International Journal of Industrial Engineering & Production Research September 2021 Vol. 32, No. 3: 1-16
DOI: 10.22068/ijiepr.32.3.1

RESEARCH PAPER

Multi-Objective Fuzzy Modeling of Project Scheduling with Limitations of Multi-Skilled Resources Able to Change Skill Levels and Interrupt Activities

Mojtaba Salehi*[1] & Efat Jabarpour[2]

Received 21 September 2019; Revised 6 November 2020; Accepted 3 August 2021;

ABSTRACT

Project scheduling is one of the most important and applicable concepts of project management. Many project-oriented companies and organizations apply variable cost reduction strategies in project implementation. Considering the current business environments, in addition to lowering their costs, many companies seek to prevent project delays. This paper presents a multi-objective fuzzy mathematical model for the problem of project scheduling with the limitation of multi-skilled resources able to change skills levels, optimizing project scheduling policy and skills recruitment. Given the multi objectivity of the model, the goal programming approach was used, and an equivalent single-objective model was obtained. Since the multi-skilled project scheduling is among the NP-Hard problems and the proposed problem is its extended state, so it is also an NP-Hard problem. Therefore, NSGA II and MOCS meta-heuristic algorithms were used to solve the large-sized model proposed using MATLAB software. The results show that the multi-objective genetic algorithm performs better than the multi-objective Cuckoo Search in the criteria of goal solution distance, spacing, and maximum performance enhancement.

KEYWORDS: *Project scheduling; Multi skilled resources; Goal programming; Multi-objective genetic algorithm; Multi-objective cuckoo search.*

1. Introduction

The use of project management methods dates back to about fifty years ago; during this time, great effort was made on project management and project scheduling model development[1]. Project scheduling is an important task in the project management context, playing a vital role in today's enterprise management. In practice, project managers deal with numerous internal/external constraints, making project objectives too difficult to achieve. Among these constraints, scarce resources and precedence relations between activities make project scheduling a challenging task [2]. In the early 21st century, a new version of the problem has been introduced, Multi-skilled Project Scheduling Problem (MSPSP), in the literature that involves simultaneously solving two project scheduling problems and allocating multi-skilled resources. In the standard case, this problem is defined as the determination of feasible scheduling of activities with regard to the prerequisite relationships and the limitations of the multi-skilled resources, mostly human resources.

The Resource Investment Problem (RIP) is a type of RCPSP in which limitations of renewable resources are considered decision variables. In these problems, resource access levels (limitations of renewable resources) are decision variables to minimize the cost of supplying resources. This paper deals with modeling the project scheduling problem by investing in multi-skilled resources. Many studies have been performed on project scheduling and resource investment; however, very little research has

* Corresponding author: *Mojtaba Salehi*
m_salehi61@yahoo.com

1. *Department of Industrial Engineering, payamenoor university, Tehran, Iran.*
2. *Department of Industrial Management, Tehran University, Tehran, Iran.*

been done on multi-skilled resource investment in project scheduling.

Following is a review of the related literature. For this purpose, the literature is divided into three parts. The first part deals with project scheduling with multi-skilled resources. The second part deals with project scheduling with resource investment, and the third part deals with project scheduling with multi-skilled resource investment.

2. Literature Review

2.1. Multi-skilled project scheduling problems

The multi-skilled project scheduling problem was first presented by Neron et al. [3]. The models they introduced are based on the assumption that the skills of the skilled individuals are the same in various working areas. Later, MSPSP was developed as the classic problem with hierarchical skill levels to minimize project execution time by assuming that the activities cannot be interrupted. They provided the lower limit for this problem.

Belenguez et al. suggested several solutions to MSPSP, including lower limit, branch and bound, and meta-heuristic tabu search and genetic methods [4,5]. Wallis et al. [6] presented a solution to the project scheduling problem with a multi-skilled workforce. They categorized the workforce into senior, standard, and novice levels to minimize the project execution time. They used a hybrid genetic algorithm to solve the problem.

Arashpour et al. [7] optimized process integration and utilization of multi-skilled resources in website development. Further, Correia and Dogama [8] proposed a branch and bound approach for multi-skill project scheduling problems. Javanmard et al. [9] presented the project scheduling problem with multi-skilled resource investment to obtain an optimal policy of simultaneous project scheduling and skills recruitment. Therefore, the objective function considered in their single-objective model was to minimize the cost of recruiting different levels of skills. They used genetic and particle swarm optimization to solve the model.

Almeida et al. [10] presented a heuristic approach based on prioritization rules for the project scheduling problem with multi-skilled resource limitations. They used the parallel scheduling method to generate the schedule; they also used weighted resources to classify activities.

Maenhot Wanhukeh [11] explored the problem of integrating project scheduling and human resources, thus scheduling project activities and simultaneously performing the personnel planning for a project. In this method, the project schedule leads to personnel planning intending to minimize costs. In this study, different types of personnel resources were used for scheduling, and a heuristic method was provided for solving the problem.

2.2. Problems of project scheduling with resource investment

The problem of minimizing resource investment was first introduced by Mohring [12]. Demolmister [13] presented a detailed algorithm for solving the problem based on the branch and bound method; then, he compared the results with those of the algorithm proposed by Mohring [12]. Drexel and Kims[14] proposed two low-level procedures for solving the problem based on column generation and Lagrangian relaxation that utilized assumptions similar to those offered by Demolmister[13] to model the problem. They also used the results presented by Mohring[12] to evaluate the performance of their proposed procedure.

