International Journal of Industrial Engineering & Production Research (2016)

June 2016, Volume 27, Number 2
pp. 121-139

http://IJIEPR.iust.ac.ir/

Formation of Manufacturing Cell Using Queuing Theory and Considering Reliability

Bahman Esmailnezhad & Parviz Fattahi*

Bahman Esmailnezhad, *MSc Student, Industrial engineering, Bu-Ali Sina University, Hamedan, Iran*
Parviz Fattahi, *Associate Professor, Faculty of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran*

KEYWORDS

Cell formation problem,
Queuing theory,
Particle swarm optimization algorithm,
Genetic algorithm,
Reliability

ABSTRACT

In this paper, a stochastic cell formation problem is studied considering the queuing theory and the reliability concept. A novel probabilistic mathematical model is presented considering inter-arrival times, processing times, and machines' breakdown. Since the cell formation problem is NP-Hard, two algorithms are developed based on genetic and modified particle swarm optimization (MPSO) algorithms. Since the structure of the problem contains multiple irregularities, a new heuristic method is developed, which produces effective feasible solutions on demand. A deterministic branch and bound (B&B) algorithm is used to evaluate the results of modified particle swarm optimization algorithm and the genetic algorithm. The results indicate that proposed algorithms have better performance than the B&B algorithm of Lingo software according to the mixed effect of solution quality and computational times. The solution of two metaheurstic algorithms is compared by the t-test method. Ultimately, the results of numerical examples indicate that considering reliability has significant effect on the block structures of machine-part.

1. Introduction

The concept of group technology (GT) has emerged to reduce setups, batch sizes, and travel distances. In essence, GT tries to retain the flexibility of a job shop with the high productivity of a flow shop. Cellular manufacturing (CM) concept is based on GT and classifies parts with closest features and processes into the part families and assigns machines into the cells. Machinery and machine tools, agricultural and construction equipment, hospital and medical equipment, defense products, automobiles and engines, piece parts and components, electronic products, chemical equipment, and packaging industries are some domains of CM application in the industry [1]. The design of CM involves four important stages: cell formation (CF), group layout, group

* **Corresponding author: Parviz Fattahi**
Email: *fattahi@basu.ac.ir*
Received 27 July 2013; revised 11 June 2016; accepted 3 December 2016

scheduling, and resource allocation. CF groups machines and parts into manufacturing cells.

Studies considering uncertainty can be categorized into three approaches: stochastic programming approach, fuzzy programming approach, and robust optimization approach. In this article, stochastic programming approach for CF problem is presented. The inter-arrival time between two consecutive customers and service time are considered stochastic. In literature review carried out for CM with stochastic programming approach, the demand, the processing time, the reliability, and the mix product had been considered stochastic, which the following studies taken in these contexts are presented, respectively.

Harhalakis et al. [2] assumed the product demand as a random variable, in CF problem. They sought out the minimization of expected inter-cell material handling cost in their model. Asgharpour and Javadian [3] considered three normal, binomial, and beta distributions for demand and minimized the total sum of the machine purchase cost, the operating cost, the inter-cell and intra-cell material handling costs, the machine relocation cost, and the absolute sum of the demand deviation from mean for part types over the planning horizon. CAO and Chen [4] offered the CF with supposed scenarios for products demand. In this model, an occurrence probability had been assigned to each scenario. Objective function of this model minimized machine cost and expected inter-cell material handling cost. Tavakkoli-Moghaddam et al. [5] examined a mathematical model to solve a facility layout problem in CM systems with stochastic demands. The main purpose of their study is to minimize the total costs of inter- and intra-cell movements in both machine and cell layout problems in CM system simultaneously. They considered part demands as an independent variable with the normal probability distribution. Egilmez and Suer [6] proposed a two-phase hierarchical methodology to find the optimal manpower assignment and cell loads simultaneously. In the first phase, the manufacture cells are formed with objective function of the production rate maximization. Then, manpower with objective function of the number of labors minimization has been assigned to the manufacture cells. In both models, the processing time and demand have a normal distribution. Ariafar et al. [7] purposed the model for layout cells in the shop and machines in the machine cells. Demand has been considered as stochastic and with the uniform distribution. This model minimizes the inter-cell and intra-cell material handling costs. Egilmez et al. [8] viewed uncertainty of processing times and customer demand with a normal distribution. The objective is to design a CM system with product families that are formed with most similar products and minimum number of cells and machines for a specified risk level. Ariafar et al. [9] examined the effect of demand fluctuation on cell layout in shop and machine layout in cell. This model minimizes the inter-cell and intra-cell material handling costs. They assumed which demand has a normal distribution. Rabbani et al. [10] proposed a bi-objective CF problem with demand of products expressed in a number of probabilistic scenarios. Their model in the first objective minimizes the sum of machine constant cost, expected machine variable cost, cell fixed-charge cost, and expected intercell movement cost; in the second objective, it minimizes expected total cell loading variation. Egilmez and Suer [11] offered two models for analyzing the interaction between CF stage and cell scheduling stage in terms of the risk taken by decision-makers. The first model formed manufacturing cells with the objective of maximizing the total pair-wise similarity among products assigned to cells and minimizing the total number of cells. The second model maximizes the number of early jobs. The demand and the processing time in both models are random variables with a normal distribution.

A review of studies done in the processing time area is provided in the following. Saidi-Mehrabad and Ghezavati [12] assumed the processing time and the time between two successive arrivals to cell described by exponential distribution in CF problem. For analyzing this problem, they used queuing theory in which the server is the machine and the customer is the part. The aim of this model is to minimize the summation of three costs: (1) the idleness costs for machines; (2) the total cost of sub-contracting for exceptional elements (exceptional elements are defined as parts which must be processed in different cells and therefore they have intercellular movements); (3) the cost of resource underutilization. Ghezavati and Saidi-Mehrabad [13] proposed a mathematical model for CM problem integrated with group scheduling in an uncertain space. Within this model, CF and scheduling decisions are optimized concurrently. It is assumed that the processing time of parts on machines is stochastic and described by discrete scenarios. Their model minimizes total expected cost including maximum tardiness cost among all parts, cost of subcontracting for exceptional elements and the cost of resource underutilization. Egilmez and Suer [14] presented a mathematical model for CF which minimized the number of tardy jobs and total probability of tardiness. They assumed that the processing time of each job, has a normal distribution. Ghezavati and Saidi-Mehrabad [15] assumed that each machine works as a server and each part is a customer where servers should provide service customers. Accordingly, they defined formed cells as a queue system which can be optimized by queuing theory. The optimal cells and part families were formed by maximizing the probability that a server is busy. Ghezavati [16] evaluated CF problem, scheduling, and layout decisions, concurrently. Also, he considered processing time as stochastic with discrete scenarios under supply chain characteristics. This model minimized holding cost and the costs regarded associated with the suppliers' network in a supply chain in order to outsource exceptional operations. Fardis et al. [17] examined CF problem while considering stochastic parameters, the arrival rate of parts into cells, and machine service rate which have been described by exponential distribution. The objective function of the presented model minimized summation of machines' cost of idleness, sub-contracting cost for exceptional parts, non-utilizing machine cost, and holding cost of parts in the cells.

However, the reliability of machine can impact on the processing time. But, due to the numerous of articles in this issue, they were presented separately. Das et al. [18] presented a multi-objective mixed integer-programming model which interval between failures is distributed exponentially. In the first objective, it minimized the variable cost of machining operations, the inter-cell material handling costs, and the penalty cost of machine non-utilization; in the second objective, it maximized system reliability with minimizing failure rate. Das et al. [19] proposed a preventive maintenance planning model for the performance improvement of CM systems in terms of machine reliability and resource utilization. Considering machine failure times following a Weibull distribution, the presented model in their study determines a preventive maintenance interval and a schedule for performing preventive maintenance actions on each machine in the cell by minimizing the total maintenance cost and the overall probability of machine failures. Das et al. [20] investigated a new approach for the design of CM system by considering machine reliability within a multi-objective optimization framework which seeks to strike a balance between the costs and reliability goals. The CM system design problem consists of assigning the machines to cells, and selecting for each part type, the process route with the highest overall system reliability for each part type while minimizing the total costs of manufacturing operations, machine under-utilization, and inter-cell material handling. It has assumed that machine failure and repair times are exponentially distributed. In another model, based on Weibull distribution and exponential distribution approach, as Das [21] suggested, designer/user selects the suitable failure rate for a specific situation. In this article, when system reliability expectation is high, the Weibull distribution may be viewed to generate better cell configuration. Jabal Ameli et al. [22] investigated the effects of machine breakdowns in the CF problem with a new perspective. The results of their study showed that although considering machine reliability can increase the movement costs, it significantly reduces the total costs and total time for the CM system. Jabal Ameli and Arkat [23] conducted a study on the configuration of machine cells considering production volumes and process sequences of parts. Further, they studied on alternative process routings for part types and machine reliability considerations. They found out that the reliability consideration has significant impacts on the final block diagonal form of machine-part matrices. Chung et al. [24] found that machine reliability has meaningful effects on reducing the total system cost in CF problem. Rafiee e al. [25] proposed the integrated approach to analyze the CM system better, since different aspects of the manufacturing system are interrelated. Weibull distribution is assigned to machine failure time distribution; to conquer the breakdowns, preventive and corrective actions were considered. Arkat et al. [26] presented CF problem in general state while considering the reliability. The generalized CF problem follows the selection of the best process plan for each part and assigning of machines to the cells. In this model, it is assumed that the number of breakdowns for each machine follows a Poisson distribution with a known failure rate. Because of the probabilistic nature of the machine breakdowns, a set of chance constraints have been introduced. These constraints guarantee that the number of breakdowns for each machine never exceeds a predefined percentile. The objective function of this model minimized Intercellular and intracellular movement costs and machine breakdown costs.

