Scientia Iranica E (2017) 24(2), 765–777

SCIENTIA IRANICA

Sharif University of Technology

Scientia Iranica

Transactions E: Industrial Engineering

www.scientiairanica.com

A novel mathematical model for a scheduling problem of dynamic machine-tool selection and operation allocation in a flexible manufacturing system: A modified evolutionary algorithm

M.H.M.A. Jahromi[a], R. Tavakkoli-Moghaddam[b,*], A. Makui[c] and A. Saghaei[d]

a. *Department of Industrial Engineering, Damavand Branch, Islamic Azad University, Damavand, Iran.*
b. *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.*
c. *Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.*
d. *Department of Industrial Engineering, Science & Research Branch, Islamic Azad University, Tehran, Iran.*

Received 22 January 2015; received in revised form 6 April 2016; accepted 10 May 2016

KEYWORDS
Flexible manufacturing systems;
Mathematical model;
Evolutionary algorithm;
Island model.

Abstract. Although a machine-tool selection and operation allocation problem of a Flexible Manufacturing System (FMS) is known for its complexity, scheduling of these systems is more operative and more complex. This paper considers scheduling of an FMS with dynamic machine-tool selection and operation allocation. In addition, due to the NP-hard nature of this problem, a modified Evolutionary Algorithm (EA) considering an island model is proposed to solve the given problem. Its performance is tested on a number of randomly generated problems. Furthermore, the related results are compared with the results obtained by a Branch-and-Bound (B&B) method. It has been found that the modified EA with the island model gives good results in terms of the objective function values and CPU times.

1. Introduction

In the recent years, importance of flexible manufacturing systems has increased. This is due to their capability to keep benefits of job shop with high variety of production and flow shop with a high volume of production together. In fact, reduction in the life cycle of products due to the role of Research and Development (R&D), marketing activities, and many other factors leads to increasing tendency of international producers to use these systems. In an FMS, many multi-functional Computer Numerical Control (CNC) machines are linked together by material handling systems and Tool Delivery Systems (TDSs) so that a central computer controls all of them. Flexibility of these systems offers different machines and tools for performing each operation. In other words, each operation can be performed by several machines and tools. This problem is considered as assignment (or allocation) problem that belongs to an upper level of decision making in production planning. In fact, in this level, determining the best path among several feasible paths is the goal, in which each path has its own production time and cost. In this level, researchers have tried to find a path whose cost and related issues are optimum. One of the most important subsets of assignment problems, which are more operative, is the scheduling of performing operations in the shop floor. Many researchers have considered this level of

*. *Corresponding author. Tel.: +98 21 82084183; Fax: +98 21 88013102*
E-mail address: tavakoli@ut.ac.ir (R. Tavakkoli-Moghaddam)

production planning that is a level lower than the allocation level. In this level, time and its issues are more important than cost. Objectives, which are significant in this level, are to minimize the time required to complete all orders named minimizing the makespan, minimizing the largest difference between completion times and due dates named minimizing maximum tardiness, minimizing the total processing time, etc. Due to the complexity of assignment and scheduling problems in these systems, both of them are known as NP-hard problems [1,2].

In this research, developing a novel 0-1 mathematical model for an FMS scheduling problem with a dynamic policy is considered and a modified Evolutionary Algorithm (EA) for solving this problem is proposed. EA has a number of components and operators (e.g., representation, fitness function, population, crossover operator, mutation, survival selection, and termination condition). The modified EA contains special crossover and mutation operators as well as a new encoding representation of chromosome. Most EAs use a single population of chromosomes and apply the corresponding operators to them [3]. In order to benefit from migration in the modified EA, many subpopulations are considered using an island model [4-6]; most authors use islands and subpopulations equivalently. In the modified EA, in addition to designing specific operators, instead of a central population, several islands are applied on which the evolutionary process is performed independently on a single core (machine). In each set of generations (usually consisting of 20 generations), a certain percentage of the best chromosomes are exchanged between neighboring islands as migration. These migrations lead to improvement in the average quality of sub-populations [6].

Many researchers have developed mathematical modeling and heuristic methods to solve the assignment and scheduling problems in FMS environments. As mentioned earlier, due to NP-hard nature of these problems, heuristic or meta-heuristic algorithms have been used for solving them. In this field, many researchers have considered only operation assignment to machines [7], and some others have considered a tool role in their studies [2,8-10]. Three policies can be used when tool roles are considered in these systems: first, a part movement policy, in which tools cannot move around the machine in the planning horizon [8,9, 11-13]; second, a tool movement policy, in which tools can move around the machines and parts remain on a machine in the whole planning horizon [10]; and third, a part and tool movement policy, in which parts and tools can move around the machines simultaneously with their corresponding Automated Guided Vehicles (AGVs) [2,14]. As mentioned earlier, scheduling problems of parts (orders) in the shop floor are more operative than allocation problems. Gamila and Motavalli [15] presented a 0-1 Mixed Integer Programming (MIP) model, considering a part movement policy, for loading and scheduling in an FMS. Their work primarily determined the loading and routing of parts and tools and, in the next step, scheduling of them was performed. Persi et al. [16] proposed a hierarchical approach for scheduling of the FMS. At the upper level of their work, sets of the performed orders were determined and, in the next level, orders were scheduled.

Tool loading times are considered as an appropriate index for sequencing and scheduling. Low et al. [17] developed a multi-objective model for scheduling of an FMS problem considering three objectives that were to minimize the mean job flow time, mean job tardiness, and mean machine idle time, simultaneously. They utilized a hybrid heuristic algorithm for solving the FMS scheduling problem. Additionally, they considered assignment of operations to machines without any tool role in their work. Furthermore, each operation had a due date that had to be considered in the scheduling. Erhan Kesen et al. [18] presented a Genetic Algorithm (GA) based heuristic approach for job scheduling. Solely considering the machines, they presented a case study where there were multiple jobs with multiple machine types. Minimization of the weighted makespan and total traveling distance was considered as the objective function. Kazerooni et al. [19] developed an integrated decision-making support system with the fuzzy approach and Analytical Hierarchy Process (AHP) for scheduling of an FMS using simulation that analyzed and compared the performance of different alternative combinations of machine selection and dispatching rules. Their results showed high performance of their approach. Also, many good studies have been presented by researchers as survey of a simulation study on the FMS scheduling [20,21]. A simulation study with the purpose of evaluating makespan, average flow time, average delay time at local buffers, and average machine utilization of a flexible manufacturing system with different control strategies contains routing flexibilities and dispatching rules as presented by Chan and Chan [22]. Chan et al. [23] presented a genetic algorithm for FMS scheduling problems with alternative production routing. In their proposed GA, determining crossover and mutation rates was not needed and the algorithm itself determined whose genes had to undergo crossover. The purpose of the proposed GA was to obtain local optimum solutions in a reasonable time.