For the first time, Shadrokh and Kianfar [15] considered the possibility of tardiness in the delivery of the project in their proposed model. In this model, the relationships between end-to-start activities were of zero time interval, and all resources were renewable. Moreover, this model activity scheduling was single-modal, and the genetic algorithm was used to solve the problem. Due to the lack of similar research, a number of sample problems were designed and solved.

Afshar Najafi [16] used the Simulated Annealing algorithm to minimize the cost of accessing resources in multi-modal activities. One of the important assumptions in this model was the date of hiring and releasing resources. Afshar Najafi and Arani [17] presented an article in which the project earliness was considered along with the penalties. In this study, the project earliness could lead to prizes. Other important assumptions presented in this paper included the inflation and discount rates for the project.

Najafi and Salimi [18] presented a multi-objective mathematical model of project scheduling in resource investment problems. The goal was to minimize the cost and time of completing the project, which also existed in the interruptions. To solve the problem, multi-objective meta-heuristic non-dominated sorting genetic algorithm and non-dominated sorting genetic algorithm were provided.

2.3. Problem of project scheduling with multi-skilled resource investment

There is only one study on this research area, namely project scheduling with multi-skilled resource investment, i.e., Javanmard et al. [9]. They provided a single-objective mathematical model to minimize the costs of using skill levels at any given period of multi-skilled resources under certain conditions.

According to the literature, the activities are merely operational, the objective functions are presented as single-objective, and all parameters are under certain conditions and non-renewable constraints.

Accordingly, the proposed paper presents a multi-objective mathematical model capable of performing activities in different states and under fuzzy conditions. Objective functions include minimizing the cost of using multi-skilled resources and project completion time. Multi-objective genetic algorithms and multi-objective cuckoo algorithms are also used to solve the proposed problem.

3. Problem Definition and Modeling

In this article, there are several skills needed to perform each activity. The goal is to optimize resource availability (skill levels) and find the best schedule to minimize investment in resources.

- Primary and secondary activities are virtual activities
- All parameters are certain
- All required resources are available at project start
- Each skill may be used on one of the levels
- Higher skill levels lead to faster operation at a higher cost
- Each activity may require one or more skills
- All the skills required are available at the start of each activity

3.1. Parameters and variables of mathematical model

P_i: Prerequisite set of activities

r_{imkl}: The number of workforces required to operate the activity i using the skill k at the level l in the operational mode m (fuzzy)

Cr_{klt}: The cost of using skill k at the level l in time t

S_i : The set of skills needed to perform the activity i

T: Project planning horizon (project completion time)

W_{ik} : Workload i related to skill k

$\tilde{v}_{imkl}$: Speed of implementation of activity i using the level I of skill k in the operational state m (in fuzzy state)

rf_{imf} : Number of units required from the non-renewable resource f to perform the activity i

RF_f : Maximum units available from non-renewable source f

R_{kl}^t : Available number of skill k at level l at time t

s_{ik} : Time of starting skill k to perform the activity i

c_{ik} : Time of ending skill k to perform the activity i

s_i: Time of starting the activity i

c_i : Time of ending the activity i

x_{imk}^{lt} : If the activity i starts with the skill k at level l in the state m at time t

y_{imk}^{lt}: If the activity i is being operated with the skill k at level l in the state m at time interval $[t-1, t]$

3.2. Modeling:

$$Min\ Z_1 = \sum_t \sum_k \sum_l Cr_{klt} \times R_{kl}^t \qquad (1)$$

$$Min\ Z_2 = c_n \qquad (2)$$

$$s_{ik} = \sum_t \sum_l \sum_m t \times x_{imk}^{lt} \qquad \forall\ k \in S_i\ , \forall\ i \qquad (3)$$

$$\sum_t \sum_l \sum_m x_{imk}^{lt} = 1 \qquad \forall\ k \in S_i\ , \forall\ i \qquad (4)$$

$$s_i = s_{ik} \qquad \forall\ k \in S_i\ , \forall\ i \qquad (5)$$

$$s_{ik} = s_{ik'} \qquad \forall\, k \neq k' \in S_i \;, \forall\, i \tag{6}$$

$$c_i \leq s_j \qquad \forall\, i, j \in P \tag{7}$$

$$\sum_t \sum_l \sum_m y_{imk}^{lt} \times \tilde{v}_{imkl} \geq W_{ik} \qquad \forall\, k \in S_i \;, \forall\, i \tag{8}$$

$$\sum_i \sum_m \sum_k \sum_l rf_{i,m,f} \times y_{imk}^{lt} \leq RF_f \qquad \forall\, f \;, \forall\, t \tag{9}$$

$$\sum_m y_{imk}^{lt} \times t \leq c_{ik} \qquad \forall\, k \in S_i \;, \forall\, i, t, l \tag{10}$$

$$s_{ik} - M \times (1 - \sum_l \sum_m y_{imk}^{lt}) \leq \sum_l \sum_m y_{imk}^{lt} \times t \qquad \forall\, k \in S_i \;, \forall\, i \;, \forall\, t \tag{11}$$

$$\sum_l \sum_m t \times x_{imk}^{lt} - M \times (1 - \sum_l \sum_m y_{imk}^{lt}) \leq \sum_l \sum_m y_{imk}^{lt} \times t \qquad \forall\, k \in S_i \,, \forall\, i \;, \forall\, t \tag{12}$$