Every factory looks for the best mix products as they are a set of part types in a factory, which can be produced. The value of demand for each product in the mix product is not known exactly at the time of designing the manufacturing cells due to customized products, shorter product life-cycles, and unpredictable patterns of demand. The composition of the product mix is determined by demand and is probabilistic in nature. For this reason, in studies done in this area, in the design phase, probability of occurrence is attributed to each possible mix product. Seifoddini [27] proposed a stochastic CF model in which a probability has been attributed to each mix product. He calculated the expected intercellular material handling cost for each machine cell arrangement under all the possible product mixes. Madhusudanan Pillai and Chandrasekharan [28] evaluated manufacturing CF under probabilistic product mix. Each product mix is specified with a scenario to which probability of occurrence has been attributed. They minimized inter-cell material handling. Jayakumar and Raju [29] presented a mathematic model for CF problem in which probability of occurrence has been attributed to each scenario. The objective function of this model is to minimize the total of the machine constant (investment) cost, the operating cost, the inter-cell material handling cost, and the intra-cell material handling cost for a particular product mix. The literature review shows that the stochastic processing time, the stochastic time between two successive arrivals to cell, and the reliability have not been studied, simultaneously. As is known, the demand, the machine availability, and the processing time are uncertain in real world and are changed randomly during the time horizon. In this paper, these stochastic parameters are investigated, simultaneously. The remainder of this paper is organized as follows. The problem formulation is described in Section 2. Modified particle swarm optimization (MPSO) algorithm and genetic algorithm (GA) are described in Sections 3. The computational results and conclusion are reported in Sections 4 and 5, respectively.

2. Problem Formulation

The presented mathematical model is a developed model of deterministic mathematical model. Deterministic state of this model has been studied by several researchers (Boctor [30], Duran et al. [31], and Sayadi et al. [32]). In this section, a new mathematical model with stochastic in which the processing time, the time of arrival parts into the cells, and the machine availability are presented. The assumptions of the proposed model are as follows:

- The inter-arrival time between two consecutive parts is described by exponential distribution with the rate λ_i for each part.
- Processing time for parts follows a general distribution with the rate μ_j for each machine.
- Each machine or server works as M/G/1 queuing system.
- The service discipline is based on first-come, first-serve.
- The breakdown time for each machine follows a general distribution with known mean time to repair and known mean time between failures.
- Capacity of cells for locating machines is known.
- Exceptional elements will be out-sourced to operate.

To formulate the problem, a queuing model is used. In queuing model, the parts are considered as customer and machines as servers. An M/G/1 model is used in this queuing model. The M/G/1 queue is a queue model where arrivals are Markovian (modulated by a Poisson process); service times have a General distribution and there is a single server. In the M/G/1 model, when those entities that are lost are included, the output stream is Poisson. This assumption is supported by several empirical results, to which it has been pointed in the article presented by the Cruz et al. [33]. Because in the presented model, the arrival rate for each queuing system is less than service rate, thus the arrival rate is equal to the output rate and the time of arrival or output parts are exponentially distributed. The queuing system is shown in Fig. 1.

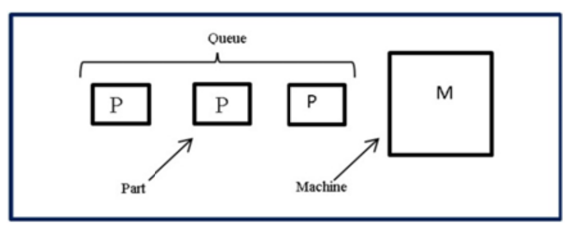

Fig. 1. Queuing system for the proposed model

2-1. Notation

Indexing sets

i: index for parts $i = 1, \ldots, P$

j: index for machines $j = 1, \ldots, M$

k: index for cells $k = 1, \ldots, C$

Parameters

λ_i: mean arrival rate for part i (mean number of parts entered per unit time).

μ_j: mean service rate for machine j (mean number of customers served per unit time by machine j).

M_{max}: The maximum number of machines per cell.

$MTBF_j$: Mean time between failures for machine j

$MTTR_j$: Mean time to repair for machine j

$$a_{ij} = \begin{cases} 1 & \text{if part i is to be processed on machine j} \\ 0 & \text{otherwise.} \end{cases}$$

Decision variables

$$x_{ik} = \begin{cases} 1 & \text{if part i is assigned to cell k} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jk} = \begin{cases} 1 & \text{if machine j is assigned to cell k} \\ 0 & \text{otherwise} \end{cases}$$

2-2. Mathematical model

Approach presented by the Ameli et al. [22] was used for considering the reliability. For investigation of the effect of the reliability on the CF problem, two definitions are presented. The number of machine breakdowns, N (t), can be acquired by dividing the production time by the MTBF.

$$N(t) = \frac{t}{MTBF} \quad (1)$$

By multiplying the MTTR by the number of breakdowns calculated in Eq. (1), the total repair time, T (t), can be obtained as follows:

$$T(t) = \frac{t \times MTTR}{MTBF} \quad (2)$$

In order to obtain the total time for a machine, the repair time for the machine is added to its production time.

$$\text{mean of total time for machine j} = \frac{E_j(t) \times MTTR_j}{MTBF_j} + E_j(t) = \frac{MTTR_j}{\mu_j \times MTBF_j} + \frac{1}{\mu_j} \quad (3)$$

$$\text{mean of total time for machine j} = \frac{1}{\mu_j}\left(\frac{MTTR_j + MTBF_j}{MTBF_j}\right)$$

Where $E_j(t)$ is the production time expectation for machine j. Finally, production rate can be obtained considering the reliability as follows:

$$\text{the production rate for machine j} = \mu_j\left(\frac{MTBF_j}{MTTR_j + MTBF_j}\right) \quad (4)$$

As might be expected, the value of production rate is reduced by considering the reliability. As mentioned in the above contents, the reliability has affects only on the production rate.

According to the queuing model and the Fig. 1, the part arrival time for processing on a particular machine is equivalent to the most minimization of the part arrival time for processing. Because the inter-arrival time between two consecutive customers has exponential distribution, then the most minimization of the part arrival time for processing has exponential distribution with parameter λ_{eff} (effective arrival rate) (Frederick and HillIer [34]). λ_{eff} can be computed as follows:

$$\lambda_{eff} = \sum_{i=1}^{n} \lambda_i$$

Where λ_i is arrival rate for part i, and n is the number of parts that is processed on the same machine. Based on the presented description, the proposed model can be formulated as follows:

$$\text{Min } z = \sum_{k=1}^{C}\sum_{j=1}^{M}\sum_{i=1}^{P} a_{ij}x_{ik}(1 - y_{jk}) \quad (5)$$

$$\text{s.t: } \sum_{k=1}^{C} x_{ik} = 1 \quad \forall i \quad (6)$$

$$\sum_{k=1}^{C} y_{jk} = 1 \quad \forall j \quad (7)$$

$$\sum_{j=1}^{M} y_{jk} \le M_{max} \quad \forall k \quad (8)$$

$$\sum_{k=1}^{C}\sum_{i=1}^{P} \lambda_i a_{ij} x_{ik} y_{jk} < \mu_j\left(\frac{MTBF_j}{MTTR_j + MTBF_j}\right) \quad \forall j \quad (9)$$

$$x_{ik}, y_{jk} \in \{0,1\} \quad \forall i, j, k \quad (10)$$

Constraint (5) minimizes the inter-cell movements of parts. Constraint (6) guarantees that each part must be allocated to one cell only. Constraint (7) guarantees that each machine must be allocated to one cell only. Constraint (8) guarantees that the number of machines to be allocated to any cell should be less than the maximum number of machines allowed in each cell. Constraint (9) avoids instability of queuing system, that is, the effective arrival rate will necessarily be less than service rate. Constraint (10) specifies the type of decision variables.

