Scientia Iranica E (2014) 21(6), 2436–2447

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

Solution procedure for generalized resource investment problem with discounted cash flows and progress payment

B. Afshar-Nadjafi[a,*], A. Parsanejad[a], V. Hajipour[b] and A. Nobari[b]

a. *Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
b. *Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran.*

Received 23 July 2013; received in revised form 20 November 2013; accepted 10 March 2014

KEYWORDS
Project scheduling;
Resource investment;
Net present value;
Simulated annealing;
Progress payment.

Abstract. In this paper, the Resource Investment Problem (RIP) has been studied in which the availability levels of the resources are considered as decision variables. The objective is to maximize the net present value of a project by a given project deadline subject to progress payments. The project has activities interrelated by Generalized Precedence Relations (GPR's), which require a set of renewable resources. A non-linear mixed integer programming formulation is proposed for the problem. The problem formed in this way is an NP-hard one, leading to the use of a Modified version of the SA (MSA) algorithm in order to obtain a satisfying solution, based on its hybridization with a local search procedure. In order to improve the MSA, the Taguchi technique is executed to tune its parameters. Moreover, the Genetic Algorithm (GA) is also applied to validate the performance of the proposed algorithm. Finally, for examining the algorithm's performance, the Relative Percent Deviation (RPD) index is applied for comparison. The results of the performance analysis of the proposed MSA show the efficiency of the presented algorithm.

1. Introduction

The Project Scheduling Problem (PSP) is the most investigated area in the fields of operations research and management science [1]. Finding a schedule for the activities of a project is subject to some side constraints, such as precedence constraints and resource constraints, although there are many other factors also involved. The classical Resource Constrained Project Scheduling Problem (RCPSP) is known as a project with a set of n activities, numbered 1 to n, where each activity has to be processed without interruption to accomplish the project [2].

The Resource Investment Problem (RIP) is one of the most important problems in the area of project scheduling. In the real world, at least two factors are included for implementation of the projects; project owner and contractor. Financial payments are achieved as a result of the type of project contracts and agreements between these two factors. In the literature, there are various models to determine how the contractor receives the employer's payment, which are proposed by Ulusoy et al. [3] in four types:

I. Lump-Sum Payment (LSP) where the contract amount is paid to the employer at the end of a successful project.

II. Payment at Event Concurrency (PEO) where the

*. *Corresponding author. Tel./Fax: +98 281 3670051*
E-mail addresses: afsharnb@alum.sharif.edu (B. Afshar-Nadjafi); parsaabozar@yahoo.com (A. Parsanejad); v.hajipour@basu.acir (V. Hajipour); arashnob@basu.acir (A. Nobari)

contractor agrees to pay the employer during project implementation at predetermined milestones.

III. Payment at Equal Time Interval (ETI) where payments are paid at equal time intervals, and the last payment paid when the project is completed.

IV. Progress Payment (PP), where payments are paid at regular intervals according to the project progress.

Tavares [2] categorized different types of objective functions utilized in PSPs, such as: minimizing project duration, maximizing the net present value of the project cash flow, and maximizing project resource utilizations. Since most projects consist of lengthy running times, the magnitude volume of cash flow during the project implementation time, high interest rates, limited capital, and large projects and activities, are among the best factors for a maximum amount of project profit. In PSP literature, most research concentrates on minimizing project duration. In this article, to make a model more realistic, maximizing the total Net Present Value (NPV) is considered as the main objective function. Thus, the scheduling problem of resource constrained projects with discounted cash flow (RCPSP-DCF) has recently attracted more attention. The NPV of a project is considered to be maximized, even if the project duration may not be minimized [4]. In addition, minimum, as well as maximum, time lags may be given between different activities. This problem is also known as the RIPDCFPP/GPR. It should be mentioned that Generalized Precedence Relations (GPRs) are applicable for cases where activities require fixed or simultaneous starting or completion times, non-delay execution, mandatory overlap with other activities, time-varying resource requirements or ready times and deadlines [5].

Vanhoucke et al. [6] considered the problem of progress payments in project scheduling problems. Since, in a large number of projects, cash flow depends on their occurrence in different ways, recent research has focused on cases where activity cash flow depends on the completion times of the corresponding activities.

The Resource Investment Problem (RIP) is a close variant of RCPSP, which consists of scheduling project activities in such a way that the total cost of the renewable resources required for completing the project by a pre-specified project deadline is minimized. Mohring [7] investigated the issue of investment in resources and demonstrated the NP-hardness of the problem. He developed an exact solution method, based on examples of graphs, with 16 activities and four resources. Shadrokh and Kianfar [8] presented the issue of investment in resources and proposed the Genetic Algorithm (GA) for its solution. Also, the computational results demonstrated the better performance of the developed GA. The problem of investment, in resources with positive and negative cash flow, for maximizing the NPV of project cash flow (RIPDCF), was introduced by Najafi and Niaki [9], and the GA was suggested for the corresponding model. Najafi et al. [10] developed the source investment problem with positive and negative cash flow for maximizing the net present value of project cash flow, and generalized precedence relationships (RIPDCF/GPR). Also, they proposed a GA for solving the model. Bianco and Caramia [11] proposed a B&B algorithm to solve RCPSP, with generalized precedence relationships and a minimum makespan objective. Recently, Afshar-Nadjafi et al. [12] developed a GA to optimize a Mode Identity Resource Constrained Project Scheduling Problem (MIRCPSP) in which the set of project activities is partitioned into disjoint subsets. Rahimi et al. [13] proposed three meta-heuristic algorithms, namely, the imperialist competitive algorithm, SA, and differential evolution, to solve MIRCPSP.

Since the RCPSP is severely NP-hard, soft computing approaches are efficient. Paraskevopoulos et al. [14] proposed a new solution representation and an evolutionary algorithm inspired by an event list-base for solving the RCPSP. He et al. [15] proposed some meta-heuristics for multi-mode capital constrained project payment scheduling. Xiao et al. [16] developed an efficient ant colony algorithm to optimize software project scheduling problems. Several meta-heuristic algorithms have since been developed which combine rules and randomness, mimicking natural phenomena. These phenomena include biological evolutionary processes, for example, the evolutionary algorithm [17], GA [12,18], animal behavior [19,20], the physical annealing process [13,21], the social process [22], and so on.