Prakash et al. [1] proposed a knowledge-based GA for scheduling of an FMS considering assignment of operations to machines, which used the power of tactic and implicit expert knowledge. Initialization, selection, crossover, and mutation in the GA are the four stages that the knowledge has been used by considering throughput and mean flow time as objectives. Chan et al. [24] addressed a distributed FMS scheduling problem. In this type of problem, assigning the orders to each factory and determining the scheduling of each factory are significant. Each factory may be distributed in different locations and can produce a variety of orders. They proposed a GA for solving the problem and considered maintenance in their work. The physical and operating parameters with respect to flexibility levels and control strategies have been studied by Chan et al. [25]. They considered deterministic environment and presented simulation study under the Taguchi's method. Their results showed that in the performance of FMS, the relative contribution percentage of variations in physical and operating parameters of different resources was insignificant. In another similar work [26], they indicated that routing flexibility and control strategy had significant relative contribution percentages in the performance, while the decision-and-information system was insignificant. The remainder of this paper is organized as follows. Section 2 describes the problem briefly and gives the assumptions for its formulation. Section 3 presents a mathematical programming model for the scheduling of the dynamic FMS problem. Section 4 is dedicated to the modified EA and island model. The computational results and discussion of the proposed EAs are presented in Section 5. Finally, the conclusions are given in Section 6.

2. Problem description

An FMS consists of many CNC machines each of which can be equipped with several tools for performing operations. The production cost and time in an FMS depend on the machine-tool combinations that are used for performing operations, because each of them has its cost and time. Assignment of the required tools in the right times and right places to tool magazines of machines, known as tool management, is difficult and significant. There are three policies in these systems in the whole planning horizon: part movement policy, tool movement policy, and part and tool movement policy (i.e., dynamic policy) simultaneously. At the beginning of the production in the part movement policy, tools are fixed to appropriate machines and just parts move around the machines by AGVs for parts. On the contrary, in the tool movement policy, parts are assigned to the machines at first and tools move around the machine with TDSs by AGVs for tools. Finally, in the third policy, parts and tools can move around the machine with corresponding AGVs in the planning horizon, simultaneously. Although planning in the last policy is more complex, it has more flexibility. It is also expected that the makespan may decrease in this situation. The following assumptions are considered for the concerned problem.

- Each part type (order) has its own operation(s), and each of them can be performed by different alternative machine-tool combinations;
- Each operation can be performed with only one machine-tool combination in the whole planning horizon;
- Performing each operation on each machine-tool combination has its own time depending on the combination;
- Each tool can be placed in each machine so that the goal of increasing flexibility is achievable;
- Tools movement time by AGVs for tools and parts movement time by AGVs for parts and construction time of their combination are not considered;
- Precedence relationship between operations of each part type is considered;
- At each time slot of the planning horizon, a tool can make at most one machine-tool combination and it cannot be duplicated;
- Raw materials are kept up in the corresponding storage and after performing the last operation of each part type, they will be sent to the finished product storage. The corresponding traveling times of storages are ignored;
- There are no constraints on availability of AGVs, pallet, fixture, etc.

3. Mathematical formulation

As mentioned earlier, scheduling of operations with dynamic machine-tool selection and operation allocation is considered in this paper. In this work, minimization of the makespan is significant as the objective function.

3.1. Notations

sets

p	Index of parts; $p = 1, ..., P$, where P is the number of part type.
o	Index of operation(s) of part type; $o = 1, ..., Op$, where O_p is the last operation of part type p.
l	Index of tools; $l = 1, ..., L$, where L is the total number of tools.
m	Index of machine; $m = 1, ..., M$, where M is the number of available machines.
k	Index of time slot; $k = 1, ..., K$, where K is the maximum time horizon.

Parameters

T_{poml}	Machining time for operation o of part type p using machine-tool combination $m - l$.
C_p	Completion time of part type p.

Decision variables

y_{poml} 1 if operation o of part type p is performed using machine m and tool l; and 0, otherwise.

v_{pomlk} 1 if operation o of part type p is performed using machine m and tool l in time slot k; and 0, otherwise.

s_{pok} 1 if performing of operation o of part type p is started in the kth slot of time horizon; and 0, otherwise.

f_{pok} 1 if performing of operation o of part type p is finished in the kth slot of time horizon; and 0, otherwise.

3.2. Mathematical model

This section presents the mathematical model for the scheduling problem:

$$\text{Min } z = \{\text{Max}\{C_p\}\}, \tag{1}$$

s.t.

$$C_p = \sum_m \sum_l (y_{p,o,m,l} * T_{p,o,m,l}) + \sum_k k * s_{p,o,k}, \tag{2}$$

$$\sum_k s_{p,o,k} = 1 \quad \forall p, o, \tag{3}$$

$$\sum_k f_{p,o,k} = 1 \quad \forall p, o, \tag{4}$$

$$\sum_k k\,(s_{p,o+1,k} + s_{p,o,k}) \geq 0 \quad \forall p, o, \tag{5}$$

$$\sum_m \sum_l y_{p,o,m,l} = 1 \quad \forall p, o, \tag{6}$$

$$\sum_k v_{p,o,m,l,k} \geq y_{p,o,m,l} \quad \forall p, o, m, l, \tag{7}$$

$$y_{p,o,m,l} \geq v_{p,o,m,l,k} \quad \forall p, o, m, l, k, \tag{8}$$

$$\sum_k s_{p,o,k} \geq y_{p,o,m,l} \quad \forall p, o, m, l, \tag{9}$$

$$\sum_k f_{p,o,k} \geq y_{p,o,m,l} \quad \forall p, o, m, l, \tag{10}$$

$$\sum_k v_{p,o,m,l,k} = T_{p,o,m,l} * y_{p,o,m,l} \quad \forall p, o, m, l, \tag{11}$$

$$\left(\sum_k k * f_{p,o,k}\right) - \left(\sum_k k * s_{p,o,k}\right) = \sum_m \sum_l T_{p,o,m,l} * y_{p,o,m,l} \qquad \forall p, o, m, l, \tag{12}$$