In the proposed mathematical model, the objective function (5) and constraint (9) are nonlinear. For linearization, new binary integer variable V_{ijk} is defined which is computed by the following equation:

$$V_{ijk} = x_{ik} \times y_{jk} \quad \forall i, j, k \quad (11)$$

For linearization, the objective function (5) and constraint (9) in the following equations should be added to the proposed model by enforcing these two linear inequalities simultaneously:

$$V_{ijk} - x_{ik} - y_{jk} + 1.5 \geq 0 \qquad \forall i, \quad (12)$$

$$1.5V_{ijk} - x_{ik} - y_{jk} \leq 0 \qquad \forall i, j, k \quad (13)$$

3. The Proposed Algorithms

The CF problem is NP-hard problem (King and Nakornchai [35]). Therefore, precise solution procedures and commercial optimization software are unable to reach global optimum in an acceptable amount of time for medium- and large-sized scale problems. To deal with this deficiency, two algorithms based on MPSO and GA metaheuristics have been developed in this paper.

3-1. The MPSO algorithm

Particle swarm optimization (PSO) algorithm by Kennedy and Eberhart (Kennedy and Eberhart [36]; Eberhart and Kennedy [37]) has been presented for problems which have continuous solution space. PSO is a nature-based evolutionary algorithm and starts with an initial population of random solutions. Each potential solution is called a particle ($\vec{x}$). Particles move around in a multidimensional search space, and during movement, each particle adjusts its position based on its own past and the experience of neighbor particles. Particle's fitness is compared with its $pbest_i$ (value of the best function resultfor particle i so far). If existing value is better than $pbest_i$, then set $pbest_i$ equals the current value, and p_i equals the current location $\vec{x}_i$ in multidimensional space. for all particles, value of the best function result so far is called gbest, and its location is assigned to p_g. In the following, the proposed PSO algorithm is illustrated.

3-1-1. Particle structure

The particle representation involves two sections: the first section indicates the cells assigned to machines; the second section represents the cells assigned to parts. The particle used for the proposed model is presented in Fig. 2.

	Machine1	Machine2	⋯	Machine M	Part1	Part2	⋯	Part P
the cell number	1	2	⋯	1	3	2	⋯	3

Fig. 2. Sample of particle structure

3-1-2. The proposed generating initial population

To present a qualified initial population, a heuristic method that always produces a feasible solution is proposed. The heuristic method is presented in Fig. 4. In the first step, machines are allocated to cells based on capacity of cells; in the second step, parts are allocated to cells considering constraint (9) for all machines.

3-1-3. Improvement procedure

In this phase, the linearization objective function is used as the fitness function of the MPSO method. The updating process is based on $\vec{x}_i$, $\vec{p}_i$, and $\vec{p}_g$, and it works as follows. In the original PSO process, the velocity of each particle is iteratively adjusted, so that the particle stochastically oscillates around $\vec{p}_i$ and $\vec{p}_g$ locations. In fact, the velocity of a particle must be understood as an ordered set of transformations that operates on a solution. Therefore, in each particle of MPSO algorithm, $(\vec{x}_i - \vec{p}_i)$ and $(\vec{x}_i - \vec{p}_g)$ indicate the necessary movements to modify from the location given by the first term to the location given by the second term of each expression. The difference between $\vec{x}_i$ and $\vec{p}_i$ represents the changes that will be needed to move the particle i from $\vec{x}_i$ to $\vec{p}_i$. If the difference between a given element of $\vec{x}_i$ and $\vec{p}_i$ is not null, it means that the mentioned position is susceptible to change through operations described below.

A new vector P is generated to record the positions, where $\vec{x}_i$ and $\vec{p}_i$ elements are not equal. The vector Q is defined by the same length with vector P. Binary elements for vector Q are randomly generated. In any position of vector Q, if the element is 0, the change is not performed; but if the element is 1, the element of the same position of vector P is selected. This element in the vector P shows the position of vector $\vec{p}_i$ which should be copied in $\vec{x}_i$. Then, the feasibility of constraints (8) and (9) are evaluated. The procedure continues, if it is true; otherwise, the made changes return and the next element of vector P will be tested, which is specified by vector Q (see Fig. 3). A similar process is done to update the new location $\vec{x'}_i$ by $\vec{p}_g$ and to obtain the new location of $\vec{x}_i$. Similarly, the feasibility of constraints (8) and (9) are examined, and

$\mathrm{pbest_i}$ (the best value of each particle) and gbest (the best value of the whole swarm) are calculated by the fitness function. Finally, a criterion for stopping algorithm (maximum number of iterations) is examined. This procedure is repeated for any particle. Flowchart of the MPSO algorithm is presented in Fig. 5.

$$\left.\begin{matrix} x_i = [3\ \ 1\ \ 1\ \ 2\ \ 2] \\ p_i = [2\ \ 1\ \ 2\ \ 3\ \ 2] \end{matrix}\right\} \rightarrow x_i - p_i = [1\ \ 0\ \ -1\ \ -1\ \ 0] \rightarrow \left\{\begin{matrix} P = [1\ \ 3\ \ 4] \\ Q = [1\ \ 0\ \ 1] \end{matrix}\right\} \rightarrow x_i' = [2\ \ 1\ \ 1\ \ 3\ \ 2]$$

Fig. 3. An example of how to conduct the first stage MPSO algorithm

3-2. The proposed genetic algorithm

Genetic algorithm has been derived from natural selection in biology. GA follows some steps to find better solutions. At first, the initial solution population is generated randomly or used by a special heuristic. Then, some members of the generated populations are selected considering evaluation function, which is called fitness function. Members with higher fitness can be selected by the high probability. So, members with less fitness are substituted by the better ones. This procedure is repeated until it reaches a certain number of iterations (Mahdavi et al. [38]). GA chromosome structure for this model is like particle structure for MPSO. The pseudo code of main steps of the proposed GA are as follows:

1. Initial population is generated using the proposed heuristic algorithm (see Fig. 4).
2. The fitness value of a chromosome is calculated by the linearization of objective function.
3. Producing a new population is based on the repetition of the following steps:

3.1. Crossover operator:

3.1.1. Selection of two-parent chromosome in one population is based on the tournament selection method. Tournament selection involves running several "tournaments" among a few individuals chosen (two or three) at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover.

3.1.2. Two parents are selected from the selection population. Then, a number between 1 and M + P (M is the number of machines, and P is the number of parts) is selected. A single crossover point on both parent’s chromosome is selected. All data beyond that point in either chromosome is swapped between the two parent chromosomes. The resulting combinations are the children. After crossover, the feasibility of constraints (8) and (9) are evaluated. The procedure continues if it is true; otherwise, the made change returns.

3.2. The fraction of the initial population is selected stochastically, and then mutations are performed on them. Used mutation alters one array value in a chromosome from its initial state. A number between 1 and M + P is selected. Then, mutation operator of the source (Mahdavi et al. [38]) is used for the mutation. After mutation, the feasibility of constraints (8) and (9) is evaluated. The procedure continues if it is true; otherwise, the made change returns.

4. The size of the next population is the same as the previous one, which is derived from selecting the best solutions by comparing the previous generations and the solutions generated by mutation and crossover operators.
5. Check stopping criteria (number of iterations).
6. If the stopping condition is not met, go to step two.