This research studies the Resource Investment Problem with Discounted Cash Flow (RIPDCF) including generalized precedence relations and a progress payment, called RIPDCFPP/GPR. To do so, a mathematical model, as a non-linear mixed integer programming, is proposed. As stated, the proposed model is NP-hard, therefore, a Modified Simulated Annealing (MSA) algorithm is proposed to solve the problem. In this respect, we extended SA, with regard to hybridizing it with a local search procedure. The Taguchi approach is also applied to tune the parameters of the algorithms. To demonstrate the applicability of the proposed algorithms, some problems with various sizes are generated. For performance evaluation of the MSA, a proper index is applied and, finally, the results are compared with one of the best-developed algorithms in the literature, called GA.

The rest of this paper is organized as follows: In Section 2, the formulation of RIPDCFPP/GPR is represented. Two meta-heuristic algorithms, namely, GA and MSA, are described in Section 3. Moreover, in order to realize the performance of the proposed model, result analysis and parameter tuning are provided in Section 4. Finally, in Section 5, the conclusion and remarks for future work are discussed.

2. Model description

In this section, the problem, notations, parameters, and decision variables are defined and then the model is developed.

2.1. Problem definition

Neumann and Zimmermann [23] classified the resource leveling problem into three categories. In one of these categories, which is called the resource investment problem, resource levels are determined at the beginning of the project, which will not change up to the end of the project. Therefore, in this problem, it is assumed that the resource leveling and scheduling of activities should be done in such a way that the objective function is optimized. A simple objective function of the problem is to minimize the cost of resource employment. Suppose a project with $n+2$ activities in which 0th and $(n+1)$th activities are dummies and represent the start and end point of the project, respectively. We assume that there exists at least one path with nonnegative length from node 0 to every other node, and at least one path from every node i to node $n+1$, which is equal to or larger than d_i. If there are no such paths, we can insert arcs $(0,i)$ or $(i,n+1)$ with weight zero or d_i, respectively. In this type of study, projects are presented by a directed graph, $G(V,E)$, in which a set of nodes, V, indicates project activities, and the set of arcs, E, indicates the precedence relations. The deterministic duration of activity i is denoted by d_i. To complete the project activities, K types of renewable resources are needed. The cost of employing each unit of resource k per unit of time is equal to C_k. Meanwhile, the amount of kth resource required for the activity i per unit of time is r_{ik}. Also, each activity, i, requires a fixed cost, including material costs, overhead costs, etc., over activity execution. The amount of fixed cost at period t for each activity, i, is presented by F_{it}. During the project progress time, the inflows are determined based on the amount of completed work of the activities. Meanwhile, the interest rate per unit time is equal to α. The precedence relation between the activities is assumed as generalized. Finally, the predetermined deadline for completing the project is denoted by DD.

Here, we illustrate the problem by an instance. The corresponding AON project network is shown in Figure 1. There are 7 real activities (0 and 8 are dummy activities) and two resource types. The number above the node denotes the activity duration, while the numbers below denote the resources requirements, respectively. However, the numbers above the arcs denote minimal start-start time lags, which are accepted as a standard form of GPRs.

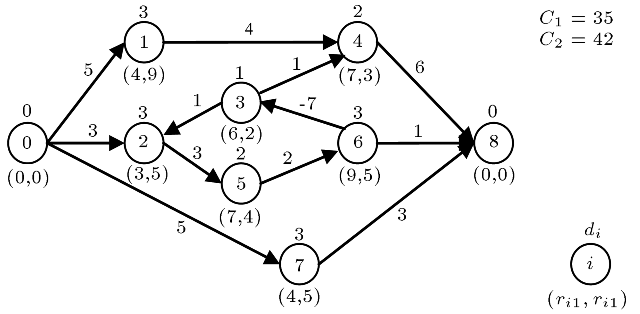

Figure 1. Problem instance for the RIPDCFPP/GPR.

2.2. Parameters and notations

The notations, parameters, and decision variables are considered for the RIPDCFPP/GPR, as follows:

d_i	Duration of activity i; $i=0,1,2,\cdots,n+1$;
C_k	Employment cost per unit of resource type k per unit of time, $k=1,2,\cdots,K$;
r_{ik}	The required level of activity, i, from resource type, k, per period;
F_{it}	Fixed cost of activity, i, at period t;
w_{it}	Completed portion work during period, t, for activity i;
C_i^+	Progress payment cash inflow of activity, i, at the end of some review period incurred;
S_i	Starting time of activity i, $i=0,1,2,\cdots,n+1$;
R_k	Required level of resource type, k, to be provided, $k=1,2,\cdots,K$;
SR_k	Employment time of resource k (start time of resource scheduling), $k=1,2,\cdots,K$;
FR_k	Releasing time of resource k (end time of resource scheduling), $k=1,2,\cdots,K$;
$U(k)$	Set of activities that use a kth resource;
$PB(t)$	Payment at t when a set of activities ends;
$PR(k)$	Series of activities that use the kth resource and have no precedence;
$UR(k)$	Series of activities that use the kth resource and have no successor;
X_{it}	A binary variable: It is one if activity i is started at period t, otherwise is zero,

$i=0,1,2,\cdots,n+1$ and $t=1,2,\cdots,DD$

y_1 and y_2 are binary variables.

2.3. RIPDCFPP/GPR formulation

As mentioned earlier, the first issue is to determine resource employment levels and the schedule of activities, so that the desired objective function is optimized. The objective functions consist of three parts. In these parts, ES_i and LS_i, respectively, indicate the earliest and latest starting times of activity, i. Reviewing points are occurred in $T, 2T, \cdots, mT$ under the following conditions: $mT \geq DD$ and $(m-1)T \leq DD$. It should be noticed that when $mT > DD$, the last period is equal to DD. Variable $w_{it}(t = T, 2T, \cdots, mT$ and $1 \leq i \leq n)$ is defined as part of the completed work of activity i during period $[t - T, t]$. Therefore, cash inflow is calculated as follows [6]:

$$\sum\nolimits_{t=T,2T,3T,\cdots mT} w_{it} C_i^{+} e^{-\alpha t}. \tag{1}$$

According to Relation (1), the first part of the objective function can be presented as follows:

$$\text{I.} \qquad +\sum_{i=0}^{n} \left(\sum\nolimits_{t=T,2T,3T,\cdots mT} w_{it} C_i^{+} e^{-\alpha t} \right). \tag{2}$$