$$\textstyle\sum_m \sum_l v_{p,o,m,l,k} \leq 1 \quad \forall p, o, k, \tag{13}$$

$$\sum_p \sum_o v_{p,o,m,l,k} \leq 1 \quad \forall m, l, k, \tag{14}$$

$$\sum_p \sum_o \sum_m v_{p,o,m,l,k} \leq 1 \qquad \forall l, k. \tag{15}$$

The objective function (1) minimizes the makespan (i.e., the time required to complete all orders). In the above mathematical model, completion time, C_p, of part type, p, equals the ending time of the last operation, O_p, of that part, p, and can be calculated according to Eq. (2). Eq. (3) states that each operation, o, of each part must be started in a time slot. Eq. (4) insures that each operation, o, of each order is finished in a time slot. Precedence constraints between operations of each part type are guaranteed with Constraint (5). Constraint (6) ensures that each operation, o, of each part, p, is performed with one machine-tool combination $m - l$. Constraint (7) guarantees that if operation, o, of part, p, is assigned to machine-tool combination, $m - l$, it is performed in at least one time slot k. Also, this constraint states the relationship between two zero-one variables. Constraints (8)-(10) indicate the relationship between the binary variables. Performing each operation, o, of each part type, p, with each machine-tool combination, $m - l$, needs a given time, T_{poml}. Eq. (11) states, as much, the given time must be time slot(s) of the assigned machine-tool combination, $m - l$. Eq. (12) assures that once operation o starts, it will be terminated consecutively on time slot(s) without interruption. Constraint (13) shows that each operation o of each part type p can be performed in a time slot k at most with one machine-tool combination, $m - l$. Constraint (14) states that with each machine-tool combination, $m - l$, at each time slot k, at most one operation o can be processed. Constraint (15) guarantees that each tool in each time slot can be used at most in one machine for performing operations.

4. Evolutionary algorithm

Evolutionary Algorithms (EA) are subsets of population-based heuristic algorithms. They are most famous as stochastic search methods and use mechanisms (e.g., crossover and mutation) inspired biological evolution. In these algorithms, the fitness function of each candidate solution, as individuals, determines the quality of the solutions. By repeating the application of many operators (Figure 1), evolution of the population takes place [27].

4.1. Modified evolutionary algorithms

As mentioned earlier, due to the NP-hard nature of the given scheduling problem, finding an optimal solution in a reasonable time to large-size problems is not possible or is said to be costly effective. Although

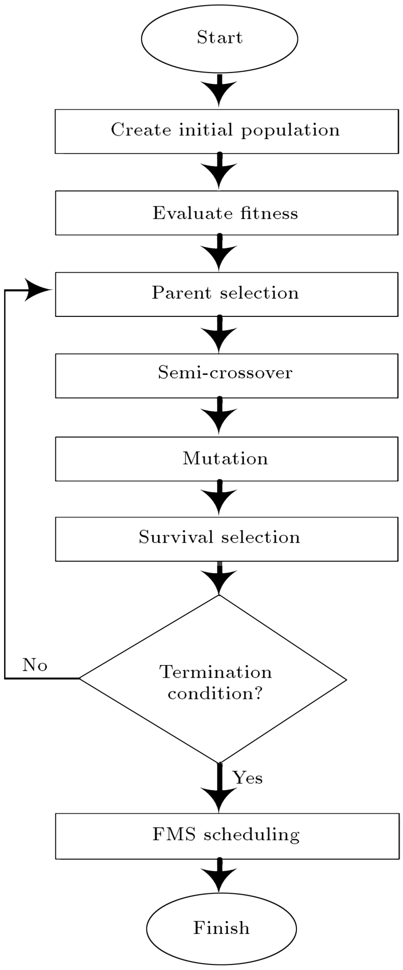

Figure 1. Framework of the proposed EA.

the exact methods find global optimum in small-size problems in most cases, a trade-off between cost and time is needed. In fact, if suitable non-exact methods are selected and designed well, then it can be expected that they can find good solutions. In this paper, a modified evolutionary algorithm is presented with its own representation and particular operators given in the following subsections are explained.

4.1.1. Representation

A good representation of solutions is the first and most important step in increasing the efficiency of any meta-heuristic algorithm. In the proposed EA, each chromosome represents a possible solution to scheduling of part types and has n genes, where n is the summation of operations of all parts. In this representation, each gene of a chromosome is related to one operation of one part type, respectively. In fact, the leftmost gene is for the first operation of the first part type, p_1o_1, and the rightmost gene of each chromosome is for the last operation of the last part type, P_{op} (Figure 2). Also, each gene consists of three parameters representing machine m, tool l, and scheduling priority s of that operation on the corresponding machine, respectively. For example,

p_1o_1	p_1o_2	p_1o_3	.	.	.	P_{OP}
mls	mls	mls	.	.	.	mls

Figure 2. Representation of a chromosome with its genes.

if the first gene has been valued "213" randomly, it means that operation 1 of part type 1 will be performed on machine 2 combined with tool 1, and the scheduling priority of this operation is after two other operations and this operation is the third operation that must be considered on machine 2 (scheduled).

Genes of each chromosome for assigning the machine-tool combination and scheduling priority are selected randomly among those not still valued so that precedent relationships between operations of each part can be considered. Selection order s of each gene is its scheduling priority. Also, for the selected gene, a machine-tool $m - l$ combination must be selected randomly.

4.1.2. Initialization population

At the first step of the EA, according to the data of the problem as input data, a set of chromosomes as random candidate solutions must be generated.

4.1.3. Fitness function

As mentioned earlier, each chromosome is constructed by assigning the machine-tool combination and scheduling priority of its machines to its genes, randomly. It is noted that each machine-tool combination has its particular time for performing the corresponding operation of that gene. Thus, the same time slot(s) is (are) needed for performing that operation numerically. The fitness value in the modified EA is the objective function (i.e., makespan) of the developed mathematical model. Among the machines used in each chromosome, chr, their machining finish time, FT_m, in the corresponding production planning is considered and the fitness of that chromosome is computed by:

$$\text{Makespan}(chr) = \text{Max}(FT_1, FT_2, ..., FT_M). \qquad (16)$$

Figure 3 shows a given chromosome for a problem with three parts, each having three operations. In this chromosome, just two machines and two tools are used. The algorithm has assigned machine-tool combinations randomly for operations (i.e., as genes in the chromosome) considering the sequence of performing them on each machine.

