Scientia Iranica E (2016) 23(4), 1893–1903

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

A hybrid genetic algorithm to maximize net present value of project cash flows in resource-constrained project scheduling problem with fuzzy parameters

F. Fathallahi and A.A. Najafi*

Faculty of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran.

Received 24 August 2014; received in revised form 18 May 2015; accepted 17 August 2015

KEYWORDS
Discounted cash flows;
Net present value;
Fuzzy sets;
Uncertainty;
Genetic algorithms.

Abstract. This paper studies a specific resource-constrained project scheduling problem under uncertainty. To do so, the problem is investigated in fuzzy environment and the goal is to maximize the net present value of the project cash flows. The problem is first mathematically formulated. Then, a hybrid Genetic Algorithm is proposed and tuned to solve this NP-hard problem. The performance of the proposed algorithm is evaluated with comparing two well-known metaheuristic algorithms through a set of instances. Finally, comprehensive computational results are illustrated and the results are analyzed and discussed.

1. Introduction

Scheduling plays a major role in project management in which scheduling process decides when the project activities will start and how they will use the available resources. The Resource-Constrained Project Scheduling Problem (RCPSP) involves assigning the resources to each activity considering the limitation of resources and precedence relationship in order to meet the specific goals. An extensive discussion on the RCPSP problems can be found in [1].

Many researchers have investigated different variations of the RCPSP in the past decades. This research can be classified based on the goals. There are two major goals which are related to time and financial aspects of the project. For this purpose, minimization of project makespan and maximization of Net Present Value (NPV) are considered.

Minimizing the project makespan is one of the popular criteria in RCPSP. In this regard, the goal is to minimize the completion time of the project. There are some available solution approaches for solving RCPSP to minimize the project makespan, including binary [2], branch-and-bound procedure [3,4], heuristic methods [5], genetic algorithm [6], simulated annealing approach [7], ant colony approach [8], particle swarm optimization [9], and integer programming [10].

While many authors have focused on minimizing the makespan, most of the research concentrates on maximizing the project NPV [11]. In NPV maximization of the projects, each activity has related positive cash flow and negative cash flow, which demonstrate the revenue of the activity and the cost of its execution, respectively. Some available solving methods for RCPSP with discounted cash flows can be mentioned as branch-and-bound [12], heuristic method [13], Tabu search [14], Lagrangian relaxation method [15,16], depth-first branch-and-bound algorithm [17], ant colony algorithm [18], and simulated annealing with Tabu search [19].

In the above research works, the parameters of the problem are considered as deterministic values, but

*. *Corresponding author. Tel.: +98 21 84063378; Fax: +98 21 88674858*
E-mail address: aanajafi@kntu.ac.ir (A.A. Najafi)

there are some issues that happen during implementation of the projects. Because of the unique characteristic of the project, predicting the parameters related to each activity is confronted with many uncertainties and lack of information. There are two major approaches to tackle these uncertainties: stochastic approach and fuzzy approach. There is much research on stochastic approach; if interested, one can refer to [20-24].

However, due to vague information and insufficient historical data related to the activities in the real world, project managers and experts are confronted with vague conditions to make decisions. The probability distributions are not usually available, implying that the scientists prefer to use fuzzy set theory instead. The fuzzy set theory is especially well suited to handle such vague information. In this area, Lootsma [25] concluded that the stochastic network planning was intractable and hence suggested fuzzy modelling instead. In his study, fuzzy modeling has been considered closer to reality, but theoretically not established. Dubois and Prade [26] provided an overview of fuzzy numbers. Janczura and Kuchta [27] investigated proactive and reactive scheduling under fuzzy task duration. Wang [28] presented a fuzzy beam search approach for solving the fuzzy RCPSP with the objective of minimizing the schedule risk. Hapke and Slownski [29] proposed uncertainty of parameters, modeled by means of L-R fuzzy numbers in RCPSP. Hapke et al. [30] presented an FPS (Fuzzy Project Scheduling) decision support system applied to software project scheduling. Wang et al. [31] considered fuzzy RCPSP and solved it with an efficient genetic algorithm and compared the different solutions gained by different ranking methods of fuzzy numbers. Atli and Kahraman [32] proposed a mathematical model to deal with project scheduling problem under vagueness and provided a framework of a heuristic approach for Fuzzy Resource-Constrained Project Scheduling Problem (F-RCPSP) using heuristic and metaheuristic scheduling methods. Kaur and Kumar [33] provided a fuzzy arithmetic in fully fuzzy linear programming for getting an optimal solution. Masmoudi and Hait [34] proposed a greedy algorithm and a genetic algorithm to minimize project makespan in the fuzzy RCPCP model. Vartouni and Khanli [35] considered Multi-Resource-Constrained Project Scheduling Problem (MRCPSP) with fuzzy activity duration times due to the nonrenewable resources and the multiple modes. Xu and Feng [36] investigated MRCPSP under fuzzy randomness.

According to the above literature and to the best of our knowledge, no research on the fuzzy RCPCP has been undertaken to maximize the project NPV. On the other, because of the fluctuated situation and the existence of inflation, the interest rate of the project, outcomes, and incomes of each activity cannot be estimated exactly. To cope with this uncertainty related to each parameter, in this paper, all of the parameters are considered as fuzzy numbers. Hence, in this paper, for the first time, a fuzzy RCPSP is discussed in which the goal is to maximize the net present value of the project cash flows. We call this problem a Fuzzy Resource-Constrained Project Scheduling Problem with Discounted Cash Flows (F-RCPSP-DCF). We formulate the problem and propose an efficient hybrid genetic algorithm to solve it.

The rest of the paper is organized as follows. In the next section, we formulate the problem mathematically. A hybrid genetic algorithm is then presented in Section 3 to solve the model. Next, in Section 4, the parameters of the proposed algorithm are tuned and their performances are investigated. Finally, Section 5 is assigned to conclusion remarks.

