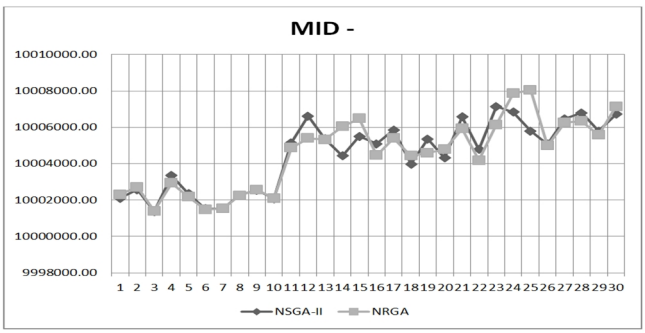

Fig. 18. MID chart for NSGA-II and NRGA

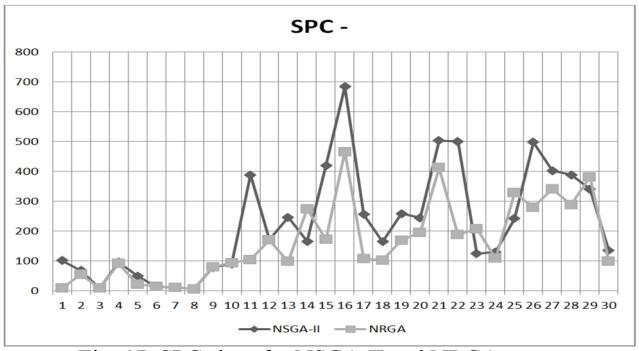

Fig. 17. SPC chart for NSGA-II and NRGA

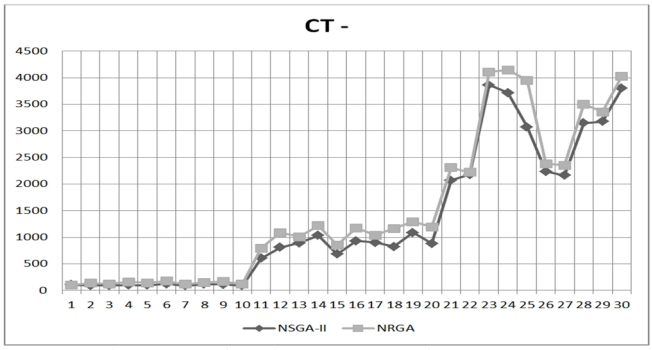

Fig. 19. CT chart for NSGA-II and NRGA

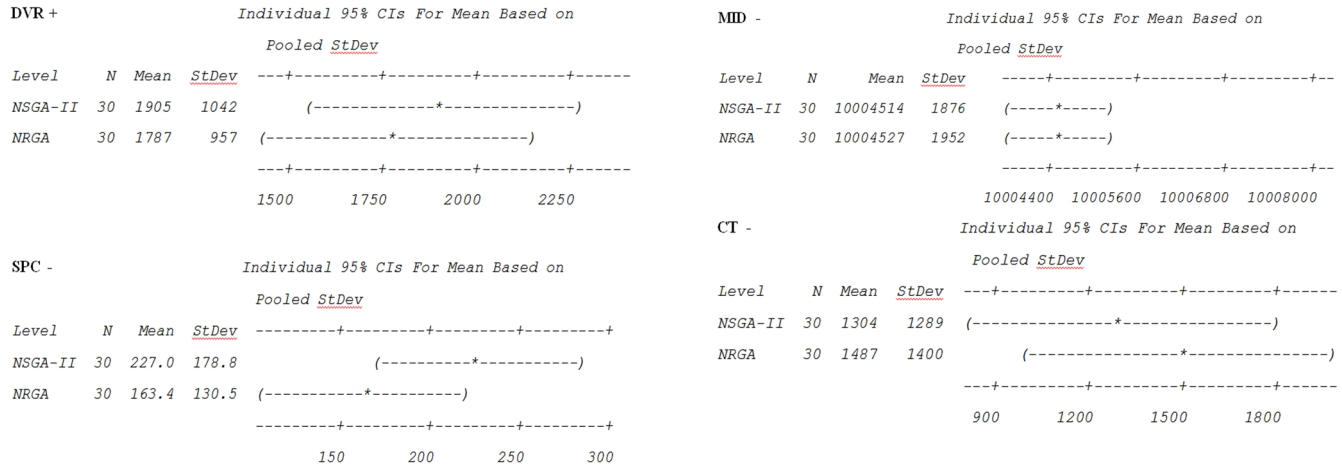

Fig.20. Comparison Chart confidence distance of evaluation criteria ofNSGA-II and NRGA