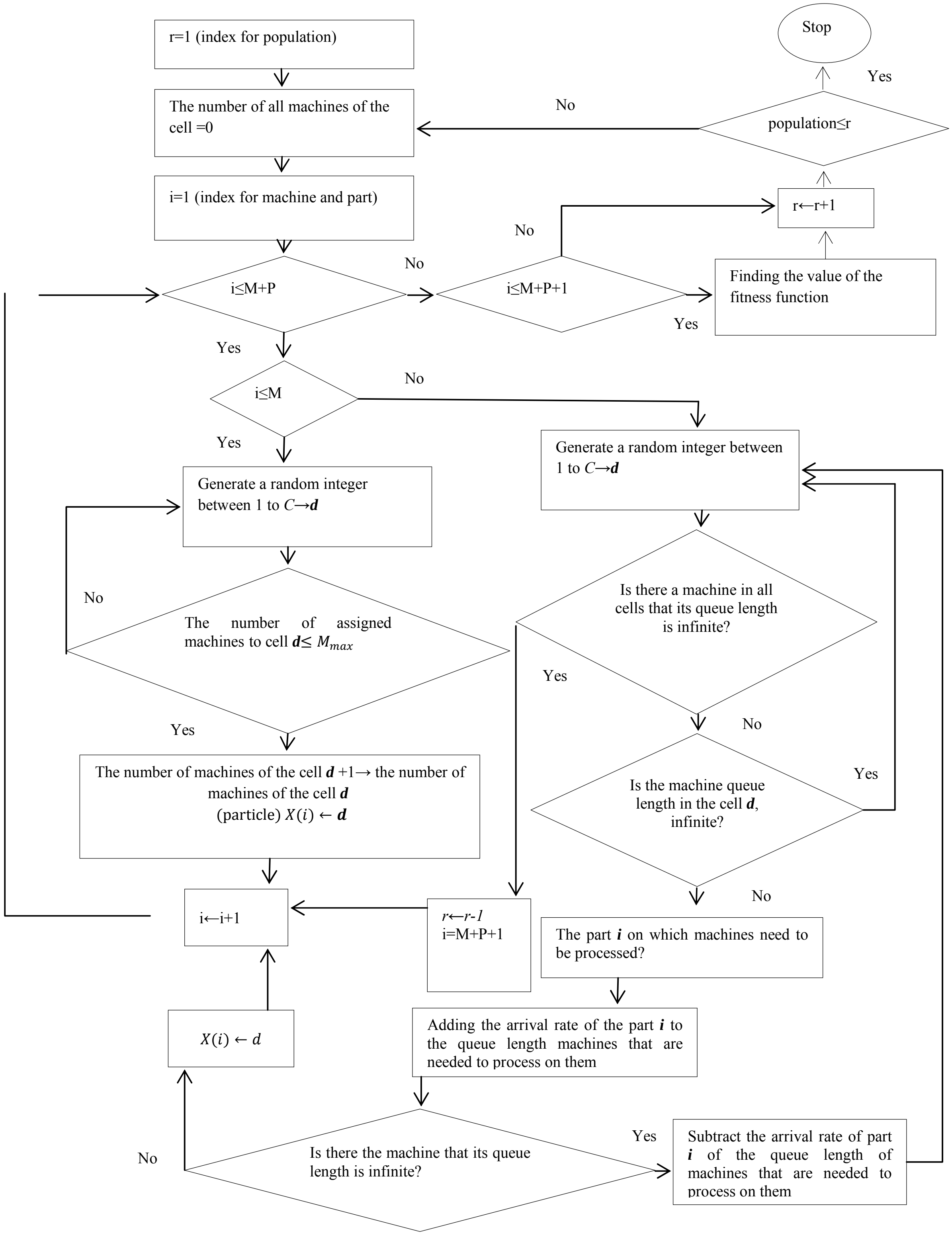

Fig. 4. Heuristic algorithm to generate a feasible initial population

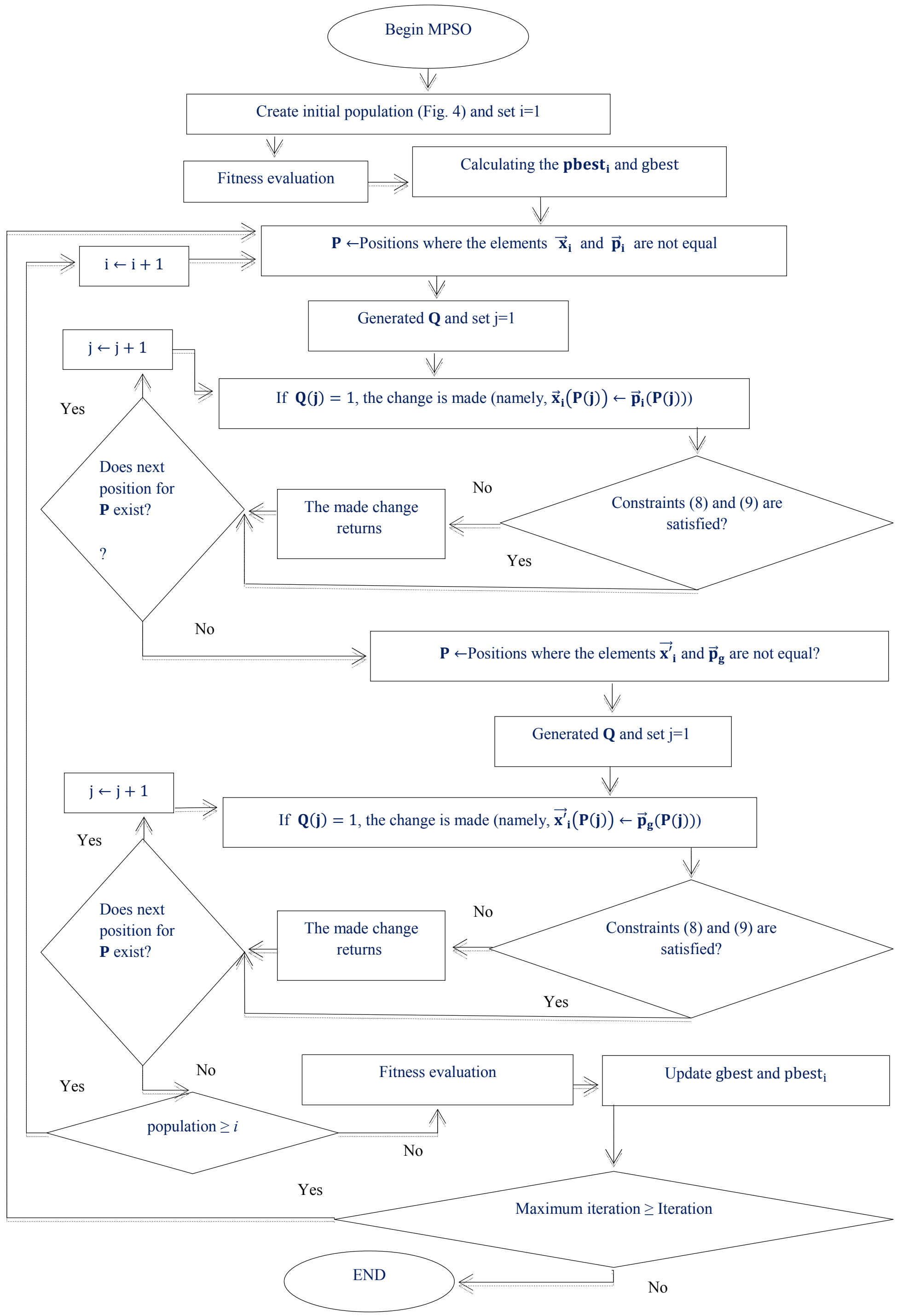

Fig. 5. Flowchart of the MPSO algorithm

4. Computational Results

This section describes some computational experiments which are implemented to evaluate the efficiency and performance of the proposed GA and MPSO algorithms in finding solutions with high quality. For this purpose, 19 sample problems are defined, and then solved by Lingo Software's B&B algorithm, MPSO, and GA. Finally, generated solutions will be compared with each other and with other solutions quality and solving time criteria. The proposed model is coded in LINGO 8.0 optimization software, and the proposed metaheuristic algorithms are coded in MATLAB 2010a on a computer with 2.99 GB RAM and core i3 with 3.1 GHz processor. For each problem, 5400 seconds (1.5 hours) are allowed to run. In B&B algorithm (obtained by Lingo software package), if the problem is solved less than 5400 seconds (1.5 hour), it is categorized as small to medium-sized problems; otherwise, it is categorized as large-sized problems. This procedure is similar to that of Safaei et al. [39].

4-1. Parameters tuning

Since the efficiency of the metaheurstics algorithms depends strongly on the operators and the parameters, design of experiments is done to set parameters. Design of experiments finds the combination of control factors that has the lowest variation, which aims for robustness in solutions. To cover different sizes, instances with small size (8×11), medium size (9×18), and large size (16×30) have been selected. The MPSO and GA parameters are set using a full factorial design and Taguchi technique design, respectively.

4-1-1. Parameter setting for the proposed MPSO

In this section, parameters of the MPSO are selected using the full factorial design. In this study, a full factorial design has been used to determine the best combination of these parameters. The resulting experiment has been analyzed by means of a multifactor analysis of variance (ANOVA) technique. The interactions of the factors are neglected, because their F ratios were small. The response variable for this experiment is the linearization objective function. The experiments are performed for three categories of small, medium, and large instances. Tables 1, 2, and, 3 present the parameters and their candidate values (levels) for small size, medium size, and large size, respectively. Tables 4, 5, and 6 present experimental design ANOVA for small size, medium size, and large size, respectively. In ANOVA table, for each factor or interaction, DF represents the degrees of freedom, SS is the sum of squares, MS is the mean squares, F represents the F-ratio, and finally P represents the p-value. The MPSO parameters are obtained using the main-effects of plot of the presented levels (see figures 6-8). A summary of the two proposed MPSO parameters is given in Table 7.