According to the produced solution, the sequence of performing operations on $m1$ is p_2o_1, p_1o_2, p_3o_2, and p_2o_3, respectively, and this sequence for $m2$ is p_1o_1, p_3o_1, p_1o_3, p_2o_2, and p_3o_3, respectively. Figure 4 represents the real scheduling of the randomly generated chromosome (i.e., Figure 3) so that it has two rows, each for each machine used in the chromosome; also, k is the time slot and the four numbers in each box mean

p_1o_1	p_1o_2	p_1o_3	p_2o_1	p_2o_2	p_2o_3	p_3o_1	p_3o_2	p_3o_3
221(2)	122(3)	223(2)	111(3)	224(2)	124(3)	212(4)	113(3)	215(2)

Figure 3. Randomly generated chromosome with three parts.

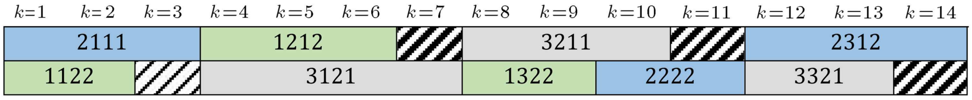

Figure 4. Real scheduling of the mentioned chromosome.

part number, operation number for each part, machine number, and tool number, respectively. Moreover, hatch lines in each row indicate the machine idle times. Chromosome fitness is 14 according to Eq. (17):

$$\text{Makespan (the example chromosome)} = \max(14, 13) = 14. \tag{17}$$

4.1.4. Parent selection
Parent selection is a mechanism for selecting appropriate individuals from the current population (generation) in order to produce the next population. The most popular parent selection mechanism used in EAs is fitness proportionate selection that has proved to have many problems [28] such as:

- A chromosome with very high quality can quickly defeat the others if the quality of the rest of the population is very low;
- In the last generations, the chromosomes will be similar and have the same fitnesses. Then, the parents selection will be difficult and it is like the random selection;
- It is very sensitive to the function transposition.

Although many approaches have been proposed that overcome some of the drawbacks of the fitness proportionate selection, this mechanism needs too much computing time to sort the solutions. Because of this problem in the proposed EA, tournament selection has been used as a parent selection mechanism. Tournament selection is good method that has not the above drawbacks and picks k (i.e., tournament size) members randomly; then, it selects the best of them and repeats the selection for more individuals. In this method, adjusting of selection pressure is easy and this occurs by changing the size of the tournament. For example, if the size of the tournament increases, then the chance of selecting weak chromosome becomes less.

4.1.5. The proposed crossover operator
Crossover is one of the popular operators of EAs that tries to use a combination of genes of parents to produce new chromosomes that share many characteristic with their parents as offspring (i.e., child). In the 1-point crossover, a number is selected randomly that corresponds to the length of parent chromosomes; then, the genes are exchanged between parents in order to create two new offspring. In the case of N-point crossover, the same N points are selected randomly and relocation of them is performed. In uniform crossover, each gene of each offspring inherits a chromosome from the parent with a pre-determined probability [29,30]. Each of these operators has different destruction rates that means how much producing offspring is different from their parents. N-point and uniform crossovers are much suitable for the proposed EA if their destruction rate can be controlled during the running of the EA. The destruction rates of these two operators are high, which is too dangerous at the end of a generation. In this paper, a new kind of N-point crossover is proposed that can be flexible for having different destruction rates. The whole generation is divided into three stages. At the first stage, the need to be cautious for destruction is not important; EA randomly chooses the value of N close to the length of the chromosome. In the middle stage of running the EA, it randomly uses the amount of N that can be appropriate to the half size of the chromosome. In the last stage of running the EA, the amount of the destruction rate must be low so that crossover acts similar to 1-point crossover with lower dissimilarity results in the offspring.

In the proposed EA, due to the nature of the scheduling problem and considering different machine-tool, $m - l$, combinations, a specific heuristic crossover as 1-point crossover, namely, semi-crossover operator is designed. In this kind of crossover, only a machine-tool, $m - l$, combination between two parent chromosomes must be exchanged. In fact, scheduling priorities identically one by one will be transferred as the parents as shown in Figure 5.

In order to retain the efficiency of the crossover operator in the proposed EA, another type of this operator is used to aid the algorithm in avoiding the premature convergence to local optimum. In this type of semi-crossover, only the third parameters of genes are exchanged between two parents one by one, and two offspring are produced. These two types of semi-crossover operator are used in the proposed EA with a given probability. It is notable that after applying semi-crossover operators, another algorithm is used for making the production of new offspring feasible. In fact,

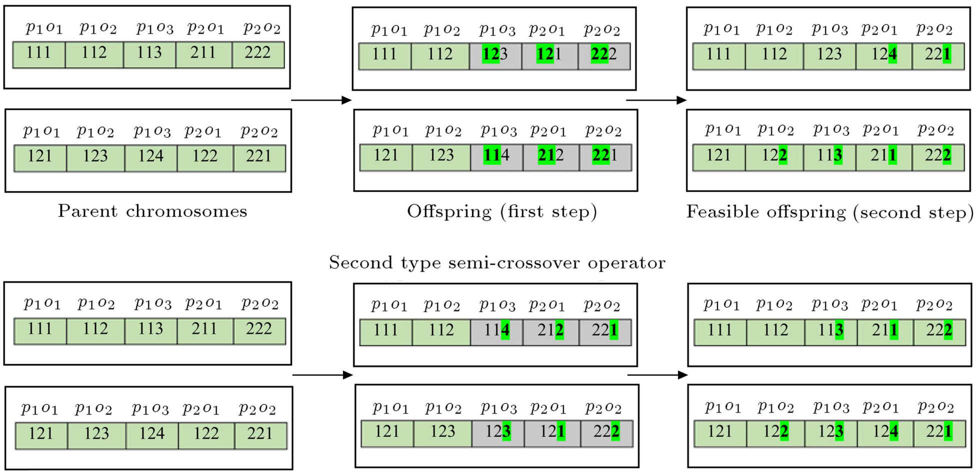

Figure 5. Two types of semi-crossover operator applied in the proposed EA.