2. Problem formulation

Suppose a project has n activities indexed from 1 to n, in which activities 1 and n are dummies that represent the start and completion of the project, respectively. The activities must be scheduled by considering the precedence relations between activities and resource constraints. Precedence relations of activities are shown by an activity on node network with no loops. The fuzzy cash flow associated with activity i occurs at the end of activity i. The parameters for F-RCPSP-DCF can be defined as follows:

$\tilde{C}_i$	Fuzzy cash flow for activity i;
$\tilde{\alpha}$	Fuzzy interest rate;
$\tilde{d}_l$	Fuzzy duration for activity i;
$\tilde{F}_l$	Fuzzy finish time for activity i;
M_r	Availability level of resource r;
$\tilde{l}_{ir}$	Fuzzy resource requirement of activity i for resource r;
$p(i)$	The set of activities that should be performed before activity i.

Considering discounted cash flows and resource constraints, the problem can be formulated as follows:

$$\max \tilde{Z} = \sum_{i=1}^{n} \tilde{C}_l e^{-\tilde{\alpha}\tilde{F}_l}. \tag{1}$$

S.T.

$$\tilde{F}_J \leq \tilde{F}_l - \tilde{d}_l \qquad \forall j, \forall p(i), \forall, \tag{2}$$

$$\sum_{\forall i} \tilde{L_{lr}} \leq m_r \qquad \forall r, \forall t, \tag{3}$$

$$\tilde{F}_l \geq 0, \qquad \forall i. \tag{4}$$

Objective function (Eq. (1)) is to maximize the net present value of the project cash flows. Constraint (2) enforces the precedence relations between activities. Constraint (3) ensures that the amount of resource consumption is less than the available resources. Finally, Constraint (4) denotes the domain of the variables.

Since Blazewicz et al. [37] showed that RCPSP is an NP-hard problem, the F-RCPSP-DCF is also NP-hard and therefore a hybrid metaheuristic algorithm is proposed in the next section to solve it.

3. A hybrid genetic algorithm

Genetic Algorithms (GAs) are search algorithms based on the mechanism of natural selection and the natural genetics. GA starts with a population of the initial feasible solutions; the given constraints are satisfied and the new solutions are generated by employing the mutation and the crossover operators. This process is continued until the pre-specified number of iterations is reached. In the project scheduling literature, many researchers have applied the GA (for example, refer to [38-40]). In this section, we propose a hybrid GA to solve the problem.

3.1. Basic scheme of hybrid algorithm

In this part, a brief explanation of the algorithm performances is provided. At first, initial population is considered. In this hybrid genetic algorithm, there are three operators: crossover operator, mutation operator, and SA operator as the last one. The first two of them have the probabilities of P_{cr} and P_{mu}, respectively, and the SA operator is implemented with a specific probability (SA rate). In crossover operator, the parents are selected based on the roulette wheel selection. A partially matched crossover (PMX) is used as the crossover operator, which is so beneficial in this kind of problem and guarantees the sequence without repeatable activities. A new mutation operator regarded to this problem is introduced, which is different from the classical mutation. In this mutation, the random sequence of some activities within the selected sequence is chosen with random length and the best permutation of these activities for enhancing the solutions is considered. The drawback of genetic algorithm is very well tackled by simulated annealing, which has the capability to search local region in the solution space exhaustively. SA algorithm is applied to some chromosomes which are selected based on SA probability. By using SA probability, the numbers of chromosomes are obtained from the initial population and SA algorithm is applied to these numbers of chromosomes. After implementing the SA algorithm on this population, the better solutions, which are improved by SA algorithm, are merged with the other solutions gotten from other operators in solution pool. Finally, all of the solutions, which are obtained by crossover operator, mutation operator, and SA algorithm, are sorted based on their values of fitness function in descendent order and the solutions are selected according to the population size. When the number of iterations reaches a specified number, the algorithm would be terminated.

3.2. Solution encoding and decoding

In this study, each chromosome i is a vector consisting n genes, where n denotes the number of the project activities. The value of each gene in a chromosome denotes an activity number and the value of gene states the position of that activity at generating the project schedule. Note that, each activity can appear in the vector at any position after all its predecessors. To generate the schedule of an individual, we select the activities one by one from the vector according to their priorities and schedules it as soon as possible in the schedule, considering precedence and resource feasibility. Because all of the variables and parameters in this research are considered as fuzzy numbers, for comparison between the fuzzy numbers, a fuzzy comparison method should be applied. In the next part, a brief explanation of this procedure for comparing the fuzzy numbers is given. For decoding the procedure, let CA be the completed activities, PA be activities in progress, and $\tilde{t}$ be the present time. As mentioned, activities in lower position have priority for scheduling; thus, activities are chosen based on their position and if there are enough resources, the activities will be scheduled. Available resource can be computed by summing up the resource consumption of activities in progress and subtracting this resource consumption from the available resource. This process is continued until all of the activities are scheduled; then, by computing finish time of the activity in the sequence, the fitness function of solution can be obtained. Decoding of the algorithm can be summarized as follows:

Set $\tilde{t} = (0, 0, 0)$, CA $= \{\emptyset\}$, PA $= \{\emptyset\}$.

Repeat
  Until all of the activities are added to CA
  Repeat
    Select the activity in the least position in the solution
  If
    The resource consumption is lower than the available resource,
    Set $\tilde{St}_l = \max\{\tilde{t}, \tilde{Ft}_{p(i)}\}$, $\tilde{Ft}_l = \tilde{t} + \tilde{d}_l$

      Update resource availability
    Add the activity to PA
  Else
    Increase present time up to the earliest finish time of the progressed activities

```
    Add this activity to CA
    Update resource availability
  End
  End
End
```

Fuzzy comparison procedure: Most of the research related to resource-constrained project scheduling problem consider crisp parameters. Because of the reasons mentioned above, the duration of activity and other parameters in this paper are considered as fuzzy numbers. The theory of fuzzy set was first introduced by Zadeh [41]. In a fuzzy environment, ranking fuzzy numbers is a prerequisite and substantial procedure. The method of ranking fuzzy numbers has been categorized by Yang et al. [42] into four classes of (1) preference relation, (2) fuzzy mean and spread, (3) fuzzy scoring, and (4) linguistic express. Wang and Kerre [43] have mentioned that in fuzzy literature, there exist 35 indices for the comparison of fuzzy quantities. They categorize them into three classes; the first one is ranking function, the second one reference sets, and the last option linguistic approach. In this paper, ranking of fuzzy numbers is considered for comparison of two fuzzy numbers.