Tab. 1. Experimental factors and their levels for small size

Parameters	Levels	Values
population	9	{100, 150, 200, 250, 300, 350, 400, 450, 500}
Iteration	11	{35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85}

Tab. 2. Experimental factors and their levels for medium size

Parameters	Levels	Values
population	11	{500, 600, 700, 800, 900, 1000, 1100, 1200, 1300,1400,1500}
Iteration	10	{30, 40, 50, 60, 70, 80, 90, 100, 110, 120}

Tab. 3. Experimental factors and their levels for large size

Parameters	Levels	Values
population	10	{4600, 4700, 4800, 4900, 5000, 5100, 5200, 5300, 5400, 5500}
Iteration	7	{30, 40, 50, 60, 70, 80, 90}

Tab. 4. ANOVA table for MPSO for small size instance

Source	DF	SS	MS	F	P
population	8	26.4714	3.3089	6.42	0.000
iteration	10	6.8350	0.6835	1.33	0.218
Population *iteration	80	34.8620	0.4358	0.85	0.803
Error	198	102.0000	0.5152		
Total	296	170.1684			

Tab. 5. ANOVA table for MPSO for medium size instance

Source	DF	SS	MS	F	P
population	10	20.5576	2.0558	3.04	0.001
iteration	9	10.7303	1.1923	1.76	0.076
Population *iteration	90	64.1697	0.7130	1.06	0.371
Error	220	148.6667	0.6758		
Total	329	244.1242			

Tab. 6. ANOVA table for MPSO for large size instance

Source	DF	SS	MS	F	P
population	9	9.314	1.035	0.88	0.548
iteration	6	13.200	2.200	1.86	0.091
Population *iteration	54	66.419	1.230	1.04	0.416
Error	140	165.333	1.181		
Total	209	254.267			

Tab. 7. The obtained value of MPSO parameters

Parameter \ Size	8×11	9×18	16×30
population	400	1000	5400
Iteration	65	110	90

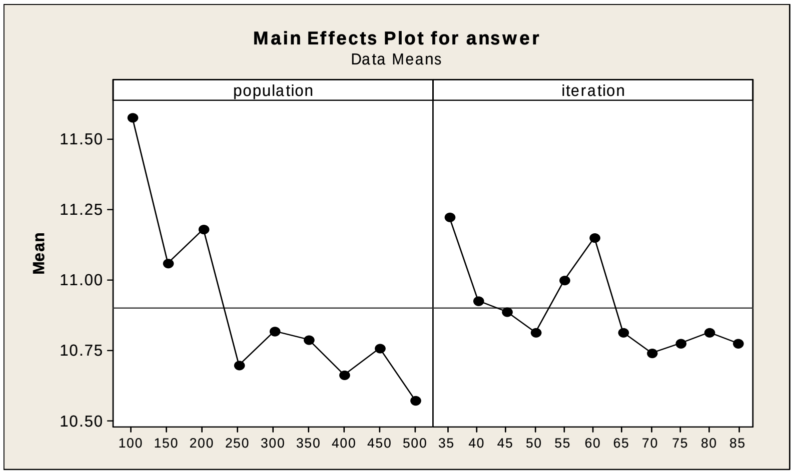

Fig. 6. Main-effects plot for small size

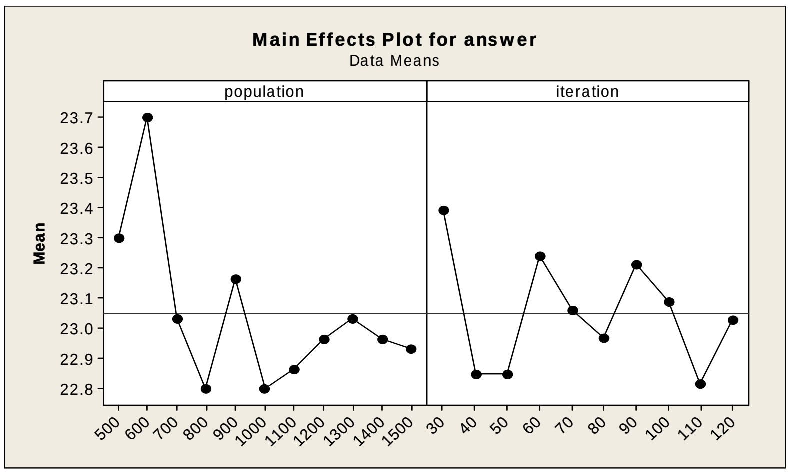

Fig. 7. Main-effects plot for medium size

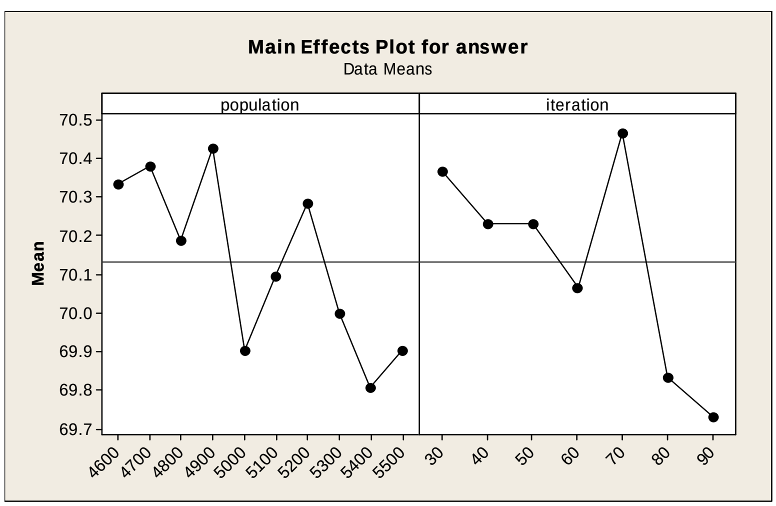

Fig. 8. Main-effects plot for large size

4-1-2. Parameter setting for the proposed GA

Due to the large number of parameters in the proposed GA algorithm, finding the best combination of parameter levels which improves the performance and robustness of the proposed algorithm is important. The best and, of course, the most exhaustive and time-consuming approach is full-factorial. However, in most cases, like this case, this approach is inefficient due to the large number of factors and their respective levels. In the Taguchi method, orthogonal arrays are used to consider a large number of decision variables with a small number of experiments. In comparison to the full factorial methods, the Taguchi method is more efficient, especially in large-scale cases. Five parameters (factors) are considered for the proposed GA. From the Taguchi standard table of orthogonal arrays, the orthogonal array L16 is selected as the fittest orthogonal array design which fulfills the experimental design. The experiments are performed for three categories of small, medium, and large instances. Tables 8, 9, and 10 present the parameters and their candidate values (levels) for small size, medium size, and large size, respectively. After obtaining the results of the Taguchi experiment, the SN ratio for each experiment is calculated. Figures 9, 10, and 11 show the mean SN ratio obtained at each level for the proposed GA. A summary of five proposed GA parameters is given in Table 11.

Tab. 8. Experimental factors and their levels for small size

Parameters	Levels	Values
population	4	{350, 400, 450, 500}
Iteration	4	{55, 60, 65, 70}
Probability of crossover	4	{0.5, 0.6, 0.7, 0.8}
Probability of mutation	4	{0.3, 0.4, 0.5, 0.6}
Number of members competing in the tournament	2	{2, 3}

Tab. 9. Experimental factors and their levels for medium size

Parameters	Levels	Values
population	4	{1000, 1100, 1200, 1300}
Iteration	4	{100, 110, 120, 130}
Probability of crossover	4	{0.8, 0.7, 0.6, 0.5}
Probability of mutation	4	{0.3, 0.4, 0.5, 0.6}
Number of members competing in the tournament	2	{2, 3}

Tab. 10. Experimental factors and their levels for large size

Parameters	Levels	Values
population	4	{5300, 5400, 5500, 5600}
Iteration	4	{80, 90, 100,110}
Probability of crossover	4	{0.5, 0.6, 0.7, 0.8}
Probability of mutation	4	{0.3, 0.4, 0.5, 0.6}
Number of members competing in the tournament	2	{2, 3}

Tab. 11. The obtained value of GA parameters

Size / parameter	8×11	9×18	16×30
Population	350	1100	5500
Iteration	60	110	100
Probability of crossover	0.8	0.7	0.7
Probability of mutation	0.3	0.6	0.4
Number of members competing in the tournament	3	2	3