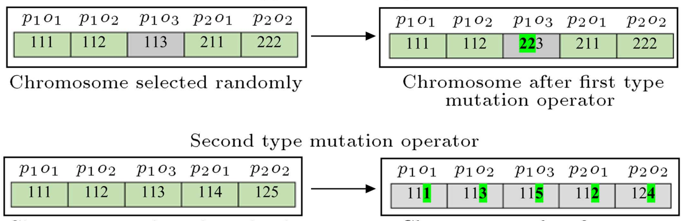

Figure 6. Mutation operators used in the proposed EA.

this algorithm is checked if the assigned machine-tool is not proper for performing the intended operation. The algorithm considers a high machining time for it (i.e., M) so that the performance of the chromosome declines and somehow a chromosome can be removed because this chromosome has high penalty and low efficient. Also, for duplicated priority parameters as the third parameter, the algorithm is checked for any infeasibility. If necessary, the priority of performing an operation on a machine is modified so that minimum change occurs compared with the initial state and, if necessary, the algorithm considers the feasibility condition and assigns the priority of performing this operation randomly.

4.1.6. Mutation

Another popular operator of EAs is mutation that is similar to biological mutation and it is used in order to retain diversification ability of the algorithm. The mutation operator tries to alter the value of gene(s) from its initial state in order to help the algorithm escape from local optimum. By applying this operator, the new offspring may be entirely different from the previous one. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search [28] and this operator can control the speed of convergence of the algorithm to local optimum [31]. In order to utilize the benefits of this operator, a kind of mutation is designed in the proposed EA. In this kind of mutation, each time a chromosome is selected randomly, a gene must be selected randomly and its machine-tool, the first and second parameters, undergoes mutation. In this type, a new machine-tool combination is assigned to the corresponding operation and after that, if the correct scheduling priority between operations of each part is violated, another algorithm is used to change the new chromosome to a feasible one. In the second-type mutation, just the scheduling priority of all genes, the third parameter of the selected chromosome, must be determined again randomly. It is noted that this exchange should not violate precedence constraints between operations. It must be noted that in the proposed EA, the mutation with a predetermined rate is used (Figure 6).

4.1.7. Survival selection

A survival selection mechanism is used in EAs to distinguish among chromosomes based on their quality and it is used after creating offspring by crossover and mutation. In fact, it seeks the chromosomes that are qualified to go to the next generation among the current population and offspring. In the proposed EA, 70% of the best offspring and the best 30% of the current population are capable to go to the next population (i.e., survived).

4.1.8. Termination condition
A termination condition checks every generation. If one of the two points of reaching stagnation of fitness or reaching the allowed maximum number of generations is satisfied, the algorithm will be stopped.

4.2. ***Island model***
As mentioned earlier, the idea of the island model that is applied to the modified EA utilizes many subpopulations (i.e., islands) to keep diverse directions in the search space at the same time. Increasing speed of the search process that leads to finding solutions with higher quality through generations is the advantage of this kind of EAs [6,32]. In this work, the modified island model is a multi-population model that enables the exchange of individuals among subpopulations (i.e., migration) by a pre-determined rate. It is notable that *migration interval* (or *migration frequency*) is the number of generations after which migration should occur and migration rate determines how many individuals should migrate at the predetermined times. These are two key parameters of the island model [30]. Island model contains migration mechanism, which has many advantages over the simple EA, and migration algorithms are more likely to evolve in the nature. Also, they help an algorithm not to get stuck in local optimum by injection of individuals to each generation. Ultimately, it expectedly provides more diversification compared to ordinary models. Having multiple subpopulations helps to preserve diversity of the algorithm, since each island can potentially follow a different search path through the search space. In implementing the island model, all of the islands can be executed in a core (i.e., machine) as a serial single population model or multiple cores (i.e., machines) as parallel implementation.

In this paper, the first method is applied. Each island works in association with other islands by periodically exchanging a portion of its population in a process called migration. Improvement in search quality is due to the fact that various islands maintain some degrees of independence and explore different regions of the search space while, at the same time, they share information by means of migration, which helps to sustain diversity of the algorithm. In this approach, a number of islands are considered whose pre-determined chromosomes or population sizes are identical. In the proposed island model, right after selection and before mating, the migration mechanism is applied and afterwards, the local iteration of EA resumes. In the proposed island model, at the time of migration or in the communication phase, the four best chromosomes are exchanged with four worse ones between each pair of neighbors. Figure 7 shows an island model that contains three islands, each of which running its evolutionary algorithm; in the predetermined interval time, migration between islands with predetermined size occurs.

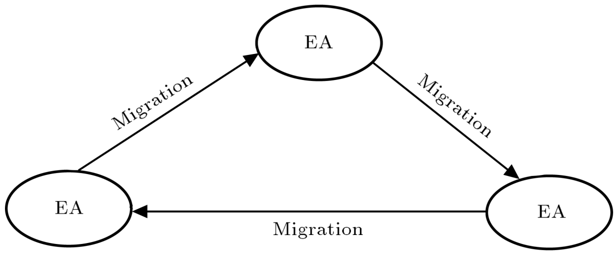

Figure 7. Island model consisting of three islands.

5. Computational results

As mentioned before, the EA is an intelligent exploitation of a random search that maintains a population of n chromosomes with the associated fitness values. Parents are selected to mate, based on their fitness, producing offspring via a reproductive plan. Consequently, highly fit solutions are given more opportunities to reproduce so that offspring inherit characteristics from each parent. As parents mate and produce offspring, room must be made for the new arrivals since the population is kept at a static size. Individuals in the population die and are replaced by the new solutions, eventually creating a new generation once all mating opportunities in the old population have been exhausted. In this way, it is hoped that by successive generations, better solutions will thrive while the least fit solutions die out.

New generations of solutions are produced containing, on average, better genes than a typical solution in the previous generation. Each successive generation will contain more good 'partial solutions' than the previous generations. Eventually, once the population has converged and it does not produce offspring noticeably different from those in the previous generations, the algorithm itself converges to a set of solutions to the problem at hand.

In this paper, many problems with different sizes are produced randomly and solved with the developed mathematical model by a Branch-and-Bound (B&B) method as exact method for finding global optimum. The proposed EA and island model do not guarantees on finding a global solution. The solutions are obtained by the B&B method using LINGO 10.0 software. The proposed EA is coded using Microsoft Visual C# on a computer with 2.0 GB Ram and Intel Core 2 Duo 3.0 GHz processor. Table 1 shows the parameters

Table 1. Details of parameters for the random sample problems.

Parameters	**Range**
Shop	
Number of machines	Uniform (1-5)
Number of tools	Uniform (2-6)
Machining time of each machine-tool $m - l$ combination	Uniform (1-10)
Orders	
Number of part types	Uniform (1-15)
Number of operations	Uniform (1-10)

Table 2. EA and island model parameters.