Also, the second part of the objective function, which is outgoing discounted cash flow, is:

$$\text{II.} \qquad +\sum_{i=0}^{n} \left(-\sum_{ES_i}^{LS_i} \sum_{u=0}^{d_i - 1} F_{iu_i}^{+} e^{-\alpha(u+t)} x_{it} \right). \tag{3}$$

In this part, F_{it} denotes different items of cost, such as material costs, overheads, etc. As mentioned earlier, we consider all of these items of cost as a fixed combined value of costs, which is denoted by F_{iu}. The third part denotes the cost of resources:

$$\text{III.} \qquad -\sum_{k=1}^{K} C_K R_K \left(\frac{e^{-\alpha SR_K} - e^{-\alpha FR_K}}{e^{-\alpha}} \right). \tag{4}$$

Therefore, the objective function can be introduced as follows:

$$\max(Z) = \sum_{i=0}^{n} \left(\sum\nolimits_{t=T,2T,3T,\cdots mT} w_{it} C_i^{+} e^{-\alpha t} - \sum_{ES_i}^{LS_i} \sum_{u=0}^{d_i - 1} F_{iu} e^{-\alpha(u+t)} x_{it} \right) - \sum_{k=1}^{K} C_K R_K \left(\frac{e^{-\alpha SR_K} - e^{-\alpha FR_K}}{e^{-\alpha}} \right). \tag{5}$$

Subject to:

$$W_{it} = \left(\frac{t - \sum_{ES_i}^{LSi} t x_{it} + d_i}{d_i} \right) y_1 + \left(\min \left\{ 1, \frac{\sum_{ES_i}^{LSi} t x_{it+T-t)}}{d_i} \right\} \right) y_2, \tag{6}$$

$$y_1 + y_2 \leq 1, \tag{7}$$

$$\sum_{ES_i}^{LSi} t x_{it} + l_{ij} \leq \sum_{ES_i}^{LSi} t x_{it} \qquad \forall (i,j) \in G, \tag{8}$$

$$\sum_{ES_i}^{LSi} t x_{n+1,t} \leq DD, \tag{9}$$

$$t \geq \sum\nolimits_{t=ES_i}^{LS_i} x_{n+1,t} + w_{it} d_i, \qquad \forall i \in pB(t)$$

$$t = T, 2T, 3T, \cdots mT, \tag{10}$$

$$SR_K \leq \sum_{t=ES_i}^{LSi} x_{n+1,t} \qquad \forall i \in pR(K)$$

$$k = 1, 2, 3, \cdots K, \tag{11}$$

$$FR_K \geq \sum_{t=ES_i}^{LSi} x_{n+1,t} + d_i \qquad \forall i \in UR(K)$$

$$k = 1, 2, 3, \cdots K, \tag{12}$$

$$\sum_{i=0}^{n+1} \sum_{l=\max(t-d_i+1, ES_i)}^{\min(t, LS_i)} r_{ik} x_{il} \leq R_K,$$

$$t = 0, 1, 2, 3, \cdots, DD \qquad \text{and} \qquad k = 1, 2, 3, \cdots, K, \tag{13}$$

$$\sum_{t=ES_i}^{LSi} x_{it} = 1 \qquad i = 0, 1, 2, 3, \cdots n+1, \tag{14}$$

$$S_i = \sum_{t=ES_i}^{LS_i} t x_{it} \qquad i = 0, 1, 2, \cdots, n+1, \tag{15}$$

$$S_i, R_k, SR_k, FR_k \geq 0 \; y_1, y_2, x_{it} = \{0, 1\} \qquad \forall k, i, t. \tag{16}$$

Constraints (6) and (7) denote the percentage of work completed constraints. Constraint (8) is a precedence constraint. The project completion deadline constraint is denoted by Constraint (9). Constraint (10) indicates project payment limits. Employment and release times of resources are represented by Constraints (11) and (12). Constraint set (13) shows the limitation of resource consumption. Constraint (14) forces one completion time for each activity. Equation (15) computes the start time of each activity. Constraint (16) specifies that the decision variables, y_1, y_2 and x_{it}, are binary, while S_i, SR_k, FR_k and R_k are integer.

It has been proved by [7] that the Resource Investment Problem (RIP) belongs to an NP-hard class of problems. Therefore, since RIPDCFPP/GPR is an extended model of RIP in both constraints and objective, this model also belongs to the NP-hard class of problems. Therefore, we utilize a soft-computing approach to solve the model. In the next section, the proposed algorithm is also described in detail.

3. Solving methodologies

As mentioned in the previous section, our model belongs to the NP-hard class of optimization problems. Generally, to solve such models, different types of meta-heuristic algorithms are applied. In this paper, we proposed a modified version of the SA algorithm called MSA. In the rest of this section, our proposed MSA is first explained, then, GA is described as well. At the end, we apply a random search approach to demonstrate the applicability of both meta-heuristics.

3.1. Modified simulated annealing algorithm

The SA algorithm searches the solution area according to the stochastic mechanism of the physical annealing process in metallurgy which is used for solving complex combinatorial problems [21]. Generally, in this algorithm, the objective value of a solution, usually called the cost factor, is equivalent to the internal energy state. This algorithm starts with a high temperature and selects the initial solution randomly. Then, it calculates a new solution within the neighborhood of the current solution. New solutions are accepted based on a probability that depends on the difference between the corresponding costs and on the current temperature. Correspondingly, high temperatures let the new solutions be accepted more easily. The probability of accepting higher costs decreases within the optimization process through an effective decreasing rate of temperature [24,25]. In order to enhance the performance of the SA, the steps of the proposed MSA, which is combined with a local search procedure, are described in the following subsections.

3.1.1. Input parameters

The performance of MSA greatly depends on the following parameters:

I. Initial temperature, T_0 (a starting point for calculating the value of temperature in different iterations);
II. Decreasing function of temperature, $T(t)$;
III. Number of iterations in each temperature;
IV. Conditions for reaching the equilibrium system;
V. Stop criterion.