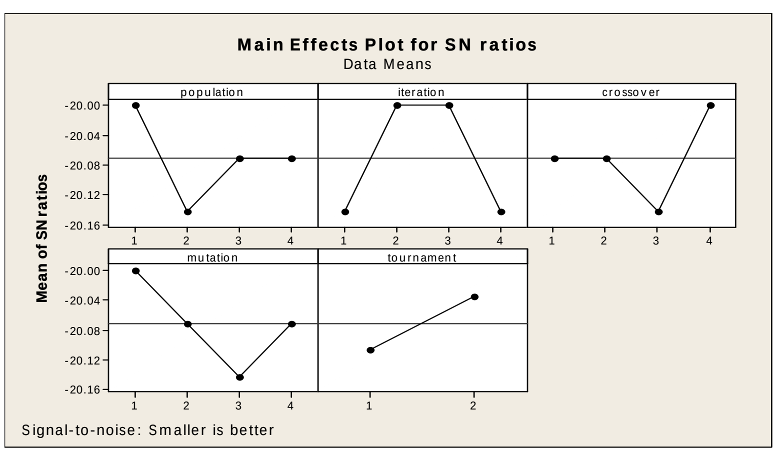

Fig. 9. The mean SN ratio plot for each factor for small-size

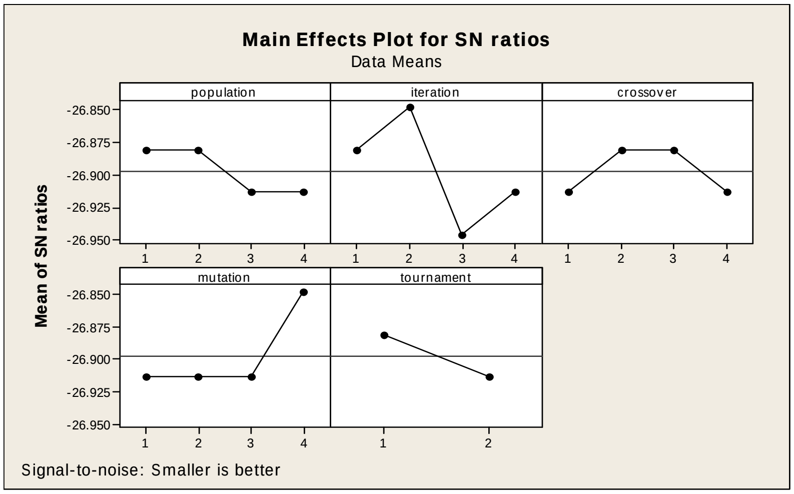

Fig. 10. The mean SN ratio plot for each factor for medium size

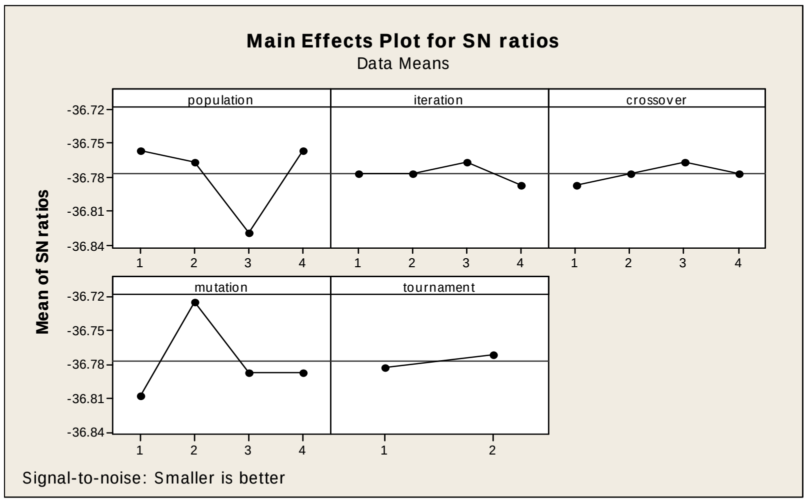

Fig. 11. The mean SN ratio plot for each factor for large-size

According to the Lingo software's results, F_{best} shows the best feasible objective function value (OFV) which has been found so far. F_{bound} indicates the bound on the objective function value. Thus, a possible domain for the optimum value of objective function (F^*) is limited between $F_{best} \geq F^* \geq F_{bound}$. Table 12 and Table 13, indicate the comparison of the Lingo software's B&B algorithm results with MPSO and GA corresponding to 19 test problems for states ignoring reliability and considering reliability, respectively. Each problem is ran 10 times and the average of OFV (Z_{ave}), best OFV (Z_{best}), and also average of run time (T_{MPSO}) are represented in these two tables. The relative gap between the best OFV found by Lingo (F_{best}) and Z_{ave} that is found by metaheuristic algorithms are shown in column ''G_{ave}''. The G_{ave} is calculated as: $G_{ave} = [(Z_{ave} - F_{best})/F_{best}] \times 100$. Also, the relative gap between F_{best} and Z_{best} is shown in column ''G_{best}''. In a similar manner, the G_{best} is calculated as: $G_{best} = [(Z_{best} - F_{best})/F_{best}] \times 100$. In Lingo software's B&B algorithm, if $F_{bound} = F_{best}$, the optimal solution is achieved. In tables, in some cases, Z_{ave} and Z_{best} are between F_{bound} and F_{best} that shows a feasible better solution; under this condition, G_{ave} and G_{best} are negative. But, in cases that Z_{ave} and Z_{best} are out of the domain of $[F_{best}, F_{bound}]$; G_{ave} and G_{best} will be positive numbers. For comparing MPSO and GA, some columns are defined as: Ga-ave, Ga-best, and R that are formulated as follows: $Ga^{ave} = (Z_{ave}^{MPSO} - Z_{ave}^{GA})/Z_{ave}^{MPSO}$, $Ga^{best} = (Z_{best}^{MPSO} - Z_{best}^{GA})/Z_{best}^{MPSO}$, and $R = (T_{MPSO} - T_{GA})/T_{GA}$, respectively.

As mentioned above in small to medium sizes examples, a limited run time (1.5 h) is considered for Lingo solver to find optimal solutions. Therefore, as it can be concluded from Table 12 and Table 13, the percent error of optimal solution is very low when different problems are selected. Also, in large-sized examples, MPSO and GA perform better than the Lingo software's B&B algorithm in some problems in limited time. It implies that MPSO and GA algorithms are so effective to solve the proposed model in all classes of problems. In large-sized problems, metaheuristic algorithms, which have been used, generate better solutions from lingo software's B&B algorithm or solve problems with negligible error. It is obvious that the solving time for metaheuristic algorithms with increasing the size of the problem is much less than Lingo software's B&B algorithm. Paired t test was conducted to analyze significant difference between the obtained solutions of the algorithms in the two states ignoring reliability and considering reliability, respectively. The statistical details are shown in Tables 14 and 15. Tests show that there is a statistical significant difference between solutions obtained by MPSO and GA in both states. By regarding both tables, it can be gathered that the obtained solutions by GA are apparently better than MPSO in both states ignoring reliability and considering reliability.

For comparison of effect of machine failure given in the Tab. 16, the best solutions obtained from three algorithms have been used in both states ignoring reliability and considering reliability. The reliability consideration makes smaller solution space, so the value of the objective function has been deteriorated. Because the objective function of this model is minimized, row of the reduction percent in the Tab. 16.

have has values negative. Optimal solutions of example 7 have been given for evaluated effect of reliability in two states ignoring reliability and considering reliability (see Fig. 13 and Fig. 12). The service capacity of machines decreases in state considering reliability (multiply term $\left(\frac{\mathrm{MTBF_j}}{\mathrm{MTTR_j}+\mathrm{MTBF_j}}\right) < 1$ by term μ_j). Thus, the number of parts, which their operations are outsourced, would increase. The results of solving numerical examples show that the reliability consideration has significant impacts on the final block diagonal form of machine-part matrices.