	Island model	**EA**
Number of islands	5	-
Subpopulation size	100	-
Migration size	4	-
Migration interval	20	-
Population size	5×100	500
Crossover probability	0.9	
Mutation probability	0.1	
Termination condition	700 generations or stagnation	
Fitness function	Makespan	

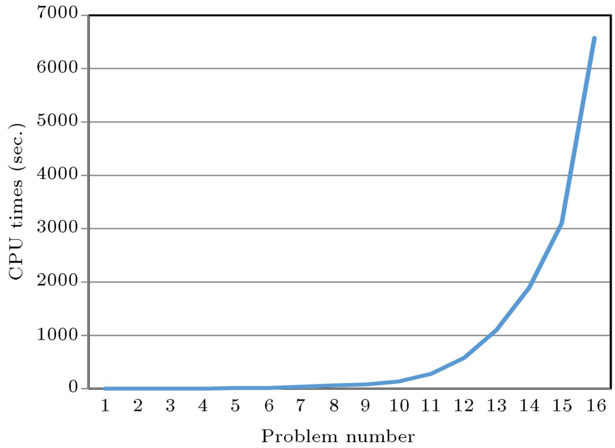

Figure 8. CPU time of the B&B method.

used for making the random FMS problems. For this purpose, a set of 16 problems ranging from small and medium-size problems to 11 large-size problems is considered. Details of these problems and the resultsobtained by the methods are shown in Tables 2 and 3. In Table 3, the makespan and CPU time are shown as the two performance measurement factors of the proposed methods. It is obvious that due to the NP-hard nature of the FMS scheduling problem, finding global optimum for large-size problems is not feasible and it increases exponentially as shown in Figure 8. According to the obtained results, the CPU times of the modified EAs for large-size problems in comparison with those of the B&B method are more efficient. Although this is predictable, the quality of the EAs should be checked. In fact, if the proposed EA for small and medium-size problems is not capable to find good solutions, resembling global solutions, then the hope for finding good solutions to large-size problems is not reasonable. According to the obtained results of EA, which are presented in Table 3, makespan deviation mean of the EA and B&B methods is 3.7% that shows the efficiency of the EA for this problem in finding global solutions. This value for the island model is 1.6% that shows it is more efficient. Due to benefits of the proposed island model, migration advantages, diversification, and intensification are used more and this fact leads to better solutions. It should be noted that due to applying of many EAs simultaneously in the island model, the computational times is more; thus, this problem can be solved by using multiple cores and assigning each EA to one of them as parallel methods. According to the obtained results, the gap between the island model and EA for small and medium-size problems is 1.8% that shows the efficiency of the island model.

In order to check the quality of the solutions among the generations, two small and large-size problems generated randomly are selected and quality of the process is shown in Figures 9 and 10. For this purpose, the average fitness of 700 generations of each problem are considered separately. According to Figure 9, for the small-size problems, both the EA and island model have reached the global optimum solution obtained by the B&B method; but, the island model has better average quality than EA and it reaches the global optimum faster. This shows the more efficiency of the island model. About the large-size problems, the average quality of the island model is better that of the EA as well. It is noted that in large-size problems, the differences between the average qualities of the solutions have been found by the EA.

6. Conclusion

In this paper, a novel 0-1 linear mathematical model has been represented for the scheduling problem of a Flexible Manufacturing System (FMS) with a part

Table 3. The results of EA, island model, and the B&B method.

	Problem	Makespan						CPU time (sec.)		
		EA	**Island model**	**B&B method**	**Gap EA and B&B (%)**	**Gap Island model and B&B (%)**	**Gap Island model and EA (%)**	**EA**	**Island model**	**B&B method**
Small-size problems	$p(1)o(2)m(2)l(4)$	6	6	6	0.0	0.0	0.0	3	14	1
	$p(1)o(4)m(4)l(6)$	7	7	7	0.0	0.0	0.0	3	16	1
	$p(2)o(3,3)m(3)l(3)$	12	12	12	0.0	0.0	0.0	6	25	3
	$p(2)o(2,6)m(2)l(4)$	13	13	13	0.0	0.0	0.0	7	37	5
	$p(3)o(5,2,3)m(3)l(5)$	17	17	17	0.0	0.0	0.0	8	44	12
	$p(3)o(3,4,4)m(2)l(5)$	18	18	18	0.0	0.0	0.0	9	47	17
	$p(4)o(5,4,2)m(3)l(6)$	22	22	22	0.0	0.0	0.0	6	49	41
	$p(4)o(2,5,3,7)m(4)l(5)$	31	29	29	6.9	0.0	6.9	9	53	63
	$p(5)o(3,2,4,4,5)m(4)l(6)$	30	30	30	0.0	0.0	0.0	11	62	79
	$p(5)o(5,2,4,7,4)m(2)l(4)$	46	45	43	7.0	4.7	2.2	12	67	135
	$p(6)o(6,4,2,3,5,3)m(3)l(4)$	39	38	38	2.6	0.0	2.6	14	68	280
	$p(6)o(5,2,3,2,5,6)m(5)l(5)$	32	31	30	6.7	3.3	3.2	17	80	580
	$p(7)o(2,3,4,5,5,6,6)m(2)l(4)$	50	48	45	11.1	6.7	4.2	25	98	1099
	$p(7)o(4,2,4,5,3,6,5)m(4)l(2)$	48	48	43	11.6	7.0	0.0	30	115	1900
	$p(8)o(2,8,3,3,4,6,5,5)m(5)l(6)$	49	47	47	4.3	0.0	4.3	36	125	3099
	$p(8)o(3,1,2,8,3,4,4,5)m(4)l(4)$	58	55	53	9.4	3.8	5.5	50	188	6570
	Mean				**3.7**	**1.6**	**1.8**			
Large-size problems	$p(9)o(3,6,4,4,5,6,8,5,6)m(4)l(4)$	70	69	NA[a]	–	–	1.4	68	263	>>
	$p(10)o(6,2,4,4,5,6,7,5,8,4)m(4)l(4)$	83	80	NA	–	–	3.8	81	348	>>
	$p(10)o(5,4,5,6,5,6,4,7,8,5)m(4)l(4)$	90	86	NA	–	–	4.7	86	373	>>
	$p(11)o(3,4,5,5,5,6,6,7,7,6,5)m(4)l(4)$	111	108	NA	–	–	2.8	94	401	>>
	$p(11)o(6,3,4,4,5,6,8,5,6,7,4)m(4)l(4)$	119	110	NA	–	–	8.2	111	450	>>
	$p(12)o(4,2,3,4,5,6,5,7,8,5,4,5)m(4)l(4)$	139	135	NA	–	–	3.0	154	621	>>
	$p(13)o(3,2,7,8,5,6,7,4,6,8,4,6,5)m(4)l(4)$	159	155	NA	–	–	2.6	198	835	>>
	$p(13)o(3,7,4,4,8,6,7,4,6,8,4,6,7)m(4)l(4)$	172	168	NA	–	–	2.4	225	924	>>
	$p(14)o(5,6,7,4,8,6,7,4,6,8,4,6,7,7,8)m(4)l(4)$	184	180	NA	–	–	2.2	295	1 023	>>
	$p(14)o(6,5,7,8,6,8,7,7,8,8,5,6,7,8,8)m(4)l(4)$	189	183	NA	–	–	3.3	332	1199	>>
	$p(15)o(7,8,7,6,5,6,7,7,8,8,7,7,8,8,6,8)m(4)l(4)$	198	190	NA	–	–	4.2	389	1288	>>
	Mean						**3.5**			