To rank fuzzy numbers, we apply a new approach proposed by Thorani et al. [44], which uses orthocenter of centroid of fuzzy numbers for its distance from original point. This method can rank all types of the fuzzy numbers, including fuzzy numbers with different membership functions, to deal with fuzzy risk analysis problem. This method is more flexible and practical than the numerous methods of ranking. The centroid of a trapezoid is measured as the balancing point of the trapezoid and the orthocentre of these centroid points is a much more balancing point for a generalized trapezoidal fuzzy number. Consider $\tilde{A} = (a, b, d, w)$ as fuzzy number. In normal cases, the value of w is 1. The orthocenter of centroid, which has the most balancing point of fuzzy numbers, can be calculated as follows:

$$O_{\tilde{A}}(\tilde{x}_0, \tilde{y}_0) = \left(b, \frac{(b-a)(d-b)+w^2}{3w}\right). \tag{5}$$

The ranking function of fuzzy numbers maps them into the values for comparison. $R(\tilde{A}) = \sqrt{\tilde{x}_0^2 + \tilde{y}_0^2}$ is the Euclidean distance from the orthocenter of the centroid as defined from the original point. The ranking of fuzzy numbers is as follows:

$$\text{If} \quad R(\tilde{A}) > R(\tilde{B}) \quad \text{then} \quad \tilde{A} > \tilde{B}, \tag{6}$$

$$\text{If} \quad R(\tilde{A}) < R(\tilde{B}) \quad \text{then} \quad \tilde{A} < \tilde{B}. \tag{7}$$

When two numbers are equal, each number of A or B will be chosen randomly. If the ranking of each number is greater than that of the other, the fuzzy number with bigger ranking will be greater than the second fuzzy number. In each current fuzzy time which is greater than the previous fuzzy time, the prior activity, based on the decoded chromosome mentioned in the previous section, would be selected. The start time of this activity can be obtained by comparison of fuzzy numbers that are the finish time of the activity's precedents and the current time, which has no resource limitation for performing the activity. The previous operations are based on resource constraints and the total resources consumption, which is used by the activities in progress, and the candidate activity for scheduling are compared with the resource availability. If there is not enough resource available for performing the selected activity, current time will be increased. Thus, in another case, ranking of fuzzy numbers is used to select the minimum finish time of activities in progress. Finally, the solution with higher NPV would be selected for the next generation according to their fitness function.

3.3. Calculating objective function

The objective function is manipulated and applied to some fuzzy arithmetic, which is inspired from [45]. As a result, the objective function is modified as follows.

Consider (C_i^p, C_i^m, C_i^o), $(\alpha^p, \alpha^m, \alpha^o)$, and (F_i^p, F_i^m, F_i^o) as fuzzy cash flows, fuzzy interest rates, and fuzzy finish times, respectively. By using fuzzy arithmetic and putting the fuzzy cash flow in Eq. (1), the following equation can be written:

$$\tilde{Z} = \sum_{i=1}^{n} \tilde{C}_i e^{-\tilde{\alpha}\tilde{F}_i} = \sum_{i=1}^{n} (C_i^p, C_i^m, C_i^o) \times e^{-\tilde{\alpha}\tilde{F}_i}. \tag{8}$$

Also, using the fuzzy arithmetic mentioned by Kaur and Kumar (2012), Eq. (6) can be produced as follows:

$$\begin{aligned}
&(\alpha^p, \alpha^m, \alpha^o) * (F_i^p, F_i^m, F_i^o) \\
&= (\min(\alpha^p F_i^p, \alpha^p F_i^o), \alpha^m F_i^m, \max(\alpha^o F_i^p, \alpha^o F_i^o)) \\
&= (\alpha^p F_i^p, \alpha^m F_i^m, \alpha^o F_i^o) \to -(\alpha^p F_i^p, \alpha^m F_i^m, \alpha^o F_i^o) \\
&= (e^{-\alpha^o F_i^o}, e^{-\alpha^m F_i^m}, e^{-\alpha^p F_i^p}).
\end{aligned} \tag{9}$$

As a result, using the above equation and putting it into objective function, the modified objective function can be obtained as Eq. (7):

$$\begin{aligned}
\tilde{Z} &= \sum_{i=1}^{n} (C_i^p, C_i^m, C_i^o) \times \left(e^{-\alpha^o F_i^o}, e^{-\alpha^m F_i^m}, e^{-\alpha^p F_i^p}\right) \\
&= \sum_{i=1}^{n} \Big(\min(C_i^p e^{-\alpha^o F_i^o}, C_i^p e^{-\alpha^p F_i^p}), C_i^m e^{-\alpha^m F_i^m}, \\
&\qquad \max(C_i^o e^{-\alpha^o F_i^o}, C_i^o e^{-\alpha^p F_i^p})\Big).
\end{aligned} \tag{10}$$

Since maximization of ranking fuzzy objective function leads to the same result in comparison with maximization of the fuzzy objective function, the fuzzy objective function can be substituted with its ranking in Eq. (8) as follows [32]:

$$\begin{aligned}\max R\left(\tilde{Z}\right) &= \frac{1}{4}\left(\sum_{i=1}^{n}\left(\frac{C_i^p e^{-\alpha^o F_i^o}+C_i^p e^{-\alpha^p F_i^p}}{2}\right)\right.\\ &-\left|\frac{C_i^p e^{-\alpha^o F_i^o}-C_i^p e^{-\alpha^p F_i^p}}{2}\right|+2C_i^m e^{-\alpha^m F_i^m}\\ &+\left(\frac{C_i^o e^{-\alpha^o F_i^o}+C_i^o e^{-\alpha^p F_i^p}}{2}\right)\\ &+\left.\left|\frac{C_i^o e^{-\alpha^o F_i^o}-C_i^o e^{-\alpha^p F_i^p}}{2}\right|\right)\\ &=\frac{1}{4}\left(\sum_{i=1}^{n}\left(\frac{C_i^p-|C_i^p|+C_i^o+|C_i^o|}{2}\right)e^{-\alpha^o F_i^o}\right.\\ &+2C_i^m e^{-\alpha^m F_i^m}\\ &+\left.\left(\frac{C_i^p+|C_i^p|+C_i^o-|C_i^o|}{2}\right)e^{-\alpha^p F_i^p}.\right. \qquad (11)\end{aligned}$$