- Decreasing function of temperature, $T(t)$:
  One of the major factors having an important effect on the performance of the MSA is the decreasing function of the temperature. In this paper, the new value of the temperature is calculated through Eq. (17):

$$T_k = \alpha T_{k-1}, \qquad \alpha < 1. \tag{17}$$

- Conditions for reaching the equilibrium system:
  In MSA, before temperature reduction, the equilibrium condition should be checked. If this condition is met, the next temperature is selected; otherwise iterations will continue at the same temperature. In this paper, the system equilibrium is calculated from the following equation [25]:

$$N_s = N_s(1 + h \times \text{step}), \tag{18}$$

  where N_s denotes the number of iterations needed for achieving the equilibrium condition.
- Stop criterion.

The stop criterion in this paper is a predetermined number of iterations.

3.1.2. Representations and evaluation

In MSA, like any other meta-heuristic algorithms, representation is severely important. It is better that the representation vector is in a way that makes all constraints feasible. To do so, we utilize a representation in which a solution is modeled through $n + 2$ component vectors (two shows the number of dummy activities) and each component shows the starting time (s_i) of the project activities. A scheme of this representation is shown in Eq. (17).

$$RE = (s_0, s_1, \cdots, s_{n+1}). \tag{19}$$

To have a feasible representation, the starting times should consider predecessor and successor activities. It means that activities should not start before the earliest start time, $E(j)$, and should not last longer than the latest start time, $L(j)$ or $E(j) \le S_i \le L(j)$.

Now, in order to evaluate the representations, the fitness value obtained by the objective function should be calculated. To do so, for each representation, we need to obtain both the activity starting times and the resource plan. The activity starting times are directly obtained from the components of the representation. Finally, for the resource plan, the resources at representation scheduling are entered that will determine the requirement level, the employment time and the releasing time of the resources.

3.1.3. Initial population

In order to create presentations, first, we should calculate the earliest and latest start times through the following algorithms [26].

It should be mentioned that the earliest start time of all activities is set at zero, which means that for calculating the earliest start time of all activities, *Earliest* (0) should be executed. *Earliest* (i) is a recursive function as follows:

1. List all the successor activities of activity i and determine the first successor activity.
2. Call the first successor by j, and the weight of arc i to j by d_{ij}. Now, if $E(i) + d_{ij} \geq E(j)$, consider $E(j) = E(i) + d_{ij}$ and execute *Earliest* (j).
3. Select the next successor and go back to step 2. The algorithm will end when all activities of the list are assessed.

For calculating the latest start times again, we first should set all $L(.)$ to T (which denotes the maximum execution time of the project). The function of the latest start time is denoted by *Latest* (.). Now, for calculating the latest start time, the *Latest* ($n + 1$) function should be calculated. *Latest* (i) is a recursive function as follows:

1. List all the predecessor activities of activity i and determine the first predecessor activity.
2. Call the first predecessor by j, and the weight of arc i to j by d_{ji}. Now, if $L(i) - d_{ji} < L(j)$, consider $L(j) = L(i) - d_{ji}$ and execute *Latest* (j).
3. Select the next predecessor and go back to step 2. The algorithm will end when all activities of the list are assessed.

After obtaining all earliest and latest times, or calculating all $E(i)$ and $L(i)$ for producing a random representation, the following algorithm is used.

1. Consider $i = 0$;
2. Generate a random number called *Rand* within the interval $[E(i), L(i)]$;
3. The start time of the activity $S_i = E(i) = L(i) = Rand$;
4. Execute *Earliest* (i);
5. Execute *Latest* (i);
6. $i = i + 1$;
7. If $i = n + 1$, the algorithm ends, otherwise go to step 2.

Earliest and latest start times can also be calculated through a Floyd-Warshall algorithm [27]. The Floyd-Warshall algorithm is of time complexity, $O(n^3)$. Calculation of the earliest start times can be related to the test for the existence of a time-feasible schedule. A time-feasible schedule for a project exists if the standard form of project network (G) has no cycle of positive length. Such cycles would enable us to compute start times for activities which satisfy generalized precedence relations (Constraint set (8)). In the Floyd-Warshall algorithm, matrix $\Pi = [\pi_{ij}]$ is often referred to as the distance the matrix is computed, where π_{ij} denotes the longest path length from node i to node j. A positive path length from node i to itself indicates the existence of a cycle of positive length, and, consequently, the non-existence of a time-feasible schedule.

To give an equal chance to each point of the solution space, the algorithm has been modified as follows:

1. Consider the activity list (act-list) and:
2. If the list is empty, the algorithm is ended, otherwise, select randomly one of the activities from the act-list, denote it, and omit it from the mentioned list;
3. Generate a random number called *Rand* within the interval $[E(i), L(i)]$;
4. The start time of the activity is $S_i = E(i) = L(i) = Rand$;
5. Execute *Earliest* (i);
6. Execute *Latest* (i);
7. $i = i + 1$;
8. Go back to the first step.

3.1.4. Neighborhood generation algorithm

1. Choose one of the project activities and denote it by j;
2. Generate a random number called *Rand* within the interval $[E(j), L(j)]$;
3. If, $S_i < RAND$, $S_i = E(i) = L(i) = RAND$.

Execute *Earliest* (j) on the *RE*, and during the process for all activities, set start times (S_i) and the latest start time to calculated start times in *Earliest* (j). Otherwise:

$$S_i = E(i) = L(i) = RAND.$$

Execute *Earliest* (j) on the *RE*, and during the process for all activities, set start times (S_i) and the latest start time to calculated start times in *Latest* (j).

At the end, to show the readability of the algorithm, the Pseudo-code of the proposed MSA is also plotted and represented in Figure 2.

Step 1 Set the initial parameters ($T0$, α (or ALPHA), *Number of the Markove chains* (Markov Chain), *number of the iterations in each chain (STEP), h, NO* (Neighbors)
Step 2 Best solution (Best-sol)=Initial solution
Current solution (Cur-sol)=Initial solution
Step 3 While $i \leq Nc$
Set $T = T0$
Number of the neighborhood structures (Ns) $= NO$
While $j \leq i$
$N_s = N_s(1 + h * step)$
While $k \leq N_s$
Create a neighborhood solution and evaluate it
Calculate the neighbor probability ($P = e(-Delta/T)$)
If it is accepted, Cur-sol=new solution
If it is better than Best-sol, Best-sol=new solution
End k
$T = \alpha T$
End j
Save best solution of this step of the chain
End i
Step 4 End of the algorithm

Figure 2. The MSA procedure.