[a] Not Available (i.e. no feasible solution is found).

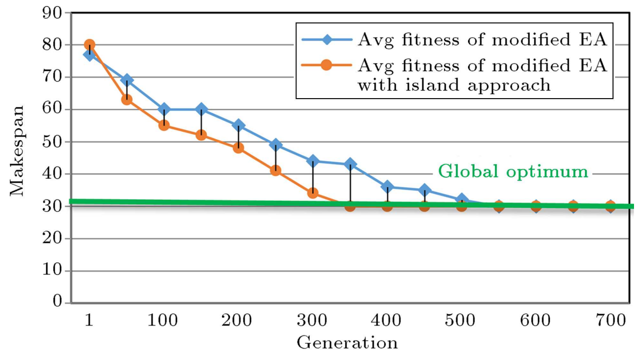

Figure 9. Results obtained by the EA and island model for small-size problems.

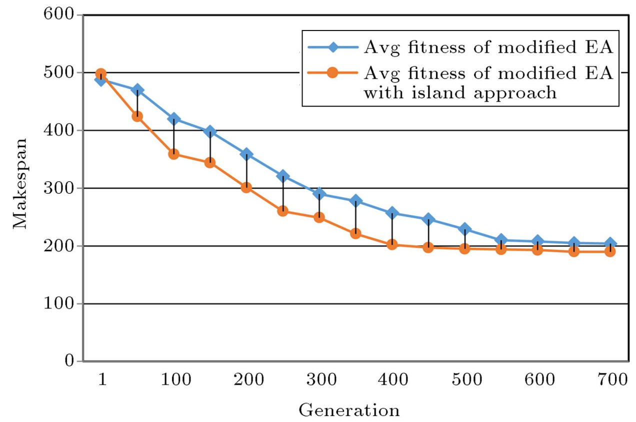

Figure 10. Results obtained by the EA and island model for large-size problems.

and tool movement policy (i.e., dynamic policy). The objective function of this developed model has been to minimize the makespan. Due to an NP-hard nature of the given problem, a modified EA and an Evolutionary Algorithm (EA) including the island model have been developed. The island model benefits from a migration mechanism, which has two essential factors of migration size and migration interval that determine when and how many individuals should be exchanged between islands. In fact, each island is an EA that is run independently. During running, migrations occur, which are similar to evolution in nature. A number of randomly generated problems with different sizes have been produced and solved with a Branch-and-Bound (B&B) method, EA, and island model. The results show that EAs perform well for small-size problems compared to the B&B method in both time and objective function values. For large-size problems, the B&B method could not reach a feasible solution area to the problem and lost its functionality while EAs showed good performance in terms of both time and quality of the derived solutions. The results show that the island model has more efficiency than the EA, which arises from a migration mechanism. For the future research, this problem can be solved with other meta-heuristics and more real situations, such as preventive maintenance and fuzzy parameters.

Acknowledgment

This work is financially supported by Islamic Azad University, Damavand Branch, as research project.

References

1. Prakash, A., Chan, Felix, T.S. and Deshmukh, S.G. "FMS scheduling with knowledge based genetic algorithm approach, expert systems with applications", *Expert Systems with Applications*, **38**, pp. 3161-3171 (2011).
2. Jahromi, M.H.M.A. and Tavakkoli-Moghaddam, R. "A novel 0-1 linear integer programming model for dynamic machine-tool selection and operation allocation in a flexible manufacturing system", *Journal of Manufacturing Systems*, **31**(2), pp. 224-231 (2012).
3. Luque, G. and Alba, E. *Parallel Genetic Algorithm Theory and Real World Applications*, Studies in Computational Intelligence, **367**, Springer (2011).
4. Gordon, V.S., Whitley, D. and Bohn, A. "Dataflow parallelism in genetic algorithms", R. Manner and B. Manderick (Eds.), *Parallel Problem Solving from Nature*, 2 (Brussels 1992), Amsterdam: Elsevier Science, pp. 533-42 (1992).
5. Adamidis, P., *Review of Parallel Genetic Algorithms Technical Report*, Department of Electrical and Computer Engineering, Aristotle University, Thessaloniki (1994).
6. Tosun, U., Dokeroglu, T. and Cosar, A. "A robust island parallel genetic algorithm for the quadratic assignment problem", *International Journal of Production Research*, **51**(14), pp. 4117-4133 (2013).
7. Chen, J. and Ho, S. "A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm", *International Journal of Machine Tools & Manufacture*, **45**, pp. 949-57 (2005).
8. Chan, F.T.S., Swarnkar, R. and Tiwari, M.K. "Fuzzy goal-programming model with an artificial immune system (AIS) approach for a machine-tool selection and operation allocation problem in a flexible manufacturing system", *International Journal of Production Research*, **43**(19), pp. 4147-63 (2005).
9. Chan, F.T.S. and Swarnkar, R. "Ant colony optimization approach to a goal programming model for a machine tool selection and operation allocation problem in an FMS", *Robotics and Computer-Integrated Manufacturing*, **22**, pp. 353-62 (2006).
10. Lee, C.S., Kim, S.S. and Choi, J.S. "Operation sequence and tool selection in flexible manufacturing system under dynamic tool allocation", *Computer & Industrial Engineering*, **45**, pp. 61-73 (2003).

11. Swarnker, R. and Tiwari, M.K. "Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing based heuristic approach", *Robotics and Computer-Integrated Manufacturing*, **20**, pp. 199-209 (2004).