3.4. Initial population

The selection of the initial population will be much important since it affects the search area of algorithm for several iterations. Numbers of chromosomes are produced based on the population size. In this part, a procedure is explained for producing feasible initial population. First of all, the list of Eligible Activity Set (EAS) should be composed, which contains the activities with scheduled precedencies. EAS contains the set of activities, among which the precedent activities are done and do not have any limitation for execution based on their precedencies. At each iteration, a set called Eligible Activity Set (EAS) is constructed. This set consists of non-selected activities of which the predecessors in sequences have been determined. After determination of the EAS, the activities are selected from the EAS and their places in the sequence are determined one by one. Then, the EAS is updated and the iteration is repeated. The algorithm stops if the sequence of all activities is determined. All steps are summarized as follows [46]:

- **Step 1.** Define the precedence of each activity;
- **Step 2.** Compose the list of EAS (if all the precedents of each activity are performed, the particular activity will join the set of activities that can be put in the sequence; these set of activities are called EAS);
- **Step 3.** Select one of the activities in the EAS randomly and locate it in the chromosome;
- **Step 4.** Update the list of EAS;
- **Step 5.** Return to Step 3 until all of the activities are allocated in the activity list.

3.5. Crossover

In GA algorithm, new offspring are produced by applying crossover operator. In this area, several methods have been developed for crossover operator. In this paper, partially matched crossover (PMX) is used to produce the offspring [47,48]. In partially matched crossover operator, two crossover points are selected randomly from the parents. The two crossover points give a matching selection, which is used to affect a cross through position-by-position exchange operations. Assume two chromosomes $Pt_1 = (j_1^1, j_2^1, \cdots, j_n^1)$ and $Pt_2 = (j_1^2, j_2^2, \cdots, j_n^2)$ as parents. For applying this type of crossover, two points (r, k) in each pair of parents are considered and the content between them is substituted.

If there are some similar activities in each offspring, they will be changed with the related element which is substituted. For example, consider the following example:

Parent 1: $(1, 2, 3, 4|5, 6, 7|8)$,

Parent 2: $(8, 5, 2, 1|3, 6, 4|7)$.

If the activities 5, 6, and 7 change their positions with 3, 6, and 4 in parent 2, the produced offspring will have the same activities in its sequence. Because activity 3 is in relation with activity 5 in parent 1, and activity 4 with 7, their positions are changed. For producing a feasible offspring, all of the precedents of each activity are considered and their position is checked. If their position is after the activity, it will be placed in a position randomly before the activity which is called repair process.

3.6. Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation to the next. For each chromosome with n activities such as $Pt_1 = (j_1^1, j_2^1, \cdots, j_n^1)$ with a probability of P_m, the mutation operator is applied in this procedure.

- **Step 1.** Consider an integer number m in the interval of $[1, n]$;
- **Step 2.** m activities in the chromosome are randomly considered; for example, the rth activity up to the $(r+m)$th activity in the solution are selected, e.g. $Pt_1 = (j_1^1, \cdots, j_r^1, \cdots, j_{r+m}^1, \cdots, j_n^1)$, which would be a small sequence;
- **Step 3.** All of the permutations in this sequence are considered. Among all of the permutations, one

feasible permutation would be selected with more objective function value.

This solution is produced by mutation operator. The highest value of fitness function and the feasibility of solution will be guaranteed.

For example, in one chromosome, consider $m = 3$ in the sequence; all the available permutations can be 345, 354, 453, 435, 543, and 534.

Among all of those permutations, select the feasible solution by considering the precedence constraint and the solution with higher value of fitness function.

3.7. Hybridizing genetic algorithm with simulated annealing

Simulated annealing was introduced by Kirkpatrick [49] and is a probabilistic metaheuristic for the local optimization problem of locating a good approximation to the global optimum of a given function in a large search space. It is based on heating and cooling the materials. It starts in high temperature and by each step, the temperature goes down until reaching the cooling temperature. In each iteration, the temperature is cooled with a specific rate, which is denoted as cooling rate. For minimization problem, the neighbourhood solution with lower fitness function replaces the pervious solution and for maximization problems, it is correct to replace solution with larger objective function. In each step, if the neighbourhood solution has a better objective function, it is accepted; however, if it has a worse value, it will be accepted by the value of specific probability $e^{-\frac{\Delta}{t}}$ or will be rejected, in which Δ is the difference between the objective functions of the previous and this iterations and t is the temperature of the ongoing process. A random number is defied and if the previous probability is greater than the random number, which is between $[0, 1]$, the solution will be accepted. SA algorithm accepts the solution with worse value to escape from local optima and accept the solution with better value to direct the algorithm to better objective function. This process is repeated until reaching the cooling temperature.

Many algorithms are hybridized for reaching a better solution which is globally optimized. Among these algorithms, many introduce GA with SA for hybridizing, e.g. [50,51].

In this paper, the rate of applying the simulated algorithm is considered as SA rate. In the beginning of the algorithm, the high temperature should be defined too and in each time, SA algorithm is applied to the population; the high temperature is cooled by the related cooling rate and this loop is continued until the defined number of iterations is reached. In each iteration, the new solution (neighbourhood solution) is obtained by swap of the two elements in the sequence, because the new solution is not feasible in some cases. Therefore, we first check the feasibility of new solution, and in case of infeasibility, a repairing approach is applied. If the solution is infeasible by considering the precedence constraints, the next neighbourhood solution will be generated and SA algorithm will be continued. These explanations are considered as better understanding of SA algorithm in the hybrid algorithm, but the location of SA algorithm should be clarified in the whole structure of hybrid algorithm. SA algorithm gets its population from the whole population of the hybrid algorithm based on its specific rate. It is implemented on these chromosomes and new solutions with better fitness function are merged with the solutions which are taken from the crossover and mutation operators. Finally, the best solutions based on the population size are chosen and they will be transferred to the new generation.