3.2. The GA

In order to compare the proposed MSA, we applied one of the best-developed algorithms in the literature called GA. The structure of GA is designed in such a way that the generation population or the chromosomes are produced in the first step. The new generation is determined with four operators: reproduction, crossover, mutation and local search (for better performance). The initial solutions are randomly created and evaluated. Then, all the chromosomes are considered for the crossover. In this phase, the crossover occurs for two pairs of chromosomes with a probability of P_{cr}, and with each crossover, two new chromosomes are created. The fitness functions of the chromosomes are calculated. Before two new chromosomes are selected, the mutation operation occurs for each of them with the probability of P_m. Then, a local search operation is performed. To choose between the child and the parent chromosomes, suppose that the first parent is called P_1, the second parent is called P_2, the first child is called Ch_1 and the second child is called Ch_2. The fitness function of chromosome i is shown by $f(i)$. At first, by considering P_1 and Ch_1 chromosomes, the random number, r, in the interval zero and one will be produced. If equation $(f(P1) + f(CH1)) \times r \geq f(P1)$ is established, the first child will be replaced by the first parent, and the otherwise Ch_1 will leave generation. The same relationship exists for Ch_2 and P_2. The same approach is used for the replacement. The replacement happens and a mutation chromosome is produced. Then, a comparison is done between the fitness of the primary chromosome and mutation chromosome. Moreover, as elitism is considered, the best chromosome is directly copied to the new generation. This is repeated continuously until the stop condition is achieved. In this model, the stop conditions either achieve 50 generations or the same objective function in 5 iterations. The penalty function is as follows:

$$f(I) = \sum_{k=1}^{m} C_k R_k^I. \tag{20}$$

3.3. Random search

To justify the obtained solutions of both meta-heuristic algorithms, a Random Search (RS) policy has been considered to represent the efficiency and intelligence of the solving methodologies. In fact, comparison of the meta-heuristic algorithms with a RS causes a lower bound to be obtained for the problem outputs, which obviously indicates the appropriate performance of the solving methodologies. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be the combined cost function, which must be minimized, and $x \in \mathbb{R}^n$ designate a position or candidate solution in the search space.

4. Computational results and comparisons

This section deals with the parameters of the proposed MSA including initial temperature, decreasing ratio, the number of the Markov chain, step, and neighbors. In GA, the parameters, such as Number of Generations, crossover operation (P_c : P_{cross}), mutation (P_m : P_{mute}), and local improvement operation (P_l : $P_{\text{local\_search}}$) of the proposed GA are used. Each of them is especially important and can affect the performance of the algorithm.

Then, the computational performance of the MSA is investigated on a set of test problems. Since the RIPDCFPP/GPR is a new mathematical model in the area of project scheduling, no standard test problems could be found to check the performance of the proposed MSA procedure. Thus, we are forced to examine the RIP/max test problems that are available in the PSPLIB library [28]. Consequently, due to RIP/max, the test problems do not contain some variables of the current. The use of the recommended procedure is applied in the manner of [9].

Table 1. GA operators and parameters.

Factor	Levels	Type			
Number of generation	4	A-1)15	A-2)30	A-3)45	A-4)60
Popsize	4	B-1)10	B-2)30	B-3)50	B-4)70
P_C	4	C-1)0.55	C-2)0.75	C-3)0.85	C-4)0.95
P_m	4	D-1)0.001	D-2)0.005	D-3)0.025	D-4)0.05
P_l	4	E-1)0.2	E-2)0.4	E-3)0.6	E-4)0.8

Table 2. MSA operators and parameters.

Factor	Levels	Type			
T_0	4	A-1)800	A-2)1200	A-3)1600	A-4)2000
ALPHA	4	B-1)0.2	B-2)0.4	B-3)0.6	B-4)0.9
Markov chain	4	C-1)1	C-2)2	C-3)3	C-4)4
Step	4	D-1)10	D-2)30	D-3)50	D-4)70
Neighbors	4	E-1)10	E-2)30	E-3)50	E-4)70

Table 3. Setting the parameters of the algorithms.

Algorithms	Factor	Low		Medium		Large	
		Level	Size	Level	Size	Level	Size
GA	Number of generation	3	45	4	60	2	30
	Popsize	2	30	2	30	1	10
	P_c	3	0.85	2	0.75	4	0.95
	P_m	3	0.025	4	0.05	1	0.001
	P_L	3	0.6	2	0.6	3	0.6
MSA	T_0	3	1600	3	1600	4	2000
	ALPHA	2	0.4	2	0.4	2	0.4
	Markov	2	2	2	2	4	4
	Step	2	70	2	70	4	70
	Neighbors	4	70	4	70	4	70

Regarding the nature of the problem, most times, LINGO is unable to obtain a global optimal solution for large-scale problems. In this research, the performances of the proposed MSA are compared to the ones of another meta-heuristic algorithm, namely GA. So, to obtain a better performance for comparing the two algorithms, the GA parameters are also tuned. To optimize the parameter values, we utilize the Taguchi method. For more details of this method, one can refer to [29]. These issues consist of three ranges of problems with 10, 20 and 30 non-dummy activities, which are considered to be 270 issues in each range. Among these issues, 90 issues in different sizes are selected. In the proposed model, the range of problems is divided into three sizes: small, medium and large for the algorithms. Therefore, three ranges for five times on 10 random issues have been considered. To do so, we utilized MINITAB15 software. The algorithms are implemented under Windows 7 and Intel (R) Core (TM) 2Duo CPU and RAM 1.96GB based on MATLAB software [30]. Then, the numbers of required levels for each algorithm are expressed in Tables 1 and 2.

The Taguchi outputs are represented by L16 (4**5). To do so, MEAN and SNR graphs are plotted for different sizes of the problem. Regarding maximization of the objective function, optimal levels for each factor are as follows:

1. SNR graph to be maximized;
2. Responses mean graph (MEAN) to be minimized.

The results of the Taguchi parameter settings for two meta-heuristic algorithms on small, medium, and large sizes are illustrated in Table 3.