$$c_{ik} = s_{ik} + \sum_t \sum_l \sum_m y_{imk}^{lt} \qquad \forall\, k \in S_i \;, \forall\, i \tag{13}$$

$$c_i \geq c_{ik} \qquad \forall\, k \in S_i \,, \forall\, i \tag{14}$$

$$\sum_l \sum_m y_{imk}^{lt} \leq 1 \qquad \forall\, k \in S_i \;, \forall\, i.t \tag{15}$$

$$\sum_i \sum_m \tilde{r}_{imkl} \times y_{imk}^{lt} \leq R_{kl}^t \qquad \forall\, k \in S_i \;, \forall\, l, t \tag{16}$$

$$c_n \leq T \tag{17}$$

$$x_{imkl}^t . y_{imk}^{lt} \in \{0.1\} \qquad \forall\, i.k.l.t \tag{18}$$

$$R_{kl}^t . s_{ik} . c_{ik} . s_i . c_i \geq 0 \qquad \forall\, i.k.l.t \tag{19}$$

(1) The first objective function of the model related to the total cost of the skills available on the time horizon.
(2) The second objective function of the model which minimizes project completion time.
(3) The first constraint calculates the starting time of skill k for activity i.
(4) The second constraint ensures that each skill k for activity i must start at a specified level and a specified time.
(5) The third constraint specifies the start time of each activity.
(6) The fourth constraint guarantees the simultaneous start of all the skills needed to perform the activity i.
(7) The fifth constraint indicates the prerequisite relationships of activities.
(8) The sixth constraint of the model that indicates the rate at which the activity is performed at each level and skill, and is determined by the workload of each activity, the duration and runtime of each activity.
(9) This relationship indicates that the amount of non-renewable resources allocated to activities should not exceed the designated access level at any time.

(10) - (11) These two constraints guarantee the relationship between the starting and ending times of skill k for activity i.
(12) This relationship shows that any activity i can be performed by a level of skill k only if that skill is assigned to activity i.
(13) Calculates the time for completion of activity i with the skill k.
(14) This constraint shows that the time to complete activity i is not less than the time to complete that activity with different skills.
(15) This relationship confirms that among the skills needed in each time period, at most one of its skill levels are used for each activity
(16) This constraint shows maximum access to different levels of each skill.
(17) This relationship ensures that the project completion time does not exceed the set date.
(18) - (19) represent the problem variables.

3.3. Uncertain parameters control

The well-known weighted average method introduced by Rimack and Rimank [19] was used in fuzzy number sorting, to deal with the values of the duration of each activity on the left of constraints (3-8) and (3-16) for the non-fuzzy and definite parameters. Therefore the corresponding definitive auxiliary constraints are as follows.

$$\sum_t \sum_l \sum_m y_{imk}^{lt} \times (w_1 . v_{imkl,\beta}^{o} + w_2 . v_{imkl,\beta}^{m} + w_3 . v_{imkl,\beta}^{p}) \geq W_{ik} \qquad \forall k \ , \forall i \qquad (20)$$

$$\sum_i \sum_m (w_1 . r_{imkl,\beta}^{o} + w_2 . r_{imkl,\beta}^{m} + w_3 . r_{imkl,\beta}^{p}) \times y_{imk}^{lt} \leq R_{kl}^{t} \qquad \forall k \ , \forall l, t \qquad (21)$$

Given: $w_1 + w_2 + w_3 = 1$, where w1, w2 and w3 are the weights of pessimistic, probabilistic, and optimistic values for the fuzzy parameters, respectively. Based on the concept of the most probable values proposed by Lai and Huang [20], the values of these parameters are set to $w_1 = w_3 = {}^1/_6$ and $w_2 = {}^4/_6$.

4. Solution Methods

In this section, first, the definite method of solving the proposed mathematical model is presented by the goal programming. On the other hand, given the problem is NP-Hard, the NSGA II and MOCS algorithms are used to solve larger problems.

4.1. Goal programming

Goal programming is one of the multi-objective decision-making techniques and a very efficient method of multi-objective decision-making. It is a technique that offers a different approach to solving a variety of scheduling problems that have multiple and conflicting objectives. The mathematical model with linear ideal programming is presented below.

$$Min\ Obj = d_1^+ + d_2^+ \qquad (22)$$

$$\sum_t \sum_k \sum_l Cr_{klt} \times R_{kl}^t + d_1^- - d_1^+ = b_1 \qquad (23)$$

$$C_n + d_2^- - d_2^+ = b_2 \qquad (24)$$

$$d_1^+ \leq M * y_1 \qquad (25)$$

$$d_1^- \leq M * (1 - y_1) \qquad (26)$$

$$d_2^+ \leq M * y_2 \qquad (27)$$

$$d_2^- \leq M * (1 - y_2) \qquad (28)$$

$$y_1, y_2 \in \{0,1\} \ ; d_1^+, d_1^-, d_2^+, d_2^- \geq 0 \qquad (29)$$

Other constraints are the same as before.

The optimal implementation of the model is made when the deviations are at their minimum, i.e. zero. It should be noted that in order to obtain the right values of the objective functions (values of b_1 and b_2), first the model is executed with the first objective function and set the value of the objective function as b_1, and in the next phase, the model is executed by the second objective function and set the obtained value as b_2. Then the model is executed in the goal programming format and GAMS presents the optimal global solution.