4. Computational results

In this section, the parameters of hybrid algorithm, which are crossover rate, population size, mutation rate, SA rate, the initial temperature, and cooling rate, are tuned. Since the problem of RCPSP has proved to be an NP-hard problem [37], two other metaheuristics are considered for comparison. Because this kind of RCPSP did not exist hitherto, there are not any standard test problems. Some instances, which are produced by Progen software [52], are used for comparison of the performance of the algorithm. The instances are used with different numbers of resources and different interest rates. Because this software produces deterministic problems, for adapting the instances for the fuzzy model, we consider the values gotten from Progen as the most likely elements; the optimistic and pessimistic values of the fuzzy numbers are obtained by decreasing and increasing random values from the deterministic values of Progen software. In this paper, in addition to the parameters which are produced by Progen software, positive cash flows and negative cash flows are considered and they are generated randomly in the range of [0,1500]. The numbers of resources are assumed 2, 3, and 4 and different fuzzy interest rates, which are 0.02 and 0.03 as the most likely elements of fuzzy interest rate, are considered.

4.1. Tuning the parameters

In order to test the effect of each parameter on the results of HGA, we need some methods to illuminate their strength. Different levels of each parameter can cause different results; thus, the best level of parameters should be chosen. In this way, the algorithm can show its best performances after tuning of the algorithm parameters. Taguchi method was used in 1940 for quality control and quality improvement. Before using Taguchi procedure, all of the possible examinations were done

Table 1. HGA parameters and their levels.

Parameters	Level (1)	Level (2)	Level (3)
Initial temperature	500	1000	1500
Cooling rate	0.94	0.96	0.98
Crossover rate	0.7	0.8	0.9
Mutation rate	0.3	0.4	0.5
Population size	50	100	150
SA rate	0.04	0.05	0.06

through the factorial design. Taguchi method decreases the number of experiments and should be performed for determining the importance of each factor. In this section, we use Taguchi approach to reduce the number of required tests. To do this, Taguchi developed a family of full-factorial experiments, called orthogonal array, in the early 1960s. He is regarded as the foremost proponent of robust parameter design, which is an engineering method to product or process a design that focuses on minimizing variation or sensitivity to noise [53].

After analysis and evaluation of the parameters, three levels for each of them are chosen. In HGA algorithm, the parameters are crossover rate, population size, mutation rate, SA rate, the initial temperature, and cooling rate, as shown in Table 1.

By using Taguchi method as the design of algorithms, 26 experiments are performed for getting the best value of each parameter as it is shown in Table 2.

By using Taguchi procedure, we get the best level for each parameter instead of implementing so many experiments in full-factorial design. The best value of each parameter in HGA algorithm is shown in Table 3.

4.2. Comparison with other algorithms

In this section, for measuring the efficiency of the proposed algorithm, we need other algorithms to compare their results. As mentioned previously, resource-constrained project scheduling problems are NP-hard problems and because this kind of RCPSP is almost new, there is not any solution approach to evaluate our proposed algorithm. Hence, we applied two well-known metaheuristic algorithms to the problem, i.e. a classical Particle Swarm Optimization (PSO) and a standard GA. We selected these algorithms due to their efficiency in solving project scheduling problems; e.g. [54-58]. The algorithms are coded with Matlab 2012 and applied to 180 instances, containing 10, 20, and 30 activities for each instance. There are different numbers of resources to which these instances are applied with 2, 3, and 4 resources. The interest rates are considered 0.02 and 0.03, i.e. the most likely elements of interest rate. The value of fitness function, which is NPV here, is calculated by using Eq. (9).

The indices, which are used for measuring performances of the algorithms, are Average Relative Deviation (ARD), Maximum Relative Deviation (MRD), and the number of experiments for HGA ($\#n$) that have the best NPV in each category in comparison with other algorithms. The best solution is taken from the comparison of outputs of three algorithms. The indices are summarized as follows:

- $\#n$: The number of instances for which the algorithm found a solution better than that of the other algorithm;
- MAD: The maximal absolute deviation from the best solution known;
- ARD: The average relative deviation from the best solution known;
- CPU: The average computational time of the algorithm (minute).

Table 4 shows that in 159 experiments, HGA algorithm has better performance than the PSO algorithm and GA; it has 0.35% average deviation from the best solution, while GA and PSO have 5.85% and 8.25%, respectively. On the other hand, MRD for HGA is 5.03%, which is lower than MRDs for PSO and GA that are 23.12% and 26.67%, respectively. The differences of HGA with PSO and HGA with GA algorithms are computed and the hypothesis test for these differences is applied. Since the two groups do not follow the normality distribution, we used the Mann-Whitney test for these two independent non-normal groups. Based on this test, the mean of the solutions of the HGA is better than the mean of the solutions of the PSO and the GA. Thus, it can be concluded that HGA contains better results. The interest rate does not have much influence on the performances of the algorithm. By increasing the number of activities and number of resources, the complexity of the instances will increase. All in all, the proposed algorithm can be employed for this kind of problem efficiently. However, the average time of HGA is 4.74, which is greater than the average times of PSO and GA that are 1.07 and 2.12, respectively. Thus, when considering time factor, other algorithms are more time-efficient.

The average relative deviation for each algorithm is explained in Table 4. Both HGA and PSO algorithms are measured based on their RDs and the average of RDs for each algorithm; there are many differences between these two algorithms. RD of the best solution for HGA is much lower than the RDs of the best solutions for PSO algorithm and standard GA. The average RD for HGA is much lower than those for PSO and standard GA. It is 0.08 for PSO, 0.001 for HGS, and 0.03 for GA in relation to instances with 10 activities; 0.07 for PSO, 0.004 for HGA, and 0.05 for standard GA in relation to instances with 20 activities;

Table 2. Design of experiments for tuning of HGA parameters.