Table 10
Result of comparison criteria for NSGA-II

Problem Number	Example Size					Evaluation Criteria				
	Size	Activities	nr^{ρ}	Mode	D_i	DVR +	SPC -	NOP +	MID -	CT -
1		5	3	3	1-10	337.05	102.54	40	10002108.10	107.70
2		5	3	3	1-10	1029.08	68.69	40	10002573.25	94.23
3		5	3	3	1-10	2050.04	8.42	40	10001370.23	91.52
4		5	3	3	1-10	2336.25	97.10	40	10003346.43	100.59
5		5	3	3	1-10	1495.10	50.61	40	10002351.53	100.46
6	**Small**	5	3	3	1-10	752.01	14.66	40	10001533.20	124.33
7		5	3	3	1-10	139.43	11.96	40	10001525.50	88.68
8		5	3	3	1-10	1061.07	6.42	40	10002276.50	119.43
9		5	3	3	1-10	514.14	78.69	40	10002547.93	115.17
10		5	3	3	1-10	749.07	91.26	40	10002085.68	89.76
11		10	3	3	1-10	1993.93	389.45	40	10005154.15	606.38
12		10	3	3	1-10	5303.00	171.49	40	10006627.73	809.47
13		10	3	3	1-10	2190.50	246.77	40	10005369.58	895.56
14		10	3	3	1-10	2216.03	165.97	40	10004425.70	1034.80
15		10	3	3	1-10	2816.54	420.89	40	10005515.10	683.10
16	**Medium**	10	3	3	1-10	3630.00	684.37	40	10005086.18	932.10
17		10	3	3	1-10	2230.40	256.33	40	10005860.47	905.56
18		10	3	3	1-10	1691.99	165.70	40	10003973.25	823.96
19		10	3	3	1-10	2689.79	259.06	40	10005364.58	1086.14
20		10	3	3	1-10	2804.31	246.11	40	10004310.95	881.98
21		15	3	3	1-10	2066.48	504.02	40	10006576.60	2072.84
22		15	3	3	1-10	1688.35	501.02	40	10004783.88	2178.87
23		15	3	3	1-10	1619.03	124.80	40	10007149.83	3866.22
24		15	3	3	1-10	1380.07	130.66	40	10006835.80	3717.74
25		15	3	3	1-10	2026.91	242.89	40	10005799.90	3076.63
26	**Large**	15	3	3	1-10	1578.60	498.02	40	10005083.84	2238.46
27		15	3	3	1-10	3066.48	404.02	40	10006473.60	2162.84
28		15	3	3	1-10	2354.76	389.19	40	10006790.25	3146.45
29		15	3	3	1-10	1925.32	342.34	40	10005799.90	3176.63
30		15	3	3	1-10	1401.94	135.76	40	10006710.37	3802.76

For most of the two criteria (DVR) and Number of Pareto (NOP) and higher values for the three categories of spacing (SPC), are the ideal solution (MID) and running time (CT) smaller amounts, are desirable.

In this study, analysis of variance is used to carry out this analysis in a way that the algorithms are analyzed with each criterion using Minitab 16.2 software, and then the results are analyzed. P-Value level less than 5% in the applied analysis of variance indicates a significant difference between the responses of the two algorithms related to that specific criteria; otherwise, it can be said that there is no

significant difference between the performance of the two algorithms in considering that criterion, and algorithms' criteria can be compared accordingly. Results of statistical analysis for each of the five criteria in the graphical illustration of confidence interval criteria comparison using Minitab 16.2 software are presented in Figure 20.

According to the obtained values for the criteria related to each algorithm of NSGA-II and NRGA, the following results can be deduced:

-Considering DVR criteria, NRGA algorithm providesbetter answers compared to NSGA-II algorithm. outperforms NRGA algorithm.

-Considering SPC criteria, NRGA algorithm outperforms the NSGA-II algorithm.

-Considering MID criteria, NRGA algorithm is superior to NSGA-II algorithm.