On the other hand, Relative Percentage Deviation (RPD), which is one of the most famous methods for efficiency measurement in objective problems, is considered. Regarding the 90 selected problems, the best solution for each algorithm is presented. RPD are obtained as follows:

$$\text{RPD}_J = \sum_{i=1}^{r=5} \left| \frac{z_i - z_{\max}}{z_{\max}} \right|, \qquad (21)$$

where z_i is solution i in each algorithm, and $z_{\max}$ is the best solution of the problem in all tests.

According to maximization of the objective function, lower RPD criteria represents the efficiency of the algorithm in the corresponding size. To do so, the RS algorithm is considered for obtaining a lower bound for the problem. The reason for selecting this heuristic algorithm is to obtain a lower bound of the target function for the problem. So, obtained solutions can evaluate the efficiency of the algorithms. The application of this heuristic algorithm is, at first, selecting a population and then considering the average of the population solutions as a minimization of the problem. Thus, to assess whether there is a significant difference for the RPD and RPD of two algorithms, a more applicable statistical analysis, called the T-paired method, with 95% confidence level, has been utilized as follows:

$$\begin{cases} H_0 : D_i \geq 0 \\ H_1 : D_i < 0 \end{cases} ; \qquad i = 1, 2, 3. \qquad (22)$$

With regard to small, medium, and large sizes of the corresponded problem, D_i are defined as follows:

$$D_1 = RPD_{\text{GA},S} - RPD_{\text{PSA},S}, \qquad (23)$$

$$D_2 = RPD_{\text{GA},M} - RPD_{\text{PSA},M}, \qquad (24)$$

$$D_3 = RPD_{\text{GA},L} - RPD_{\text{PSA},L}, \qquad (25)$$

$$D_4 = NPV_{\text{PSA},s} - NPV_{\text{GA},s}, \qquad (26)$$

$$D_5 = NPV_{\text{PSA},M} - NPV_{\text{GA},M}, \qquad (27)$$

$$D_6 = NPV_{\text{PSA},L} - NPV_{\text{GA},L}. \qquad (28)$$

In this way, when $P-\text{value} > \alpha$, the H_0 hypothesis would not be rejected. Table 4 represents the hypothesis tests of Eq. (23)-(28), respectively.

Moreover, in order to demonstrate the performance of proposed MSA, the algorithms are statistically compared based on the properties of their obtained solutions via analysis of variance (ANOVA) tests. The procedure of ANOVA, including the F-test value, P-values, and the results on each metric, are summarized in Table 5 for small size (NPV-S), medium size (NPV-M), and large size (NPV-L).

To clarify the statistical outputs, box-plots are shown in Figure 3. In order to increase the readability of the comparisons, the algorithms are also compared graphically on the whole test problems in Figures 4-6. These figures show a comparison between the proposed MSA, GA, and RS, based on NPV criteria, for all test problems.

Finally, with regard to paired comparison and acceptance criteria for GA and MSA, we perceive that the proposed MSA has a better performance than GA in NPV solutions and RPD criteria, especially in large-size problems.

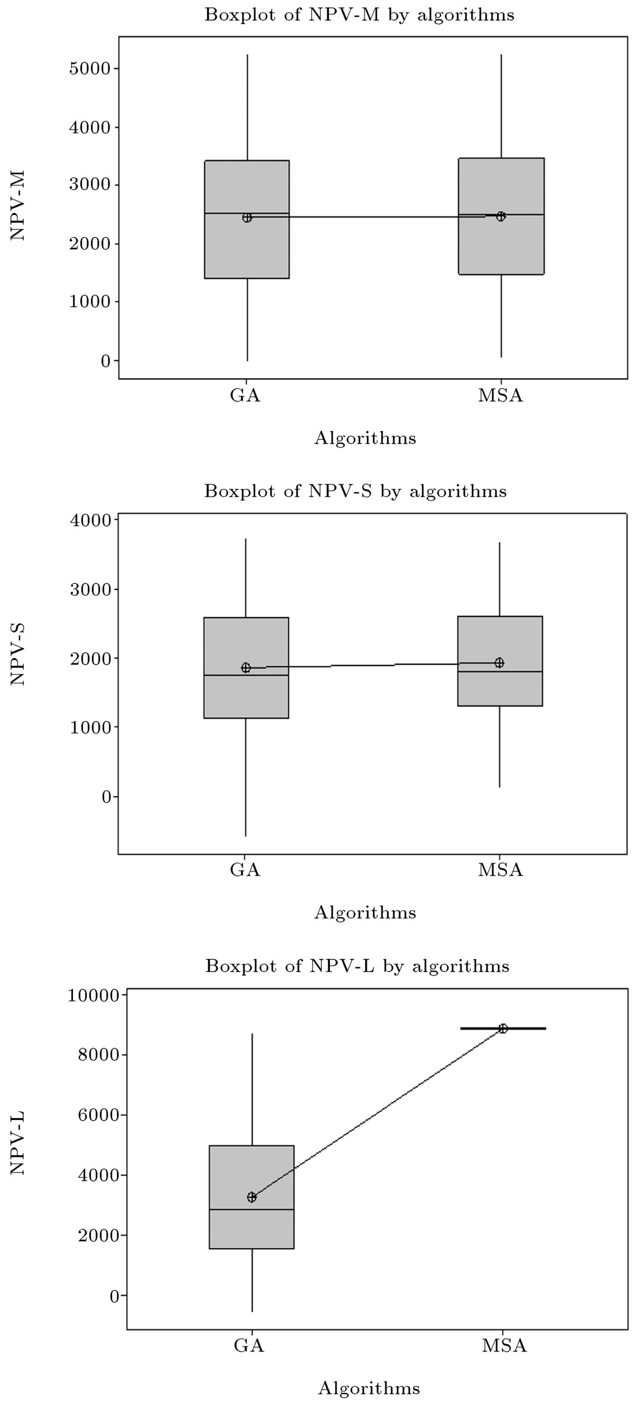

Figure 3. Box-plot of NPV metric for different sizes of the problem.

Table 4. Hypothesis tests.