Experiments	Initial temperature	Cooling rate	Crossover rate	Mutation rate	Population size	SA rate
1	1	1	1	1	2	2
2	1	1	1	1	3	3
3	1	2	2	2	1	1
4	1	2	2	2	2	2
5	1	2	2	2	3	3
6	1	3	3	3	1	1
7	1	3	3	3	2	2
8	1	3	3	3	3	3
9	2	1	2	3	1	2
10	2	1	2	3	2	3
11	2	1	2	3	3	1
12	2	2	3	1	1	2
13	2	2	3	1	2	3
14	2	2	3	1	3	1
15	2	3	1	2	1	2
16	2	3	1	2	2	3
17	2	3	1	2	3	1
18	3	1	3	2	1	3
19	3	1	3	2	2	1
20	3	1	3	2	3	2
21	3	2	1	3	1	3
22	3	2	1	3	2	1
23	3	2	1	3	3	2
24	3	3	2	1	1	3
25	3	3	2	1	2	1
26	3	3	2	1	3	2

Table 3. The best values of factors in HGA.

Factors	The best values
Initial temperature	1000
Cooling rate	0.98
P_{cr}	0.9
P_{mu}	0.4
Population size	150
SA rate	0.04

and 0.07 for PSO, 0.002 for HGA, and 0.08 for standard GA in relation to instances with 30 activities. Thus, it can be concluded that most of the time, relative deviation from the best solution in HGA is much lower than average RDs for PSO and GA.

5. Conclusion

In this paper, a class of resource-constrained project scheduling problems was introduced to cope with project uncertainty, in which the goal was to maximize the project NPV. Project managers can use the proposed model for planning the projects to maximize the NPV of the project under uncertainty of the real word. For this purpose and due to the lack of available information about project activities, the parameters such as duration, cash flows, resource consumption of the activities, and interest rate were assumed as fuzzy numbers. To solve the problem, a tuned hybrid GA with SA algorithm was proposed. The performance of the algorithm was evaluated by comparing it with two well-known metaheuristic algorithms. The result showed the effectiveness of the proposed algorithm. Some extensions of this research as a future study might be of interest. While in this paper we only considered the "payments at pre-specified event nodes", some other payment models such as progress payments and payments at pre-specified time points may be considered in the project. Furthermore, it would be interesting to generalize the model to include non-renewable resources as well as multiple execution modes for each activity.

Table 4. Results of experiments.

Number of activities	Most likely discount rate	Number of resources	Number of instances	#n (PSO)	#n (GA)	#n (HGA)	HGA (%)			PSO (%)			GA (%)		
							ARD	MRD	CPU time	ARD	MRD	CPU time	ARD	MRD	CPU time
10	0.02	2	10	0	0	10	0.00	0.00	3.37	7.61	15.82	0.42	4.21	12.65	1.02
	0.02	3	10	0	2	8	0.43	2.71	3.58	6.59	15.39	0.56	2.42	5.39	1.21
	0.02	4	10	0	1	9	0.34	3.34	4.23	5.78	15.81	1.03	3.21	10.20	1.53
	0.03	2	10	0	0	10	0.00	0.00	3.59	11.5	22.90	0.32	4.28	11.25	1.00
	0.03	3	10	0	1	9	0.32	3.34	4.01	8.87	23.22	0.47	3.78	9.79	1.32
	0.03	4	10	0	0	10	0.00	0.00	4.52	9.19	23.22	0.55	4.03	12.2	1.45
	Total		**60**	**0**	**4**	**56**	**0.18**	**3.34**	**4.28**	**8.25**	**23.22**	**0.55**	**3.65**	**11.25**	**1.25**
20	0.02	2	10	0	0	10	0.00	0.00	3.47	6.85	15.25	0.53	4.94	11.3	1.26
	0.02	3	10	0	1	9	0.34	3.38	4.58	6.35	12.08	1.02	2.65	5.49	1.55
	0.02	4	10	2	2	6	0.82	5.03	5.18	3.25	9.18	1.28	6.25	26.67	2.02
	0.03	2	10	1	2	7	0.73	3.41	4.12	10.51	19.98	0.43	6.47	18.61	1.14
	0.03	3	10	0	2	8	0.49	4.36	4.48	9.31	17.91	0.58	6.62	15.86	1.35
	0.03	4	10	1	0	9	0.23	2.25	5.03	7.06	17.55	1.34	8.45	14.42	2.13
	Total		**60**	**4**	**7**	**49**	**0.68**	**5.03**	**4.47**	**7.12**	**19.98**	**1.26**	**5.80**	**26.67**	**1.57**
30	0.02	2	10	0	1	9	0.11	1.06	4.52	7.79	17.95	1.17	6.58	21.87	1.71
	0.02	3	10	0	0	10	0.00	0.00	5.54	5.58	11.83	1.29	6.32	12.95	2.17
	0.02	4	10	0	1	9	0.12	1.18	6.09	3.30	8.27	1.59	6.30	22.27	2.41
	0.03	2	10	0	2	8	0.43	2.73	4.12	9.07	17.52	1.08	7.54	25.76	2.01
	0.03	3	10	0	0	10	0.00	0.00	6.23	10.85	23.12	1.33	11.35	21.10	1.96
	0.03	4	10	1	1	8	0.57	3.41	6.47	8.28	15.6	1.97	10.68	24.90	2.13
	Total		**60**	**1**	**5**	**54**	**0.20**	**3.41**	**5.49**	**7.47**	**23.12**	**1.40**	**8.12**	**26.67**	**2.06**
Grant Total			**180**	**5**	**16**	**159**	**0.35**	**5.03**	**4.74**	**8.25**	**23.12**	**1.07**	**5.85**	**26.67**	**2.12**