-Considering CT criteria, NRGA algorithm outperforms NSGA-II algorithm

–Considering the NOP criteria, since the obtained value of thesecriteria is always greater through NSGA-II algorithm, it

Table 11
Result of comparison criteria for NRGA

Problem	Example Size					Evaluation Criteria				
Number	**Size**	**Activities**	nr^{ρ}	**Mode**	D_i	**DVR +**	**SPC -**	**NOP +**	**MID -**	**CT -**
1		5	3	3	1-10	210.06	8.98	29	10002301.48	102.36
2		5	3	3	1-10	797.04	55.89	29	10002722.66	138.48
3		5	3	3	1-10	2050.04	8.43	29	10001390.00	121.27
4		5	3	3	1-10	1281.21	91.79	29	10002937.69	161.34
5	**Small**	5	3	3	1-10	1318.11	22.12	29	10002180.90	143.18
6		5	3	3	1-10	752.01	14.89	29	10001493.52	176.23
7		5	3	3	1-10	139.43	11.95	29	10001527.90	119.86
8		5	3	3	1-10	1061.07	6.42	29	10002258.34	155.31
9		5	3	3	1-10	514.14	80.54	29	10002565.38	165.85
10		5	3	3	1-10	749.07	94.03	29	10002092.93	125.67
11		10	3	3	1-10	1252.69	105.92	29	10004872.14	794.90
12		10	3	3	1-10	2107.64	170.25	29	10005396.00	1085.71
13		10	3	3	1-10	2608.42	98.69	29	10005322.28	1010.90
14		10	3	3	1-10	3387.43	274.74	29	10006047.79	1220.05
15	**Medium**	10	3	3	1-10	2814.46	172.39	29	10006493.03	853.08
16		10	3	3	1-10	2756.01	465.76	29	10004461.28	1176.63
17		10	3	3	1-10	2217.34	108.24	29	10005396.00	1036.95
18		10	3	3	1-10	1845.98	103.19	29	10004434.03	1165.03
19		10	3	3	1-10	2016.02	170.04	29	10004597.03	1294.18
20		10	3	3	1-10	1304.58	196.56	29	10004781.24	1192.94
21		15	3	3	1-10	1933.54	413.02	29	10005936.43	2312.67
22		15	3	3	1-10	3221.10	188.18	29	10004198.24	2220.92
23		15	3	3	1-10	1939.54	208.98	29	10006149.83	4109.75
24		15	3	3	1-10	1054.51	108.92	29	10007861.10	4145.44
25	**Large**	15	3	3	1-10	3883.74	328.50	29	10008053.48	3950.27
26		15	3	3	1-10	2326.18	281.18	29	10004995.34	2383.92
27		15	3	3	1-10	3101.37	341.28	29	10006267.73	2355.67
28		15	3	3	1-10	1954.36	289.19	29	10006340.32	3505.55
29		15	3	3	1-10	1825.15	382.28	29	10005579.54	3355.24
30		15	3	3	1-10	1191.11	98.31	29	10007153.41	4029.36

Parameters Sensitivity Analysis

In this research, the most important and effective parameter is unit cost of renewable resource type which is named as $C^{\rho}{}_{k}$ in the second objective function of model. In order to conduct a sensitivity analysis of this parameter, we have investigated a problem with three modes and 10 activities and have run it by the presented algorithm. As can be seen in Table 12, with increasing $C^{\rho}{}_{k}$ and keeping other parameters constant, the first objective function value is increased. Increasing trend of the first objective model explanation is that by increasing $C^{\rho}{}_{k}$ value, the second objective function raises up. However, considering algorithm performance and logic of non-dominate Pareto solutions, makespan value increases.

Table 12
$C^{\rho}{}_{k}$ sensitivity analysis

$C^{\rho}{}_{k}$	50	75	100	125	150	175	200
F1	450	525	800	500	875	1050	1200
F2	42	48	64	71	71	71	87

Another parameter is the number of activities shown as *n*. As can be seen in Tables 10 and 11, increasing n causes extreme increase of CPU Time in three categories of small, medium, and large problems.