Criteria	Size	Test statistic	*P*-value	Result
RPD	Small	T-value = 2.16	P-value = 0.983 > 0.05	$D_1 \geq 0$; $RPD_{\text{GA},S} \geq RPD_{\text{PSA},S}$
	Medium	T-value = 1.83	P-value = 0.965 > 0.05	$D_2 \geq 0$; $RPD_{\text{GA},M} \geq RPD_{\text{PSA},M}$
	Large	T-value = 24.28	P-value = 1.000 > 0.05	$D_3 \geq 0$; $RPD_{\text{GA},L} \geq RPD_{\text{PSA},L}$
NPV	Small	T-value = 2.78	P-value = 0.997 > 0.05	$D_4 \geq 0$; $NPV_{\text{GA},S} \leq NPV_{\text{PSA},S}$
	Medium	T-value = 0.91	P-value = 0.818 > 0.05	$D_5 \geq 0$; $NPV_{\text{GA},M} \leq NPV_{\text{PSA},M}$
	Large	T-value = 0.91	P-value = 0.818 > 0.05	$D_6 \geq 0$; $NPV_{\text{GA},L} \leq NPV_{\text{PSA},L}$

Table 5. The results of ANOVA test for different sizes of the problem.

Metrics	Source	DF	SS	MS	F	*P*-value	Test results
NPV-S	Algorithms	1	295119	295119	0.34	0.558	Null hypothesis is not rejected
	Error	178	152459058	856512			
	Total	179	152754177				
NPV-M	Algorithms	1	14968	14968	0.01	0.926	Null hypothesis is not rejected
	Error	178	305821143	1718096			
	Total	179	305836111				
NPV-L	Algorithms	1	1426578994	1426578994	594.90	0.000	Null hypothesis is rejected
	Error	178	426849654	2398032			
	Total	179	1853428648				

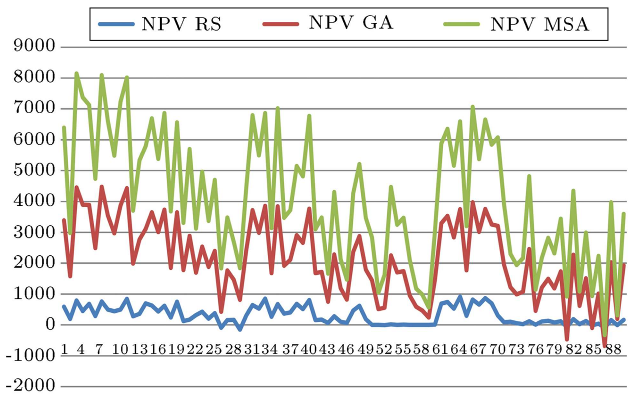

Figure 4. The comparison between GA and MSA based on NPV criteria for small size.

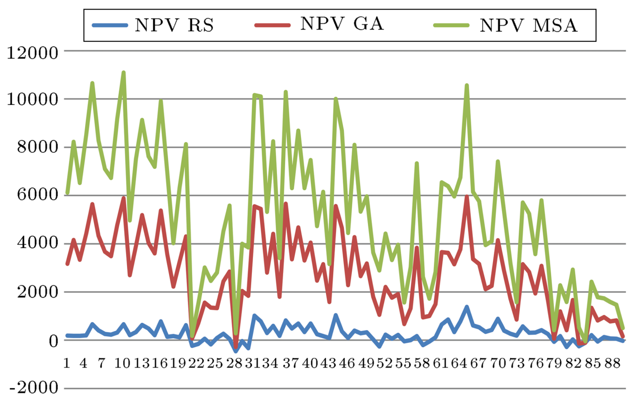

Figure 5. The comparison between GA and MSA based on NPV criteria for medium size.

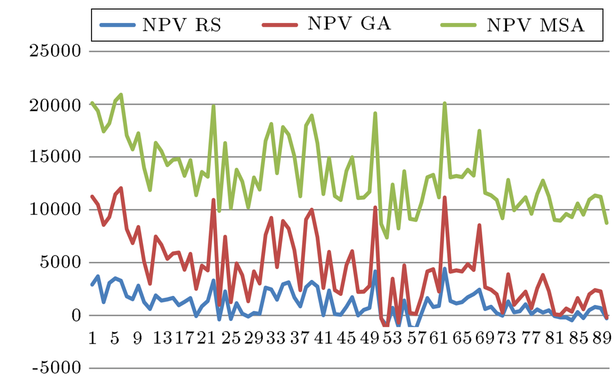

Figure 6. The comparison between GA and MSA based on NPV criteria for large size.

5. Conclusions and future work

In this paper, a new model in RIP, by maximizing NPV, based on payment progress, has been investigated. The following items have been considered in this paper: positive and negative cash flow during the project, money time value, and issue of prerequisite types with minimum and maximum time delay of payment progress. Since the proposed model is NP-hard, two meta-heuristic algorithms, including GA and MSA, were proposed. To do so, the Taguchi parameter setting technique and RPD index were applied. To demonstrate the applicability of the model, various sizes of problem, including the 270th issue with 10, 20 or 30 activities, were carried out. Ultimate results represent the better performance of MSA over GA, in terms of the metrics, especially in large-size problems. For future research, the multi-objective multi-mode RIPDCFPP/GPR in which tardiness is permitted with penalty, is suggested.

Acknowledgments

The authors are indebted to the anonymous referees whose comments greatly improved the first version of the paper.

This research was supported by a research project grant at the Qazvin Branch, Islamic Azad University, Iran (91-121).

References

1. Icmeli, O., Erenguc, S.S. and Zappe, C.J. "Project scheduling problems: a survey", *Int. J. of Oper. Prod. Manage.*, **13**, pp. 80-91 (1993).

2. Tavares, L.V. "A review of the contribution of operation research to project management", *Eur. J. Oper. Res.*, **136**, pp. 1-18 (2002).

3. Ulusoy, G. and Sivrikaya, F. "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows", *Ann. Oper. Res.*, **102**, pp. 237-261 (2001).

4. Herroelen, W., Van Dommelen, P. and Demeulemeester E. "Project network models with discounted cash flows: A guided tour through recent developments", *Eur. J. Oper. Res.*, **100**, pp. 97-121 (1997).

5. Heilmann, R. "A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags", *Eur. J. Oper. Res.*, **144**, pp. 348-365 (2003).