References

1. Patterson, J.H. and Roth, J.W. "Scheduling a project under multiple resource constraints: A zero-one programming approach", *AIIE Transactions*, **4**, pp. 449-455 (1976).
2. Hartmann, S. and Briskorn, D. "A survey of variants and extensions of the resource-constrained project scheduling problem", *European Journal of Operational Research*, **207**(1), pp. 1-14 (2010).
3. Christofides, N., Alvarez-Valdes, R. and Tamarit, J.M. "Project scheduling with resource constraints: A branch and bound approach", *European Journal of Operational Research*, **29**(3), pp. 262-273 (1987).
4. Demeulemeester, E. and Herroelen, W. "The discrete time/resource trade-off problem in project networks: a branch-and-bound approach", *IIE Transactions*, **32**(11), pp. 1059-1069 (2000).
5. Hartmann, S. and Kolisch, R. "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem", *European Journal of Operational Research*, **127**, pp. 394-407 (2000).
6. Hartmann, S.A. "Competitive genetic algorithm for resource constrained project scheduling", *Naval Research Logistics*, **45**(7), pp. 733-750 (1998).
7. Bouleimen, K. and Lecocq, H. "A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple mode versions", *European Journal of Operational Research*, **149**, pp. 268-81 (2003).
8. Merkle, D., Middendorf, M. and Schmeck, H. "Ant colony optimization for resource-constrained project scheduling", *IEEE Transactions on Evolutionary Computation*, **6**(4), pp. 333-346 (2002).
9. Jarboui, B., Damak, N., Siarry, P. and Rebai, A.A "Combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems", *Applied Mathematics and Computation*, **195**(1), pp. 299-308 (2008).
10. Berthold, T., Heinz, S., Lübbecke, M.E., Möhring, R.H. and Schulz, J. "A constraint integer programming approach in resource constraint project scheduling", *Computer Science*, **6**(140), pp. 313-317 (2010).
11. Khalili, S., Najafi, A.A. and Niaki, S.T.A. "Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: two meta-heuristic algorithms", *The International Journal of Advanced Manufacturing Technology*, **69**(1-4), pp. 617-626 (2013).
12. De Reyck, B. "A branch-and-bound procedure for the resource-constrained project scheduling problem with

generalized precedence relations", *European Journal of Operational Research*, **111**(1), pp. 152-174 (1998).

13. Ozdamar, L. and Ulusoy, G. "A survey on the resource-constrained project scheduling problem", *IIE Transactions*, **27**, pp. 574-586 (1995).
14. Icmeli, O. and Erenguc, S.S. "A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows", *Computers & Operations Research*, **21**(8), pp. 841-853 (1994).
15. Kimms, A. "Maximizing the net present value of a project under resource constraints using a Lagrangian relaxation based heuristic with tight upper bounds", *Annals of Operations Research*, **102**(1-4), pp. 221-236 (2001).
16. Gu, H., Schutt, A. and Stuckey, P.J. "A Lagrangian relaxation based forward-backward improvement heuristic for maximising the net present value of resource-constrained projects", *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 340-346, Springer, Berlin, Heidelberg (2013).
17. Vanhoucke, M., Demeulemeester, E. and Herroelen, W. "On maximizing the net present value of a project under renewable resource constraints", *Management Science*, **47**(8), pp. 1113-1121 (2001).
18. Chen, W.N., Zhang, J., Chung, H.S.H., Huang, R.Z. and Liu, O. "Optimizing discounted cash flows in project scheduling-an ant colony optimization approach", *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, **40**(1), pp. 64-77 (2010).
19. Waligóra, G. "Simulated annealing and tabu search for discrete-continuous project scheduling with discounted cash flows", *RAIRO-Operations Research*, **48**(01), pp. 1-24 (2014).
20. Igelmund, G. and Radermacher, F.J. "Preselective strategies for the optimization of stochastic project networks under resource constraints", *Applied Mathematics*, **13**(1), pp. 1-28 (1983).
21. Tsai, Y.W. and Gemmil, D. D. "Using tabu search to schedule activities of stochastic resource- constrained projects", *European Journal of Operational Research*, **111**, pp. 129-14 (1998).
22. Deblaere, F., Demeulemeester, E. and Herroelen, W. "Proactive policies for the stochastic resource-constrained project scheduling problem", *European Journal of Operational Research*, **214**(2), pp. 308-316 (2011).
23. Tseng, C.C. and Ko, P.W. " Analysis of scheduling uncertainty in stochastic resource constrained network", In *Proceedings of the 2012 Second International Conference on Electric Technology and Civil Engineering* IEEE Computer Society, pp. 345-349 (2012).
24. Fliedner, T., Gutjahr, W.J., Kolisch, R. and Melchiors, P. "Solving the dynamic stochastic resource constrained multi-project scheduling problem with SRCPSP-methods", *13th Int. Conf. on Project Management and Scheduling*, Leuven, Belgium, pp. 148-151 (2012).
25. Lootsma, F.A. "Stochastic and fuzzy PERT", *European Journal of Operational Research*, **43**, pp. 174-183 (1989).
26. Dubois, D. and Prade. H. "Fuzzy numbers: An overview", J.C. Bezdek, Ed., In *Analysis of Fuzzy Information*, CRC Press, pp. 3-39 (1987).
27. Janczura, M. and Kuchta, D. " Proactive and reactive scheduling with fuzzy activity times", *13th Int. Conf. on Project Management and Scheduling*, Leuven, Belgium, pp. 175-178 (2012).
28. Wang, J. "A fuzzy project scheduling approach to minimize schedule risk for product development", *Fuzzy Sets and Systems*, **127**(2), pp. 99-116 (2002).
29. Hapke, M. and Slowinski, R. "A DSS for resource-constrained project scheduling under uncertainty", *Decision Systems*, **2**(2), pp. 111-128 (1993).
30. Hapke, M., Jaskivicz, A. and Slowinski, R. "Fuzzy project scheduling system for software development", *Fuzzy Sets and Systems*, **67**(1), pp. 101-117 (1994).
31. Wang, H., Lin, D. and Li, M. "A genetic algorithm for solving fuzzy resource-constrained project scheduling", *Computer Science*, **36**(12), pp. 171-180 (2005).
32. Atli, O. and Kahraman, C. "Fuzzy resource constrained project scheduling using taboo search algorithm", *International Journal of Intelligent Systems*, **27**(10), pp. 873-907 (2012).
33. Kaur, J. and Kumar, A. "Exact fuzzy optimal solution of fully fuzzy linear programming problem with unrestricted fuzzy variables", *Applied Intelligent*, **37**(4), pp. 145-154 (2012).
34. Masmoudi, M. and Hait, A. "Project scheduling under uncertainty using fuzzy modeling and solving techniques", *Engineering Applications of Artificial Intelligence*, **26**(1), pp. 135-149 (2013).
35. Vartouni, A.M. and Khanli, L.M. "A hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem", *Journal of Intelligent and Fuzzy Systems*, **26**(3), pp. 1103-1112 (2014).
36. Xu, J. and Feng, C. "Multimode resource-constrained multiple project scheduling problem under fuzzy random environment and its application to a large scale hydropower construction project", *The Scientific World Journal*, **2014**, pp. 1-20 (2014).
37. Blazewicz, J., Lenstra, J.K. and Rinooy Kan, A.H.G. "Scheduling subject to resource constraints: Classification and complexity", *Discrete Applied Mathematics*, **5**(1), pp. 11-24 (1983).
38. Najafi, A.A. and Niaki, S.T.A. "A genetic algorithm for resource investment problem with discounted cash flows", *Applied Mathematics and Computation*, **183**(2), pp. 1057-1070 (2006).