5. Conclusions

In this paper, a conceptual framework and a mathematical model were proposed as a queue system. Machine as server and part as customer were considered. The inter-arrival time between two consecutive customers had exponential distribution, and service time had been distributed generally. The objective function of model minimized inter-cell movements until it formed optimal cells. Machines are key elements in manufacturing systems and their breakdowns can badly affect system performance measures. The results show that although the reliability consideration has significant impacts on the final block diagonal form of machine-part matrices, it can reduce the value of production rate. As the CF problem is NP-hard, by increasing the size of the problem, Lingo stops solving it; due to the increase of computational time, the B&B algorithm is unable to give good solutions. Therefore, it is necessary to present a metaheuristic approach to solve this model for large-scale problems. Two metaheuristic algorithms based on genetic and MPSO algorithms were developed to solve problems. Finally, solutions generated by MPSO, GA, and Lingo software's B & B were compared with each other by considering solving times. This comparison showed the high efficiency of the proposed metaheuristic algorithms for large-sized problems versus Lingo software's B&B. Solutions generated by GA are apparently better than MPSO algorithm solutions. The following scopes can be interesting fields for future research:

- Development of the model considering facility layout
- Considering costs in the framework of this study
- Development of the model considering dynamic state

Tab. 12. Comparison B&B, MPSO, and GA runs for state ignoring machine reliability

						B&B			MP O						GA				MPSO & GA comparison(%)	
Problem No.	No. of parts	No. of machines	No. of cells	Mmax	Fbest	Fbound	TB&B (s)	Zave	Zbest	TMPSO (s)	Gave (%)	Gbest (%)	Zave	Zbest	TGA (s)	Gave (%)	Gbest (%)	Ga-ave	Ga-best	R
1	4	4	2	3	2	2	0	2	2	0	0.00	0.00	2	2	1	0.00	0.00	0.00	0.00	-100
2	5	5	2	3	0	0	0	0	0	0	0.00	0.00	0	0	1	0.00	0.00	-	-	-100
3	7	6	2	3	1	1	1	1	1	2	0.00	0.00	1	1	1	0.00	0.00	0.00	0.00	100
4	8	6	2	4	7	7	1	8	8	0.9	14.29	14.29	8	8	1	14.29	14.29	0.00	0.00	-10.00
5	9	7	3	4	6	6	1	6.6	6	1	10.00	0.00	6	6	1	0.00	0.00	9.09	0.00	0.00
6	11	8	3	4	10	10	14	10.4	10	1	4.00	0.00	10.3	10	1	3.00	0.00	0.96	0.00	0.00
7	12	9	3	4	16	16	60	17.5	17	6.1	9.38	6.25	17	17	12.5	6.25	6.25	2.86	0.00	-51.20
8	18	8	3	5	19	19	115	20.3	19	4.9	6.84	0.00	19.4	19	13	2.11	0.00	4.43	0.00	-62.31
9	17	10	3	5	20	20	204	21.7	21	5.8	8.50	5.00	21	21	13.6	5.00	5.00	3.23	0.00	-57.35
10	18	9	3	5	22	22	1531	23.1	22	5.3	5.00	0.00	22.2	22	13.1	0.91	0.00	3.90	0.00	-59.54
11	19	9	3	5	28	28	251	30.9	30	8	10.36	7.14	30.1	30	12.7	7.50	7.14	2.59	0.00	-37.01
12	20	9	3	5	27	27	1303	28.8	28	7	6.67	3.70	27.9	27	12.8	3.33	0.00	3.13	3.57	-45.31
13	19	10	3	5	34	34	772	36.2	35	11	6.47	2.94	35.8	34	12.8	5.29	0.00	1.10	2.86	-14.06
14	24	11	4	5	31	28	5400	34.2	32	34.1	10.32	3.23	32.9	32	264.8	6.13	3.23	3.80	0.00	-87.12
15	24	14	4	5	22	19	5400	24.5	23	34.6	11.36	4.55	23	22	279	4.55	0.00	6.12	4.35	-87.60
16	30	16	4	5	68	56	5400	70	68	58.6	2.94	0.00	69	68	267.6	1.47	0.00	1.43	0.00	-78.10
17	35	20	4	7	79	63	5400	78.5	75	62.1	-0.63	-5.06	77.9	75	290.9	-1.39	-5.06	0.76	0.00	-78.65
18	37	20	5	7	108	87	5400	106.3	104	61.2	-1.57	-3.70	106.3	105	308.3	-1.57	-2.78	0.00	-0.96	-80.15
19	43	22	5	7	72	57	5400	74.5	72	64.4	3.47	0.00	73.4	72	342.6	1.94	0.00	1.48	0.00	-81.20
average											5.65	2.02				3.09	1.48	2.49	0.55	-48.93

Tab. 13. Comparison B&B, MPSO, and GA runs for state considering machine reliability

						B&B			MPSO						GA				MPSO & GA comparison(%)	
Problem No.	No. of parts	No. of machines	No. of cells	Mmax	Fbest	Fbound	TB&B(s)	Zave	Zbest	TMPSO(s)	Gave(%)	Gbest(%)	Zave	Zbest	TGA(s)	Gave(%)	Gbest(%)	Ga-ave	Ga-best	R
1	4	4	2	3	2	2	0	2	2	0	0.00	0.00	2	2	1	0.00	0.00	0.00	0.00	-100
2	5	5	2	3	1	1	0	1	1	0	0.00	0.00	1	1	1	0.00	0.00	0.00	0.00	-100
3	7	6	2	3	1	1	0	1	1	2	0.00	0.00	1	1	1	0.00	0.00	0.00	0.00	100.00
4	8	6	2	4	8	8	1	8	8	1	0.00	0.00	8	8	1	0.00	0.00	0.00	0.00	0.00
5	9	7	3	4	7	7	3	7.8	7	1	11.43	0.00	7	7	1	0.00	0.00	10.26	0.00	0.00
6	11	8	3	4	10	10	7	10.8	10	1	8.00	0.00	10	10	1	0.00	0.00	7.41	0.00	0.00
7	12	9	3	4	17	17	35	18	17	1	5.88	0.00	17	17	12.5	0.00	0.00	5.56	0.00	-92.00
8	18	8	3	5	20	20	111	21.5	21	5.6	7.50	5.00	20.4	20	12.9	2.00	0.00	5.12	4.76	-56.59
9	17	10	3	5	21	21	63	23.4	23	1.2	11.43	9.52	23	23	13.1	9.52	9.52	1.71	0.00	-90.84
10	18	9	3	5	23	23	551	24.5	23	6.5	6.52	0.00	23.1	23	12.9	0.43	0.00	5.71	0.00	-49.61
11	19	9	3	5	31	31	128	31.3	31	8.4	0.97	0.00	31	31	12.7	0.00	0.00	0.96	0.00	-33.86
12	20	9	3	5	28	28	415	29.6	29	8	5.71	3.57	28.3	28	12.9	1.07	0.00	4.39	3.45	-37.98
13	19	10	3	5	36	36	1952	37.4	36	10.1	3.89	0.00	37	36	13.1	2.78	0.00	1.07	0.00	-22.90
14	24	11	4	5	33	31	5400	35.8	34	36.3	8.48	3.03	34.4	33	259.6	4.24	0.00	3.91	2.94	-86.02
15	24	14	4	5	25	19.23	5400	26.5	25	37.4	6.00	0.00	25.2	25	277.9	0.80	0.00	4.91	0.00	-86.54
16	30	16	4	5	73	59.59	5400	71.5	68	58.2	-2.05	-6.85	70.9	69	269.6	-2.88	-5.48	0.84	-1.47	-78.41
17	35	20	4	7	79	69	5400	79.9	77	68.8	1.14	-2.53	79.7	77	290.3	0.89	-2.53	0.25	0.00	-76.30
18	37	20	5	7	110	91.03	5400	107.3	104	64.4	-2.45	-5.45	108	106	308.1	-1.82	-3.64	-0.65	-1.92	-79.10
19	43	22	5	7	74	63	5400	75.7	74	65.2	2.30	0.00	75.4	74	335.4	1.89	0.00	0.40	0.00	-80.56
average											3.93	0.33				1.00	-0.11	2.73	0.41	-51.09

Tab. 14. Detailed statistics of paired t test for state ignoring machine reliability

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair1 MPSO-GA	.59474	.47897	.10988	.36388	.82559	5.412	18	.000

Tab. 15. Detailed statistics of paired t test for state considering machine reliability

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair1 MPSO-GA	.55789	.59002	.13536	.27351	.84228	4.122	18	.001

Tab. 16. comparison of effect of machine failure for two states ignoring reliability and considering reliability

Problem number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ignoring machine reliability	2	0	1	7	6	10	16	19	20	22	28	27	34	31	22	68	75	104	72
considering machine reliability	2	1	1	8	7	10	17	20	21	23	31	28	36	33	25	68	77	104	74
Reduction percent	0.00	-	0.00	-14.29	-16.67	0.00	-6.25	-5.26	-5.00	-4.55	-10.71	-3.70	-5.88	-6.45	-13.64	0.00	-2.67	0.00	-2.78

Machine \ Part	7	8	9	1	2	Part		4	5	6	11	12
1	1	0	1	0	1	0	1	0	0	0	0	0
3	1	1	1	0	0	0	0	0	1	0	0	0
6	1	1	1	1	1	1	1	0	1	0	0	0
7	0	1	1	1	0	0	1	0	0	0	0	0
4	0	0	1	0	0	1	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	1	0	1
5	1	0	0	0	0	0	0	0	1	0	1	0
8	1	1	0	1	0	0	0	0	1	0	1	0
9	0	0	0	1	0	0	0	1	0	1	0	1

Fig. 12. Optimal solution of example 7 for state considering machine reliability

Machine \ Part	1	11	3	4	5	Part		8	12	2	9	10
8	1	1	0	0	1	0	1	1	0	0	0	0
2	0	0	0	1	0	1	0	0	1	0	0	0
3	0	0	0	0	1	0	1	1	0	0	1	0
5	0	1	0	0	1	0	1	0	0	0	0	0
9	1	0	0	1	0	1	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	1	1	1
4	0	0	1	0	0	0	0	0	1	0	1	1
6	1	0	1	0	1	0	1	1	0	1	1	1
7	1	0	0	0	0	0	0	1	0	0	1	1

Fig. 13. Optimal solution of example 7 for state ignoring machine reliability

References

[1] Irani SA. Handbook of cellular manufacturing systems. Wiley & Sons, New York, (1999).