12. Nagarjuna, N., Mahesh, O. and Rajagopal, K. "A heuristic based on multi-stage programming approach for machine-loading problem in a flexible manufacturing system", *Robotics and Computer-Integrated Manufacturing*, **22**, pp. 342-52 (2006).

13. Mahdavi, I., Jazayeri, A., Jahrom, M.H.M., Jafari, R. and Iranmanesh, H. "P-ACO approach to assignment problem in FMSs", In: *Proceedings of Word Academy of Science, Engineering and Technology*, **42**, pp. 196-203 (2008).

14. Sadrzadeh, A. "Knowledge-based genetic algorithm for dynamic machine-tool selection and operation allocation", *Arabian Journal for Science and Engineering*, **39**, pp. 4315-4323 (2014).

15. Abou Gamila, M. and Motavalli, S. "A modeling technique for loading and scheduling problems in FMS", *Robotics and Computer-Integrated Manufacturing*, **19**, pp. 45-54 (2003).

16. Persi, P., Ukovich, W., Pesenti, R. and Nicolich, M. "A hierarchic approach to production planning and scheduling of a flexible manufacturing system", *Robotics and Computer-Integrated Manufacturing*, **15**, pp. 373-85 (1999).

17. Low, C., Yip, Y. and Wu, T. "Modeling and heuristics of FMS scheduling with multiple objectives", *Computers & Operations Research,* **33**, pp. 674-694 (2006).

18. Erhan Kesen, S., Das, S.K. and Gungor, Z. "A genetic algorithm based heuristic for scheduling of virtual manufacturing cells (VMCs)", *Computers & Operations Research*, **37**, pp. 1148-1156 (2010).

19. Kazerooni, A., Chan, F.T.S. and Abhary, K. "A fuzzy integrated decision-making support system for scheduling of FMS using simulation", *International Journal of Computer Integrated Manufacturing Systems*, **10**(1), pp. 27-34 (1997).

20. Chan, F.T.S., Chan, H.K. and Lau, H.C.W. "The state of the art in simulation study on FMS scheduling: A comprehensive survey", *International Journal of Advanced Manufacturing Technology*, **19**(11), pp. 830-849 (2002).

21. Chan, F.T.S. and Chan, H.K. "A comprehensive survey and future trend of simulation study on FMS scheduling", *Journal of Intelligent Manufacturing*, **15**(1), pp. 87-102 (2004).

22. Chan, F.T.S. and Chan, H.K. "Analysis of dynamic control strategies of an FMS under different scenarios", *Robotics and Computer Integrated Manufacturing*, **20**(5), pp. 423-437 (2004).

23. Chan, F.T.S., Chung, S.H. and Chan, L.Y. "An introduction of dominant genes in genetic algorithm for FMS", *International Journal of Production Research*, **46**(16), pp. 4369-4389 (2008).

24. Chan, F.T.S., Chung, S.H., Chan, L.Y., Finke, G. and Tiwari, M.K. "Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach", *Robotics and Computer-Integrated Manufacturing*, **22**, pp. 493-504 (2006).

25. Chan, F.T.S., Bhagwat, R. and Wadhwa, S. "Flexibility performance: Taguchi's method study of physical system and operating control parameters of FMS", *Robotics and Computer Integrated Manufacturing*, **23**(1), pp. 25-37 (2007).

26. Chan, F.T.S., Bhagwat, R. and Wadhwa, S. "Comparative performance analysis of a flexible manufacturing system (FMS): A review-period-based control", *International Journal of Production Research*, **46**(1), pp. 1-24 (2008).

27. Alba, E. and Tomassini, M. "Parallelism and evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, **6**(5), pp. 443-462 (2002).

28. Eiben, A.E. and Smith, J.E. "Introduction to evolutionary computing", *Natural Computing Series*, Springer (2003).

29. Bate, S.T. and Jones, B. "A review of uniform crossover designs", *Journal of Statistical Planning and Inference*, **138**(2), pp. 336-351 (2008).

30. Hu, X.B. and Paolo, E.D. "An efficient genetic algorithm with uniform crossover for air traffic control", *Computers & Operations Research*, **36**(1), pp. 245-259 (2009).

31. 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).

32. Whitley, D., Rana, S. and Heckendor, R.B. "The island model genetic algorithm: On separability, population size and convergence", *Journal of Computing and Information Technology*, **7**, pp. 33-47 (1999).

Biographies

Meghdad Haji Mohammad Ali Jahromi is a Faculty Member at Islamic Azad University, Damavand Branch. He is Assistant Professor in Industrial Engineering Department. His research interests are facilities planning, meta-heuristics, and manufacturing systems. He has published a great number of papers in this area.

Reza Tavakkoli-Moghaddam is Professor of Industrial Engineering in College of Engineering at University of Tehran, Iran. He obtained his BSc in 1989, MSc in 1994, and PhD in 1998 from Iran University of Science and Technology in Tehran, University of Melbourne in Melbourne, and Swinburne University of

Technology in Melbourne, respectively. He serves as the Editor-in-Chief of two journals and the Editorial Board member of five journals. He was the recipient of the 2009 and 2011 Distinguished Researcher Awards as well as the 2010 and 2014 Distinguished Applied Research Awards by University of Tehran, Iran. He was also selected as National Iranian Distinguished Researcher in 2008 and 2010 by the Ministry of Science, Research, and Technology (MSRT) in Iran. He obtained the outstanding rank of being among the top 1% scientists and researchers in the world elite group, reported by Thomson Reuters in 2014. Professor Tavakkoli-Moghaddam has published 4 books, 17 book chapters, and more than 700 papers in reputable academic journals and conferences.

Ahmad Makui is the Associate Professor of Industrial Engineering at Iran University of Science and Technology in Tehran. He was born in Iran in 1960. He obtained his BSc degree in Industrial Engineering from Iran University of Science and Technology. He achieved his MSc degree in Industrial and Systems Engineering in 1991. His PhD thesis was titled "Development of an Approach in Production Planning based on TOC" under the Prof. Aryanezhad's supervision and obtained his degree in 2000. His research interests are in decision analysis and optimization of manufacturing systems. He has published more than 60 papers in the field of Industrial Engineering.

Abbas Saghaei is the Associate Professor of Industrial Engineering at Islamic Azad University, Science and Research Branch in Tehran. He received his PhD degree in Industrial Engineering from Iran University of Science and Technology. He also holds the BS and MS degrees in Industrial Engineering. His research interests include statistical learning and process optimization.