4.2. Non-dominated sorting genetic algorithm II

This algorithm, like the genetic algorithm, begins with a randomly generated primitive population. In the next step, the generated population is evaluated from the viewpoint of the defined objective functions (suppose we have two minimization goal functions). After dividing the population into different categories using the nondominated sorting process, we calculate the control parameter called the crowding distance. This parameter is calculated for each of two members in each group, and represents a measure of the proximity of the target member to the other members in of that group. The large amount of this parameter will lead to a divergence and a wider range of population members. On the other hand, in this algorithm, among the answers of each generation P_t, some of them are selected using the binary tournament selection method. In the binary selection method, two random responses are selected from the population, and then a comparison is made between the two answers so the best one is eventually selected. The selection criteria in NSGA II are primarily the response rank and, secondly, the crowding distance which is related to the answer. The lowest response rank and the highest crowding distance are preferred. By repeating the binary selection on the population of each generation, a set of individuals of that generation is selected to participate in the combination and mutation. Combination function is performed on a part of the selected individuals and the mutation function is carried out on the rest. As the result, the population Q_t is made up of children and mutated individuals. Subsequently, this population is merged with the main population. The members of the newly formed population R_t are sorted based in their rank in ascending order. Members of the same ranked, are sorted based on crowding distance in a descending order. At present, population members are sorted primarily based on their rank and secondly based on crowding distance. Equal to the number of people in the main population, P_{t+1} members are selected from the top of the sorted list and the rest of the members are discarded. Selected members form the next generation population, and the cycle in this section is often called the Pareto Front. None of the answers in the Pareto front are superior to each other and, depending on the circumstances, all of them can be considered as an optimal decision.

4.2.1. Primary chromosome design

The proposed chromosome consists of three parts. The first part defines the operational modes of each activity. The second specifies the skill levels required for each activity, so that during the execution of each activity the level of each skill could vary. The third part of the chromosome is related to the timing of the start of activities and the time durations of their operations.

4.2.1.1. First part of the primary chromosome

The mode of each activity is obtained by designing the first part of the chromosome. In the first part of the chromosome, each element corresponds to one activity. A number of between zero and one is randomly generated in each element and this applies to all elements. Then, according to the number of modes executed for each problem, the executable mode of each activity is determined, such that the number 1 is divided by the number of modes and number intervals are generated with the same number as the modes. First interval is related to the mode 1 and the second to the mode 2 and so on. The first part of the chromosome is shown below.

	Activity1	Activity2	…	Activity i	…	Activity n
Mode(M)	[0,1]	[0,1]	…	[0,1]	…	[0,1]

Fig. 1. First part of the primary chromosome

4.2.1.2. Second part of the primary chromosome

After generating the first part of the chromosome and determining the mode of each activity, the speed of each activity, the number of forces required to execute each activity are obtained using the skill k at the level l and the number of non-renewable resources.

In the design of the second part of the chromosome, the level(s) of each skill during the execution of each activity is determined; in such a way that in each activity, considering the workload of the skills and the speed of performing each skill level (specified by the determination of the mode), the duration of each activity is determined by each skill (d_{ik}). In other words, to produce the second part of the chromosome and to determine the matrix dimensions of each activity, the maximum duration of each skill of the activity $di=(Maxk(d_{ik}))$;$\forall i$ and $d_{ik}=Maxl(Wik/vikl)$ is used.

	Time period 1		Time period 2		Time period 3		Time period4	
	Level1	Level 2	Level1	Level 2	Level1	Level 2	Level1	Level 2
Skill 1	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
Skill 2	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]

Fig. 2. Second part of the primary chromosome

4.2.1.3. Third part of the primary chromosome

The third part of the chromosome is also a matrix with the size of the number of activities in the programming horizon, consisting of random numbers between zero and one to determine the project execution time. This section determines the time periods in which each activity is running.

To do this, in each activity, considering the earliest starting time of the activity, constraints of non-renewable resources, and the maximum time required for completion of the prerequisites of the relevant activity, the feasible time duration (Sd) is determined. Then, from among the elements determined from the set duration (Sd) until the latest ending time (LF) as the maximum duration of each activity (di), the maximum random value and element is determined. Considering the determined elements (time periods), the execution periods of each activity are determined. Finally, the chromosome produces a justified timing in terms of prerequisite relationships.

	Time period 1	Time period 2	…	Time period DD-1	Time period DD
activity 1	[0,1]	[0,1]	…	[0,1]	[0,1]
…	…	…	…	…	…
Activity n	[0,1]	[0,1]	…	[0,1]	[0,1]

Fig. 3. Third part of the primary chromosome

4.2.2. Crossover operator

Based on the cross-permutations in the chromosome of the parents, the crossover operator creates a new sequence in the offspring's chromosome. The two-point method is used here.

Parent1	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Parent2	m'_1	m'_2	m'_3	m'_4	m'_5	m'_6	m'_7	m'_8

child 1	m'_1	m'_2	m_3	m_4	m_5	m'_6	m'_7	m'_8

child 2	m_1	m_2	m'_3	m'_4	m'_5	m_6	m_7	m_8

Fig. 4. Crossover on first part of the chromosome

Parent1	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
Parent2	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8
child 1	s'_1	s'_2	s_3	s_4	s_5	s'_6	s'_7	s'_8
child 2	s_1	s_2	s'_3	s'_4	s'_5	s_6	s_7	s_8

Fig.5. Crossover on third part of the chromosome

4.2.3. Mutation operator

Mutation operators are used to avoid being stuck in local search loops. They present a new combination of genes in the parent chromosome. With the mutation of each chromosome, the sequence of genes also changes. In this method, two integers (genes) are selected randomly assuming that one of these random numbers must strictly be greater than the other. Then the place value selected with the first random number is deleted, and all values between two random numbers move back one place. Finally, the value of the first place is exchanged with the value of the second place selected by the random numbers. The neighborhood structure of deleting and moving is shown below.