[2] Harhalakis G, Ioannou G, Minis I, Nagi R. Manufacturing cell formation under random product demand, International Journal of Production Research, Vol. 32, No. 1, (1994), pp. 47-64.

[3] Asgharpour MJ, Javadian N. Solving a stochastic cellular manufacturing model by using genetic algorithms, International

Journal of Engineering Transactions, Vol. 17, No. 2, (2004), pp. 145-156.

[4] Cao D, Chen M. A robust cell formation approach for varying product demands, International Journal of Production Research, Vol. 43, No. 8, (2005), pp. 1587–1605.

[5] Tavakkoli-Moghaddam R, Javadian N, Javadi B, Safaei N. Design of a facility layout problem in cellular manufacturing systems with stochastic demands, Applied Mathematics and Computation, Vol. 184, (2007), pp. 721-728.

[6] Egilmez G, Suer GA. Stochastic manpower allocation and cell loading in cellular manufacturing systems, the 41st International Conference on Computers & Industrial Engineering, Los Angeles, CA USA, (2011), pp. 193-198.

[7] Ariafar S, Ismail N, Tang SH, Ariffin M, Firoozi Z. A stochastic facility layout model in cellular manufacturing systems, International Journal of the Physical Sciences, Vol. 6, No. 15, (2011), pp. 3666-3670.

[8] Egilmez G, Suer GA, Huang J. Stochastic cellular manufacturing system design subject to maximum acceptable risk level, Computers & Industrial Engineering, Vol. 63, (2012), pp. 842-854.

[9] Ariafar S, Ismail N, Tang SH, Ariffin M, Firoozi Z. The reconfiguration issue of stochastic facility layout design in cellular manufacturing systems, International Journal of Services and Operations Management, Vol. 11, No. 3, (2012), pp. 255-266.

[10] Rabbani M, Jolai F, Manavizadeh N, Radmehr F, Javadi B. Solving a bi-objective cell formation problem with stochastic production quantities by a two-phase fuzzy linear programming approach, International Journal of Advanced Manufacturing Technology, Vol. 58, (2012), pp. 709-722.

[11] Egilmez G, Suer GA. The impact of risk on the integrated cellular design and control, International Journal of Production Research, Vol. 52, No. 5, (2014), pp. 1455-1478.

[12] Saidi-Mehrabad M, Ghezavati VR. Designing Cellular Manufacturing Systems under Uncertainty, Journal of Uncertain Systems, Vol. 3, No. 4, (2009), pp. 315-320.

[13] Ghezavati VR, Saidi-Mehrabad M. Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time, International Journal of Advanced Manufacturing Technology, Vol. 48, (2010), pp. 701-717.

[14] Egilmez G, Suer GA. Stochastic cell loading, family and job sequencing in a cellular manufacturing environment, the 41st International Conference on Computers & Industrial Engineering, Los Angeles, CA USA, (2011), pp. 199-204.

[15] Ghezavati VR, Saidi-Mehrabad M. An efficient hybrid self-learning method for stochastic cellular manufacturing problem: A queuing-based analysis, Expert Systems with Applications, Vol. 38, (2011), pp. 1326-1335.

[16] Ghezavati VR. A new stochastic mixed integer programming to design integrated cellular manufacturing system: A supply chain framework, International Journal of Industrial Engineering Computations, Vol. 2, (2011), pp. 563-574.

[17] Fardis F, Zandi A, Ghezavati VR. Stochastic extension of cellular manufacturing systems: a queuing-based analysis, Journal of Industrial Engineering International, Vol. 9, (2013), pp. 1-8.

[18] Das K, Lashkari RS, Sengupta S. Reliability considerations in the design of cellular manufacturing systems: a simulated annealing based approach, International Journal of Quality & Reliability Management, Vol. 23, No. 7, (2006), pp. 880-904.

[19] Das K, Lashkari RS, Sengupta S. Machine reliability and preventive maintenance planning for cellular manufacturing systems, European Journal of Operational Research, Vol. 183, (2007), pp. 162-180.

[20] Das K, Lashkari RS, Sengupta S. Reliability consideration in the design and analysis of cellular manufacturing systems, International

Journal of Production Economics, Vol. 105, (2007), pp. 243-262.

[21] Das K. A comparative study of exponential distribution vs Weibull distribution in machine reliability analysis in a CMS design, Computers & Industrial Engineering, Vol. 54, (2008), pp. 12-33.

[22] Ameli MSJ, Arkat J, Barzinpour F. Modelling the effects of machine breakdowns in the generalized cell formation problem, International Journal of Advanced Manufacturing Technology, Vol. 39, (2008), pp. 838-850.

[23] Ameli MSJ, Arkat J. Cell formation with alternative process routings and machine reliability consideration, International Journal of Advanced Manufacturing Technology, Vol. 35, (2008), pp. 761-768.

[24] Chung SH, Wu TH, Chang CC. An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations, Computers & Industrial Engineering, Vol. 60, (2011), pp. 7-15.

[25] Rafiee K, Rabbani M, Rafiei H, Rahimi-Vahed A. A new approach towards integrated cell formation and inventory lot sizing in an unreliable cellular manufacturing system, Applied Mathematical Modelling, Vol. 35, (2011), pp. 1810-1819.

[26] Arkat J, Naseri F, Ahmadizar F. A stochastic model for the generalised cell formation problem considering machine reliability, International Journal of Computer Integrated Manufacturing, Vol. 24, No. 12, (2011), pp. 1095-1102.

[27] Seifoddini H. A probabilistic model for machine cell formation, Journal of Manufacturing Systems, Vol. 9, No. 1, (1990), pp. 69-75.

[28] Pillai VM, Chandrasekharan MP. Manufacturing cell formation under probabilistic product mix, Industrial Engineering and Engineering Management (IEEM), Macao, (2010), pp. 580-584.

[29] Jayakumar V, Raju R. A multi-objective genetic algorithm approach to the probabilistic manufacturing cell formation problem, South African Journal of Industrial Engineering, Vol. 22, No. 1, (2011), pp. 199-212.

[30] Boctor FF. A linear formulation of the machine-part cell formation problem, International Journal of Production Research, Vol. 29, No. 2, (1991), pp. 343-356.

[31] Duran O, Rodriguez N, Consalter LA. Collaborative particle swarm optimization with a data mining technique for manufacturing cell design, Expert Systems with Applications, Vol. 37, (2010), pp. 1563-1567.

[32] Sayadi MK, Hafezalkotob A, Naini SGJ. Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation, Journal of Manufacturing Systems, Vol. 32, (2013), pp. 78-84.

[33] Cruz FRB, Van Woensel T, Smith JMG. Buffer and throughput trade-offs in M/G/1/K queueing networks: A bi-criteria approach, International Journal of Production Economics, Vol. 125, (2010), pp. 224-234.

[34] Frederick GJL, HillIer S. Introduction to Operations Research, McGraw-Hill, New York, (2001).

[35] King JR, Nakornchai V. Machine-component group formation in group technology: review and extension, International Journal of Production Research, Vol. 20, (1982), pp. 117-133.

[36] Kennedy J, Eberhart R. Particle swarm optimization, Neural Networks IV, Perth, Australia, (1995), pp. 1942–1948.

[37] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory, The sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, (1995), pp. 39-43.

[38] Mahdavi I, Paydar MM, Solimanpur M, Heidarzade A. Genetic algorithm approach for solving a cell formation problem in cellular manufacturing, Expert Systems with Applications, Vol. 36, (2009), pp. 6598-6604.

[39] Safaei N, Saidi-Mehrabad M, Jabal-Ameli MS. A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, European Journal of Operational Research, Vol. 185, (2008), pp. 563-592.