6. Vanhoucke, M., Demeulemeester, E. and Herroelen, W. "Progress payments in project scheduling problems", *Eur. J. Oper. Res.*, **148**, pp. 604-620 (2003).

7. Mohring, R.H. "Minimizing costs of resource requirements in project networks subject to a fix completion time", *Oper. Res.*, **32**, pp. 89-120 (1984).

8. Shadrokh, S. and Kianfar, F. "A genetic algorithm for resource investment project scheduling problem; tardiness permitted with penalty", *Eur. J. Oper. Res.*, **181**(1), pp. 86-101 (2007).

9. Najafi, A.A. and Niaki, S.T.A. "A genetic algorithm for resource investment problem with discounted cash flows", *Appl. Math. Comput.*, **183**, pp. 1057-1070 (2006).

10. Najafi, A.A., Niaki, S.T.A. and Shahsavar, A.A. "A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations", *Comput. Oper. Res.*, **36**(11), pp. 2994-3001 (2009).

11. Bianco, L. and Caramia, M. "An exact algorithm to minimize the makespan in project scheduling with scarce resources and generalized precedence relations", *Eur. J. Oper. Res.*, **219**, pp. 73-85 (2012).

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

13. Rahimi, A., Karimi, H. and Afshar-Nadjafi, B. "Using meta-heuristics for project scheduling under mode identity constraints", *Appl. Soft Comput.*, **13**(4), pp. 2124-2135 (2013).

14. Paraskevopoulos, D.C., Tarantilis, C.D. and Ioannou, G. "Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm", *Expert Syst. Appl.*, **39**, pp. 3983-3994 (2012).

15. He, Z., Liu, R. and Jia, T. "Metaheuristics for multi-mode capital-constrained project payment scheduling", *Eur. J. Oper. Res.*, **223**, pp. 605-613 (2012).

16. Xiao, J., Ao, X. and Tang, T. "Solving software project scheduling problems with ant colony optimization", *Comput. Oper. Res.*, **40**, pp. 33-46 (2013).

17. Koza, J.R. "Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems", Rep. No. STAN-CS-90-1314, Stanford University, CA, USA (1990).

18. Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI (1975).

19. Poli, R., Kennedy, J. and Blackwell, T. "Particle swarm optimization: an overview", *Swarm Intell.*, **1**, pp. 33-57 (2007).

20. Hajipour, V. and Pasandideh, S.H.R. "Proposing an adaptive particle swarm optimization for a novel bi-objective queuing facility location model", *Econ. Comput. Econ. Cybern. Stud. Res.*, **3** (2012).

21. Kirkpatrick, S., Gelatt, C. and Vecchi, M. "Optimization by simulated annealing", *Sci.*, **220**, pp. 671-680 (1983).

22. Atashpaz-Gargari, E. and Lucas, C. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialist competitive", *IEEE Congr. Evol. Comput.*, Singapore (2007).

23. Neumann, K. and Zimmermann, J. "Resource leveling for project with schedule dependent time windows", *Technical Report 508*, Institut für Wirtschaftstheorie und Operations Research, Universitat Karlsruhe (1997).

24. König, M. and Beißert, U. "Construction scheduling optimization by simulated annealing", *26th Int. Symp. Autom. Rob. Constr.* (2009).

25. Bouleimen, K. and Lecocq, H. "A new efficient simulated annealing algorithm for the resource-constrained

project scheduling problem and its multiple mode version", *Eur. J. Oper. Res.*, **149**, pp. 268-281 (2003).

26. Shahsavar, M., Niaki, S.T.A. and Najafi, A.A. "An efficient genetic algorithm to maximize net present value of project payments under inflation and bonus-penalty policy in resource investment problem", *Adv. Eng. Software*, **41**, pp. 1023-1030 (2010).

27. Demeulemeester, L. and Herrolen, W., *Project Scheduling: A Research Handbook*, Kluwer Academic Publishers, Boston, London (2002).

28. Available from <http://www.bwl.uni-kiel.de/prod /psplib/ index.html >.

29. Mousavi, M., Hajipour, V., Niaki, S.T.A. and Aalikar, N. "Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation approaches: two calibrated meta-heuristic algorithms", *Appl. Math. Modell.*, **37**, pp. 2241-2256 (2013).

30. MATLAB Version 7.10.0.499 (R2010a). The Math Works, Inc. Protected by U.S. and international patents (2010).

Biographies

Behrouz Afshar-Nadjafi received his PhD degree in Industrial Engineering from Sharif University of Technology, Tehran, Iran, in 2008, and is currently Assistant Professor and Graduate Programs Manager of Industrial Engineering at the Islamic Azad University, Qazvin Branch, in Iran. His research interests include project scheduling, and inventory and production planning, and, more specifically, modeling and solution methods, including exact procedures, meta-heuristic algorithms and artificial intelligence techniques regarding discrete optimization problems in reality.

Abozar Parsanejad received BS and MS degrees, both in Industrial Engineering from the Islamic Azad University, Qazvin Branch, Iran, in 2007 and 2010, respectively. His research interests include project scheduling and logistics, and, more specifically, fuzzy multi-criteria decision making, graph methods, and gray systems.

Vahid Hajipour holds BS and MS degrees in Industrial Engineering from Islamic Azad University (IAU), Iran, and is currently a doctoral student and Visiting Professor in the Department of Industrial Engineering at Bu-Ali Sina University, Hamadan, Iran. He is a member of the Iranian National Elite Foundation and Young Researchers Club, and recipient of the 2011 Distinguished Young Researcher Award at the IAU of Qazvin, Iran. His research interests include Queuing Theory, Facility Location, Multi-Objective Optimization, Computational Intelligence, and Fuzzy Sets Theory. Vahid Hajipour has published papers in a number of journals, such as Applied Mathematical Modeling, Journal of Intelligent Manufacturing, Applied Soft Computing, Measurement, International Journal of Advanced Manufacturing Technology, Economic Computation and Economic Cybernetic Studies and Research, among others.

Arash Nobari obtained BS and MS degrees in Industrial Engineering from Khaje Nasir Toosi University of Technology (KNTU), Tehran, Iran, in 2010 and 2012, respectively, and is currently studying for a PhD degree in Industrial Engineering in Bu-Ali Sina University, Hamadan, Iran. His research interests include logistic and supply chain management, mathematical modeling, and optimization algorithms.