Parent1	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
child1	m_1	m_3	m_4	m_4	m_5	m_2	m_7	m_8

Fig. 6. Mutation on first part of the chromosome

Parent1	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
child1	s_1	s_3	s_4	s_5	s_6	s_3	s_7	s_8

Fig. 7. Mutation on third part of the chromosome

4.3. Cuckoo search optimization algorithm

The cuckoo algorithm is based on the life of a bird named cuckoo proposed by Young and Deb [21] and developed by Rajabiyun [22] Cuckoo optimization algorithm is one of the successful implementations of natural processes. The specific lifestyle, egg laying method and unique growth process of cuckoo bird, along with its reproduction form the basis of this optimization algorithm. A prominent feature of this algorithm is simulation of the concept of survival, migration to find food sources, and choosing the optimal environment for living. Population of cuckoo optimization algorithm consists of mature cuckoos and cuckoo eggs.

As with other evolutionary algorithms, the cuckoo optimization algorithm starts with producing an initial population of cuckoo. Initial populations of cuckoos lay eggs in the nest of host birds. The eggs most closely resembling those of the host bird will have the opportunity to grow into mature cuckoo. Other eggs are identified by the host and subsequently killed.

Cuckoos evaluate the living environment for survival. The environment in which the largest number of cuckoo eggs grows into maturity will produce the highest profit. In other words, the cuckoo lays eggs in an environment where the survival rate of the eggs is maximal. Eggs that survive to maturity form communities of cuckoos in their living environment. Each cuckoo community lives in a specific habitat of the environment.

The best habitat, which has the greatest amount of food resources and chance of survival for the cuckoo, is selected as the migration destination of other communities. Immigrant communities reside near the best habitat. Depending on the number of eggs per cuckoo and the distance to the best habitat, an Egg Laying Area is determined for each cuckoo. Then, each cuckoo will randomly lay eggs in the nests within this area. This will continue until the best area with the highest profit is identified (greatest food sources and highest chance of survival). Most cuckoos converge around this point. In the following, the steps of the cuckoo optimization algorithm will be explained.

In recent years many types of CS have been developed. Multi-objective cuckoo search algorithm (MOCS) was applied to solve multi-objective optimization problems [23] This paper uses a Pareto approach to develop a multi-objective version of the cuckoo algorithm and provides a multi-objective version of the cuckoo algorithm using the operators used in the multi-objective genetic algorithm. In this algorithm, the initial population of cuckoos (cavities or nests) is first generated. The answers are then sorted and the best cuckoo is identified Used to create the next generation. Then in the next stages, each cuckoo tries to reach the best cuckoo with two movements of excellence and randomness. After determining the movement and its amount, the answers are rearranged and this process continues until the optimal answer is reached.. Each solution of the algorithm is a three-part solution similar to the process of chromosome formation process. So, no further description of the solution generation process is provided for this algorithm.

5. Indices for Comparison of Multi-Objective Meta-Heuristic Algorithms

Suppose the model in Eq. (30) consists of three objectives solved by the two algorithms mentioned above and N efficient solutions are obtained by the model. The following indices are introduced to investigate which algorithm is more functional than the others. Suppose the set of efficient solutions as follows:

$$(f_1^1, f_2^1), \dots, (f_1^n, f_2^n) \tag{30}$$

5.1. Maximum spread index (MSI)

This criterion measures the expansion of the space for efficient responses. The more efficient the answers are in a wider space, the larger is the index so the higher values of this index are intended [24] suppose:

f_j^{max} : The maximum value of the objective function for the purpose j among efficient responses

f_j^{min} : The minimum value of the objective function for the purpose j among efficient responses

This index is indicated as D and is calculated using equation (31).

$$D = \sqrt{\sum_{j=1}^{k} (f_j^{max} - f_j^{min})^2} \tag{31}$$

5.2. Mean ideal solution distance (MID)

In this method, an answer is considered as an ideal point and the distance of the Pareto set answers from the ideal point are calculated Euclidean. The ideal point is calculated by calculating each objective function separately without considering the constraints. Obviously the higher index is better.

$$MID = \frac{\sum_{i=1}^{n} ci}{n} \tag{32}$$

$$Ci = \sqrt{\sum_{i=1}^{k} (f_{1i} - f_1^*)^2 + (f_{2i} - f_2^*)^2 + \cdots + (f_{mi} - f_m^*)^2} \tag{33}$$

In this respect Ci represents the Euclidean distance from the ideal point corresponding to the objective function[25].

5.3. Metric distance/ spacing (SM)

The spacing index is used to calculate the relative spacing of consecutive answers. [26] When the solutions are placed uniformly close to each other, SM will diminish, so, the smaller the index, the better the performance of the algorithm, and it is calculated from Eqs. (34) and (35) In this respect $\bar{d}$ is mean of all d_i that are distance between consecutive objective functions.

$$d_i = \min_{j=1,..,n\ j\neq i} \left(\sum_{k=1}^{3} \left| f_k^i - f_k^j \right| \right), \quad \forall\, i = 1, \dots, n \tag{34}$$

$$SM = \frac{\sum_{i=1}^{n-1}\left|\bar{d} - d_i\right|}{(n-1)\bar{d}} \qquad (35)$$

5.4. The number of effective responses or pareto index (NPS)

This index indicates the number of effective responses that can be extracted using the model. Obviously, higher values for this index are preferred.