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

40. Shahsavar, M., Najafi, A.A. and Niaki, S.T. "A statistical design of genetic algorithms for combinatorial optimization problems", *Mathematical Problems in Engineering*, **2011**, pp. 1-17 (2011).

41. Zadeh, L.A. "Fuzzy sets", *Information and Control*, **8**(3), pp. 338-353 (1965).

42. Yang, M.S., Hwang, P.Y. and Chen, D.H. "Fuzzy clustering algorithms for mixed feature variables", *Fuzzy Sets and Systems*, **141**(2), pp. 301-317 (2004).

43. Wang, X. and Kerre, E.E. "Reasonable properties for the ordering of fuzzy quantities", *Fuzzy Sets and Systems*, **118**(3), pp. 375-385 (2001).

44. Thorani, Y.L.P., Rao, P.P.B. and Shankar, N.R. "Ordering generalized trapezoidal fuzzy numbers using orthocentre of centroids", *International Journal of Algebra*, **6**(22), pp. 1069-1085 (2012).

45. Fathallahi, F. and Najafi, A.A. "A multi-objective approach based on frank-wolf algorithm for fully fuzzy project scheduling with discounted cash flows", *Technical Journal of Engineering of Applied Sciences*, **3**(22), pp. 3076-3084 (2013).

46. Khoshjahan, Y., Najafi, A.A. and Afshar-Nadjafi, B. "Resource constrained project scheduling problem with discounted earliness-tardiness penalties: Mathematical modeling and solving procedure", *Computers & Industrial Engineering*, **66**(2), pp. 293-300 (2013).

47. Tao, Z. "TSP problem solution based on improved genetic algorithm", *9th Int. Conf. in Natural Computation*, Jinan, Shandong, China, pp. 686-690 (2008).

48. Nirmala, D.G. and Ramprasad, D. "Innovative PMX-CX-Operators for GA to TSP", *International Journal of Scientific and Research Publications*, **2**(10), pp. 1-6 (2012).

49. Kirkpatrick, S. "Optimization by simulated annealing: Quantitative studies", *Journal of Statistical Physics*, **34**(5-6), pp. 975-986 (1984).

50. Weijun, X., Zhiming, W., Wei, Z. and Genke, Y. "A new hybrid optimization algorithm for the job-shop scheduling problem", In *American Control Conference, 2004*, Proceedings of the 2004, Boston, USA, pp. 5552-5557, IEEE (2004).

51. Azardoost, E.B. and Imanipour, N. "A hybrid algorithm for multi objective flexible job shop scheduling problem", *2th. Int. Conf. on Industrial Engineering and Operations Management*, Kuala Lumpur, Malaysia, pp. 795-801 (2011).

52. Kolisch, R., Sprecher, A. and Drexl, A. "Characterization and generation of a general class of resource constrained project scheduling problems", *Management Science*, **41**(10), pp. 693-1703 (1995).

53. Taguchi, G. *Introduction to Quality Engineering: Designing Quality into Products and Processes*, Transportation Research Board, Washington (1986).

54. Khalilzadeh, M., Kianfar, F., Shirzadeh Chaleshtari, A., Shadrokh, S. and Ranjbar, M. "A modified PSO algorithm for minimizing the total costs of resources in MRCPSP", *Mathematical Problems in Engineering*, **2012**, pp. 1-18 (2012).

55. Zhang, H., Li, X., Li, H. and Huang, F. "Particle swarm optimization-based schemes for resource-constrained project scheduling", *Automation in Construction*, **14**(3), pp. 393-404 (2005).

56. Wang, W. and Zhao, G.J. "Particle swarm optimization for resource-constrained project scheduling problem", *Journal of Harbin Institute of Technology*, **24**(1), pp. 83-92 (2007).

57. Hartmann, S. "A competitive genetic algorithm for resource-constrained project scheduling", *Naval Research Logistics (NRL)*, **45**(7), pp. 733-750 (1998).

58. Padman, R. and Roehrig, S.F. "A genetic programming approach for heuristic selection in constrained project scheduling", *Interfaces in Computer Science and Operations Research*, Springer, US, pp. 405-421 (1997).

Biographies

Fatemeh Fathallahi received her BS and MS degrees in Industrial Engineering from Khaje Nasir University of Technology, Tehran, Iran, in 2011 and 2013, respectively. Her research interests are mainly in the field of project scheduling and management.

Amir Abbas Najafi received his BS degree in Industrial Engineering from Isfahan University of Technology in 1996, and his MS and PhD degrees in Industrial Engineering from Sharif University of Technology in 1998 and 2005, respectively. He is currently an Associate Professor at K. N. Toosi University of Technology. His research interests include applied operations research, project scheduling and management, and portfolio selection models.