6. Conclusions and Future Research Directions

In this paper, we have attempted to solve the preemptive multi-objective multi-mode project scheduling model for the Resource Investment Problem (P-MMRIP). The first objective function is to minimize the completion time of project (makespan); the second objective function is to minimize and optimize the cost of using renewable resources. Nonrenewable resources are also considered, but as parameters in this model. This problem has not been studied ever before. The problem was described with an integer programming model, and then the non-dominate sorting genetic algorithm (NSGA-II) was proposed to solve it. The preemption of activities is allowed at any integer time units, and for each activity, the best execution mode is selected according to the duration, resource, and two approaches of Serial Schedule Generation Scheme (S-SGS) and Parallel Schedule Generation Scheme (P-SGS). The parameters of the proposed NSGA-II are tuned based on Response Surface Methodology (RSM). The performance of the proposed algorithm on 30 test problems was compared with the NRGA algorithm. From the computation results, we could clearly see that the proposed NSGA-II and NRGA could efficiently solve the project scheduling problem.

Acknowledgement

The authors would like to thank the anonymous reviewers and the editors for their insightful comments and suggestions.

References

Blazewicz, J., Lenstra, J.K. & RinnooyKan, A.H. (1983). Scheduling subject to resource constraints:classification and complexity. *Discrete Applied Mathematic*s, 5(1), pp. 11–24.

Kolisch, R.., Hartmann, S., (2006). Experimental investigation of heuristics for resource-constrained project scheduling: an update. *European Journal of Operational Research*, 174(1), 23–37.

Zhang, H., Li, H., Tam, C.M., (2006). Particle swarm optimization for resource-constrained project scheduling.*International Journal of Project Management*, 24(1), 83–92.

Montoya-Torres, J.R.., Gutierrez-Franco, Pirachican-Mayorga, C.E., (2010).Project scheduling with limited resources using a genetic algorithm.*International Journal of Project Managemen*t, 28(6), 619–628.

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), 1–14.

Agarwal, A., Colak, S., Erenguc, S., (2011). A neurogenetic approach for the resource-constrained project scheduling problem.*Computers & Operations Research*, 38(1), 44–50.

Fang, C.,Wang., (2012). An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem.*Computers & Operations Research*, 39(5).890–901.

Kone, O.,(2012). New approaches for solving the resource-constrained project scheduling problem. 4OR, 10(1), 105–106.

Paraskevopoulos, D.C., Tarantilis, C.D.,Ioannou, G.,(2012). Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm.*Expert Systems with Applications*, 39(4), 3983–3994.

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

Zhang, H., Tam, C.M., Li, H.,(2006). Multimode project scheduling based on particle swarm optimization.*Computer-Aided Civil and Infrastructure Engineering*, 21(2), 93–103.

Lova, A.,Tormos, P., Barber, F. (2006). Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules.*Inteligencia Artificial*, vol. 10, no. 30, pp. 69–86.

Jarboui, B., Damak, N., Siarry, P. & Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. Applied Mathematics and Computation, 195(1), 299–308.

Ranjbar, M., de Reyck, B., Kianfar, F. (2009). A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling.*European Journal of Operational Research*, 193(1), 35–48.

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.

Coelho, J.,Vanhoucke, M., (2011). Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers.*European Journal of Operational Research*, 213(1), 73–82.

Ranjbar, M., (2011).An optimal NPV project scheduling with fixed work content and payment on milestones.*International Journal of Industrial Engineering & Production Research*, 22(3), 181–186.

Barrios, A.,Ballestin, F.,Valls, V., (2011). Adouble genetic algorithm for the MRCPSP/max.*Computers & Operations Research*, 38(1), 33–43.

Afshar-Nadjafi, B., Rahimi, A., Karimi, H., (2013). A genetic algorithm for mode identity and the resource-constrained project scheduling problem.*ScientiaIranica*, 20 (3), 824–831.

Afruzi, E.N.,Roghanian, E.,Najafi, A.A.,Mazinani, M., (2013). Amulti-mode resource-constrained discrete time-cost trade off problem solving using an adjusted

fuzzy dominance genetic algorithm.ScientiaIranica, 20(3), 931–944.

Mohring, R.H., (1984). Minimizing costs of resource requirements in project networks subject to a fixed completion time.*Operations Research*, 32(1), 89–120.

Rangaswamy, B., (1998). Multiple resource planning and allocation in resource- constrained project networks [Ph.D. thesis].Graduate School of Business, University of Colorado.

Demeulemeester, E., (1995). Minimizing resource availability costs in time-limited project networks. *Management Science*, 41(10), 1590–1598.