6. Computational Results

6.1. Generating sample problems

Sample problems are generated randomly for the evaluation of meta-heuristic algorithms. For each problem combination, 20 problems are generated according to the table below. Sample problems are generated by RanGen 1 presented by Demolemister (2003).

The following table shows the method of setting the input parameters of the sample problems. Using this software, several sample problems were constructed based on the number of activities and also the number of non-renewable resources and skills, so that three states were considered for the number of activities, and three others for the number of non-renewable resources and skills. The states considered for the number of activities are: 10 activities, 20 activities, and 30 activities; also, the number of states for the non-renewable resources is 1 to 3 resources and the number of skills is between 1 and 3. The number of states of each activity is 2 and 3. The last column of the table (OS), which is one of the entries in RanGen 1, represents the percentage of existing prerequisites among all possible prerequisites. There are two execution states and two network complexity modes for each size listed in the table below. It should be noted that the rest of the parameters are generated randomly and based on a (random) coefficient of random numbers obtained from RanGen 1.

Tab. 1. Information about sample problems generated by RanGen 1

Size of problem	number of activities	number of states of each activity	number of skills	number of non-renewable resources	OS
small	10	2&3	{1-2}-{1-3}	1	0.25 & 0.75
medium	20	2&3	{1-3}-{2-4}	2	0.25 & 0.75
big	30	2&3	{2-4}-{3-5}	3	0.25 & 0.75

6.2. Setting the parameter of multi-objective genetic algorithm and multi-objective cuckoo algorithm

Obviously, the performance of an algorithm depends on its parameters So that different parameters may produce completely different answers with different qualities. Before solving the problem and analyzing the results, the initial parameters of the NSGA II and MOCS algorithms were adjusted by Taguchi method. In Taguchi method, orthogonal arrays are used to study a large number of decision variables with a small number of experiments. Multiple arrays organize parameters whose values must change and affect the process. This method, first identify the appropriate factors and then select the levels of each factor and then determine the appropriate test design for these control factors. Table 2 and table 3 show the proposed levels of the parameters of these algorithms and the optimal value of each parameter obtained by Taguchi method.

Tab. 2. Value of adjusted parameters (optimized) for NSGA II

level	population	Maximum number of repetitions	(Combination rate)pc	(Mutation rate) pm
1	60	40	0.7	0.1
2	80	60	0.8	0.15
3	100	80	0.9	0.20
optimum	60	60	0.80	0.10

Therefore, the optimal levels are obtained for the parameters of the multi-objective genetic algorithm. By placing these levels in the parameters of the algorithm, a higher-performance genetic algorithm searches for the optimal solution.

Tab. 3. Value of adjusted parameters (optimized) for MOCS

level	population (cuckoo)	Maximum number of repetitions	Steps of cuckoo (α)	movement area (β)
1	20	120	0.15	2.5
2	25	150	0.1	2
3	30	180	0.05	1.5
opt	30	180	0.15	1.5

6.3. Results of comparison of goal programming and meta-heuristic algorithms

In this section, in order to determine the coding accuracy, the small size sample problem presented in Table (1) has been considered and the problem has been solved by mentioned algorithms to determine the difference between the objectives functions of the meta- Heuristics algorithms and Goal programing method. Results are presented below. These are the results of four problem samples.

Tab. 4. Comparison of NSGA II & goal programing

Problem	Goal programing			NSGA II			error	
no	Obj1	Obj2	time	Obj1	Obj2	time	Obj1	Obj2
1	1258	39	5	1297	40	10.7	3.1%	2.5%
2	16910	142	81	17452	151	16.3	3.2%	6.3%
3	230573	228	197	239108	239	18.2	3.7%	4.8%
4	776418	453	262	809316	475	25.4	4.2%	5.1%

Tab. 5. Comparison of *MOCS* & goal programing

Problem	Goal programing			*MOCS*			error	
no	Obj1	Obj2	time	Obj1	Obj2	time	Obj1	Obj2
1	1258	39	5	1297	41	13.1	3.1%	5.1%
2	16910	142	81	17519	149	19.9	3.6%	4.9%
3	230573	228	197	240260	242	26.5	4.2%	6.1%
4	776418	453	262	821094	481	32.02	5.6%	6.2%

The comparison shows that the algorithms show a good performance.

6.4. Comparing algorithms performance

The results of solving the problems by multi-objective cuckoo and non-dominated sorting algorithms are given in the following tables.

Tab. 6. Multi-Objective meta- heuristics algorithms indicators in solving larger-size problems in MOCS

Problem no	NPS	MID	D	S
1	8	48.2	250.7	4
2	4	35.8	280.1	5
3	5	52.5	180.1	5
4	4	138.8	50.2	6
5	5	88	450.4	3
6	12	190.76	801.04	6
7	9	340.4	550.3	7
8	6	980.78	571.19	3
9	8	280.52	471.2	11
10	7	490.11	380.75	8

11	9	810.4	371.47	7
12	6	360.9	521.11	6
13	7	180.13	110.87	5
14	9	138.8	1020.04	3
15	10	450.3	210.2	8
16	17	801.05	730.19	17
17	12	550.36	980.1	6
18	7	61.18	770.05	8
19	6	150.61	360.08	7
20	6	113.48	680.1	8

Tab. 7. Multi-Objective meta- heuristics algorithms indicators in solving larger-size problems in *NSGA II*

Problem no	NPS	MID	D	S
1	8	30.8	1330.3	20.51
2	6	49.64	960.13	4.1
3	7	12.73	1060.3	3.26
4	8	3.29	450.4	0
5	10	12.61	700.035	0
6	7	20.86	1060.31	23.88
7	11	70.9	732.63	10.14
8	12	33.1	1083.2	0
9	11	60.22	705.8	12.88
10	9	22.41	664.43	9
11	3	41.20	816.68	7
12	7	85.03	685.12	7
13	6	88.75	738.10	10
14	4	54.4	450.4	3
15	15	32.19	834.06	10
16	14	100.57	985.21	21
17	10	155.03	906.49	9
18	3	221.26	900.2	13
19	10	107.58	870.13	16
20	13	140.41	710.20	9

Since evaluation of the performance of multi-objective algorithms directly is not possible, standard indices should be used.
The performance of these two algorithms is also evaluated with respect to the four indices of maximum expansion, problem solving time, and the difference with the ideal solution.

In this study, the t-test and 95% confidence interval were used to compare the two algorithms. T-test was used to compare the performance of the two algorithms. Using this test, we can determine whether there is a difference between the averages of the two independent populations that have a normal distribution.

6.5. Index of the number of pareto solutions

Tab. 8. Statistical test results at 95% confidence level for the index of the number of pareto solutions

algorithm	Problem no	mean	St dev	Se mean
MOCS	20	8.15	2.907	0.65
NSGA II	20	8.70	3.435	0.768
Difference	20	0.55	3.859	0.863
		p-value=0.531		

The null hypothesis indicates that the equality of the average number of solutions produced by both algorithms compared to the hypothesis of their inequality. Given the p-value, which is greater than 5%, the null hypothesis cannot be ruled out. The test results show that there is no significant difference between the numbers of Pareto solutions produced by the algorithms.

6.6. Index of spacing of the proposed algorithms

Tab. 9. Statistical test results at 95% confidence level for the index of spacing

algorithm	Problem no	mean	St dev	Se mean
MOCS	20	6.65	3.17	0.71
NSGA II	20	9.44	6.96	1.56
Difference	20	2.79	5.94	1.33
p-value=0.049				

The results show the inequality of the spacing index of the algorithms. According to the results, it is clear that the multi-objective genetic algorithm performs better than the multi-objective cuckoo algorithm.

6.7. Index of the difference with the ideal solution

Tab. 10. Statistical test results at 95% confidence level for the index of the difference with the ideal solution

algorithm	Problem no	mean	St dev	Se mean
MOCS	20	313.2	283.9	63.5
NSGA II	20	67.2	6.96	12.5
Difference	20	246	5.94	65.2
p-value=0.001				

Considering the results of the t-test, it was shown on this index that due to the fact that the p-value approaches zero, there is a significant difference between the two algorithms in terms of the mean difference from the ideal point. Also, given the lower average value of this index for multi-objective genetic algorithm, this algorithm performs better in this index.

6.8. Index of maximum extension

The results for the index of maximum extension are as follows.

Tab. 11. Statistical test results at 95% confidence level for the index of maximum extension

algorithm	Problem no	mean	St dev	Se mean
MOCS	20	537	295.3	66
NSGA II	20	857.2	265.6	59.4
Difference	20	320.2	397.9	89
p-value=0.002				

According to the results of the t-test, it is clear that the null hypothesis is rejected. Since it is better for the index of maximum extension to be higher, the multi-objective genetic algorithm performs better than the multi-objective cuckoo algorithm.

Comparing the algorithms with the aforementioned indices shows that the multi-objective genetic algorithm performs far better than the multi-objective cuckoo algorithm in terms of difference with the ideal solution, spacing and maximum extension. However, in terms of the number of Pareto solutions, no algorithm had a better performance over the other.

7. Conclusions and Recommendations for Future Studies

This paper presents a multi-objective fuzzy mathematical model for the problem of project scheduling with the constraint of multi-skilled resources able to varying skill levels, the objective of which is to optimize project scheduling and skills recruitment policies. The following are some of the innovations that distinguish the proposed model from other similar ones:

- Provision of a multi-objective mathematical model in which the cost of multi-skilled resources and time of project completion are both considered as two simultaneous objectives;
- Considering multi-state activities: In the proposed model, there are several modes for each activity, being closer to the real world;
- Provision of a mathematical model under fuzzy conditions that includes the duration of activities and the cost of using each skill level in each period;
- Considering the non-renewability constraint in the proposed model.

Since the multi-skilled project scheduling is among the NP-Hard problems and the proposed problem is its extended state, so it is also an NP-Hard problem. Therefore, *NSGA II* and *MOCS* algorithms are used to solve the proposed model in large sizes in Matlab. Initially, the optimal parameters of the proposed algorithms are determined using the Taguchi approach. Then, the computational results are presented for a set of sample problems generated by RanGen 1, and the performance of the algorithms is evaluated. The results show that the multi-objective genetic algorithm performs better than the multi-objective cuckoo algorithm in terms of the ideal solution, spacing, and maximum extension.

We can recommend the following to develop and bring this research area closer to the real world in future research:

- Using other meta-heuristic algorithms to compare the results with NSGA II and MOCS;
- Using various other objective functions such as minimizing project completion time;
- Considering renewable resources in addition to problem constraints.