Drexl, A.,Kimms, A. (2001). Optimization guided lower and upper bounds for the resource investment problem.*Journal of the Operational Research Society*, 52(3), 340–351.

Rodrigues, S.B., Yamashita, D.S., (2010). An exact algorithm for minimizing resource availability costs in project scheduling.*European Journal of Operational Research*, 206 (3), 562–568.

Yamashita, D.S., AmaralArmentano, V., Laguna, M., (2006).Scatter search for project scheduling with resource availability cost. *European Journal of Operational Research*, 169(2), 623–637.

Shadrokh, S.,Kianfar, F., (2007). A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty.*European Journal of Operational Research,* 181(1), 86–101.

Ranjbar, M.,Kianfar, F., Shadrokh, S., (2008). Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. *Applied Mathematics and Computation*, 196(2), 879–888.

Van Peteghem, V.,Vanhoucke, M., (2013). An artificial immune system algorithm for the resource availability cost problem.*Flexible Services and Manufacturing Journal*, 25,(1-2), 122–144.

Kaplan, L., (1988). Resource constrained project scheduling with preemption of jobs [Ph.D. thesis], University of Michigan, Ann Arbor, Mich, USA.

Demeulemeester, E.L., Herroelen, W.S., (1996). An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem.*European Journal of Operational Research*, 90(2), 334–348.

Ballestin, F.,Valls, V., Quintanilla, S., (2008). Pre-emption in resource-constrained project scheduling.*European Journal of Operational Research*, 189(3), 1136–1152.

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(1), 140–154.

Damay, J.,Quilliot, A.,Sanlaville, E., (2007). Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Researc*h, 182(3), 1012–1022.

Buddhakulsomsiri, J., Kim, D.S., (2006). Properties of multimode resource-constrained project scheduling problems with resource vacations and activity splitting.*European Journal of Operational Research*, 175(1), 279–295.

Van Peteghem, V.,Vanhoucke, M., (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem.*European Journal of Operational Research*, 201(2), 409–418.

Afshar-Nadjafi, B.,Arani, M.,(2014). Multimode preemptive resource investment problem subject to due dates for activities: formulation and solution procedure," Hindawi Publishing Corporation.*Advances in Operations Research*, 2014, Article ID 740670.

Tavana, M., Abtahi, A.R.,Khalili-Damghani, K., (2014). new multi-objective multi-mode model for solving preemptive time–cost–quality trade-off project scheduling problems.*Expert Systems with Applications*, 41, 1830–1846.

Bulbul, K.,Kaminsky, P., Yano, C., (2007).Preemption in single machine earliness/tardiness scheduling. Journal of Scheduling, 10(4-5), 271–292.

Baptiste, P.,Carlier, J.,Kononov, A.,Queyranne, M.,Sevastyanov, S.,Sviridenko, M., (2009). Structural properties of optimal preemptive schedules.DiskretnyiAnaliziIssledovanieOperatsii, 16(1), 3–36.

Baptiste, P.,Carlier, J.,Kononov, A.,Queyranne, M.,Sevastyanov, S.,Sviridenko, M., (2011). Properties of optimal schedules in preemptive shop scheduling. Discrete Applied Mathematics, 159(5), 272–280.

Sprecher, A., Hartmann, S.,Drexl, A., (1997). An exact algorithm for project scheduling with multiple modes.OR Spektrum. Quantitative Approaches in Management, 19(3), 195–203.

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., (2000). A fast elitist non-dominated sortinggenetic algorithm for multi-objective optimization: NSGA-II, Lecture notes in computer science ,849-858 .

Zitzler, E., (1999). Multi objective evolutionary algorithms: a comparative case study and thestrength Pareto approach. *Evolutionary Computation, IEEE Transactions*, 3, 227-257.

Schott, J.R., (1995). Design Using Single and Multi-Criteria Genetic Algorithms.Master'sThesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology ,Boston, M.

This article can be cited: Salimi M. & Najafi, A.A. (2018). Modeling and Solution Procedure for a Preemptive Multi-Objective Multi-Mode Project Scheduling Model in Resource Investment Problems. *Journal of Optimization in Industrial Engineering.* 11 (1), 169- 183

URL: http://www.qjie.ir/article_535423.html
DOI: 10.22094/JOIE.2017.592.1381