References

[1] Imannezhad R, Avakh Darestani S. Project scheduling problem with resource constraints and activities interruption using Bees algorithm. International Journal of Industrial Engineering & Production Research 10;Vol. 29, No. 3, (2018), pp. 277-291.

[2] Noori S, Taghizadeh K. Multi-mode resource constrained project scheduling problem: a survey of variants, extensions, and methods. International Journal of Industrial Engineering & Production Research. 10; Vol. 29, No. 3, (2018), pp. 293-320.

[3] Néron, E. and Baptista, D. Heuristics for the multi-skill project scheduling problem. In International Symposium on Combinatorial Optimization (2002).

[4] Bellenguez-Morineau, O. and Néron, E. A branch-and-bound method for solving multi-skill project scheduling problem. RAIRO-operations Research, Vol. 41, No. 2, (2007), pp.155-170.

[5] Bellenguez-Morineau O. “Methods to solve multi-skill project scheduling problem” (Doctoral dissertation, Springer-Verlag) (2008).

[6] Valls, V., Pérez, Á. and Quintanilla, S.” Skilled workforce scheduling in service centers”. European Journal of Operational Research, Vol. 193, No. 3, (2009), pp.791-804.

[7] Arashpour, M., Wakefield, R.,Blismas, N. and Minas, J. Optimization of process integration and multi-skilled resource utilization in off-site construction. Automation in Construction, Vol. 50, (2015), pp.72-80.

[8] Correia, I. and Saldanha-da-Gama, F. A note on “branch-and-price approach for the multi-skill project scheduling problem”. Optimization Letters, Vol. 9, No. 6, (2015), pp.1255-1258.

[9] Javanmard, S., Afshar-Nadjafi, B. and Niaki, S.T.A. Preemptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches”. Computers & Chemical Engineering, Vol. 96, (2017), pp. 55-68.

[10] Almeida, B.F., Correia, I. and Saldanha-da-Gama, F. “Priority-based heuristics for the multi-skill resource constrained project scheduling problem. Expert

Systems with Applications, Vol. 57, (2016), pp. 91-103.

[11] Maenhout B, Vanhoucke M. "A hybrid scatter search heuristic for personalized crew rostering in the airline industry. European Journal of Operational Research" Vol. 206, No. 1, (2010), pp. 155-167.

[12] Möhring RH. " Minimizing costs of resource requirements in project networks subject to a fixed completion time". Operations Research. Vol. 32, No. 1, (1984), pp. 89-120.

[13] Demeulemeester E. " Minimizing resource availability costs in time-limited project networks. Management Science". Vol. 41, No. 10, (1995), p. 159.

[14] Drexl A, Kimms A. "Optimization guided lower and upper bounds for the resource investment problem". Journal of the Operational Research Society, 1; Vol. 52, No. 3, (2001), pp. 340-351.

[15] Shadrokh S, Kianfar F. "A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty". European Journal of Operational Research, 16; Vol. 181, No. 1, (2007), pp. 86-101.

[16] Afshar-Nadjafi B. "Multi-mode resource availability cost problem with recruitment and release dates for resources. Applied Mathematical Modelling". 1;Vol. 38, Nos. 21-22, (2014), pp. 5347-55.

[17] Afshar-Nadjafi B, Arani M. "Multimode preemptive resource investment problem subject to due dates for activities: formulation and solution procedure". Advances in operations research (2014).

[18] Salimi M, Najafi AA. "Modeling and Solution Procedure for a Preemptive Multi-Objective Multi-Mode Project Scheduling Model in Resource Investment Problems". Journal of Optimization in Industrial Engineering. 1; Vol. 11, No. 1, (2018), pp. 169-183.

[19] Ramík,J.etal Inequality relation between fuzzy numbers and its use in fuzzy optimization. *Fuzzy sets and systems*, Vol. 16, No. 2, (1985), pp. 123-138.

[20] Y. J. Lai and C. L. Hwang, Fuzzy Mathematical Programming Methods and Applications, Springer, Berlin (1999).

[21] Yang, X. S., Deb, S. "Cuckoo search via L´evy flights, Proceeings of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009, India)", IEEE Publications, USA, (2009), pp. 210-214.

[22] Rajabioun, R. "Cuckoo Optimization Algorithm", Applied Soft Computing, Vol. 11, (2011), pp. 5508-5518.

[23] Yang, X.S. and Deb, S. "Multiobjective cuckoo search for design optimization", Comput. Oper. Res., Vol. 40, No. 6, (2013), pp. 1616-1624.

[24] Karimi, Neda, Mostafa Zandieh, and H. R. Karamooz. "Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach." Expert Systems with Applications Vol. 37, No. 6, (2010), pp. 4024-4032.

[25] Zitzler E., Deb K., Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation Vol. 8, No. 2, (2000), pp. 173-195.

[26] Schott J. Fault tolerant design using single and multicriteria genetic algorithms optimization.Department of Aeronautics and Astronautics. Cambridge: Master's thesis, Massachusetts Institute of Technology (1995).

Follow This Article at The Following Site:

Salehi M, Jabarpoor E. Multi-objective fuzzy modeling of project scheduling with limitations of multi-skilled resources able to change skill levels and interrupt activities. IJIEPR. 2021; 32 (3) :1-16
URL: http://ijiepr.iust.ac.ir/article-1-958-en.html