# Designing Stochastic Cell Formation Problem Using Queuing Theory

Parviz Fattahi[a,*], Bahman Esmailnezhad[b], Amir Saman Kheirkhah[a]

*[a]Associate Professor, Faculty of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran*

*[b]MSc, Faculty of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran*

**Abstract**

This paper presents a new nonlinear mathematical model to solve a cell formation problem which assumes that processing time and inter-arrival time of parts are random variables. In this research, cells are defined as a queue system which will be optimized via queuing theory. In this queue system, each machine is assumed as a server and each part as a customer. The grouping of machines and parts are optimized based on the mean waiting time. For solving exactly, the proposed model is linearized. Since the cell formation problem is NP-Hard, two algorithms based on genetic and modified particle swarm optimization (MPSO) algorithms are developed to solve the problem. For generating of initial solutions in these algorithms, a new heuristic method is developed, which always creates feasible solutions. Also, full factorial and Taguchi methods are used to set the crucial parameters in the solutions procedures. Numerical experiments are used to evaluate the performance of the proposed algorithms. The results of the study show that the proposed algorithms are capable of generating better quality solutions in much less time. Finally, a statistical method is used which confirmed that the MPSO algorithm generates higher quality solutions in comparison with the genetic algorithm (GA).

*Keywords:* Cell formation, Queuing theory, Particle swarm optimization, Branch and bound.

## 1. Introduction

Group Technology (GT) can be defined as a manufacturing philosophy which identifies similar parts and groups them together to take advantage of their similarities in manufacturing and design (Papaioannou and Wilson, 2010). Cellular Manufacturing System (CMS) is an application of GT and has appeared as a promising alternative manufacturing system. CMS could be characterized as a hybrid system linking the advantages of both the jobbing (flexibility) and mass (efficient flow and high production rate) production approaches. There are three basic and significant steps in the design of CMS: Cell formation, intracellular machine layout, and cell layout (Ahi et al., 2009). The design of CMSs has been called cell formation (CF). Given a set of part types, processing requirements, part type demand and available resources (machines, equipment, etc.), a general design of cellular manufacturing comprises the following approaches: (a) part families are formed according to their processing requirements, (b) machines are grouped into manufacturing cells, and (c) part families are assigned to cells (Papaioannou and Wilson, 2010).

Many studies investigate CMS problems in certain conditions, demand, machine availability, processing time, raw materials, and etc., but they are uncertain in real world and are changed randomly during the time horizon. Therefore, cellular manufacturing in uncertainty condition is an important area of investigation and making more accurate decisions. In this paper, uncertainty in processing time and inter-arrival time of parts are considered to fill this gap in the literature. Regardless of fuzzy uncertainty, two types of uncertainties might be introduced. The first is due to our limited knowledge of the input parameters and the second associates with planning for a future. The former leads to stochastic problems and the latter causes dynamic problems. It is assumed that our information of model parameters is incomplete in the static stochastic problems. In other words, the exact value of the parameters is unknown. It can only be predicted with probability; however, parameters are uncertain, static and do not change during time. There are two main approaches to static stochastic problems: scenario planning and the probabilistic approach. The scenario-based planning is an approach in which uncertainty is captured by determining a number of possible future states by decision makers. Each possible future state is called a scenario. The goal is to find solutions performing well under all scenarios. Such a solution is called robust solution or compromise solution. In the probabilistic approach, the probability distribution of random variables is considered explicitly. Some researchers have placed probability distributions in the form of mathematical programming and some have put it in the context of queuing theory.

* Corresponding author Email address: pfattahi@gmail.com

In the following, a brief review of previous studies based on the mentioned framework is presented. In uncertain CF problems, most articles are conducted on the uncertainty in demand, the processing time and the reliability of machines. However, the reliability of machine can have an impact on the processing time, but due to the large number of articles, this issue is investigated separately. Regarding stochastic demand, Arzi et al. (2001) investigated multi-objective approach for the formation problem in a lumpy demand environment. They found out that in lumpy demand the required capacity is influenced by demand variability and there is a correlation between the part types assigned to the cells and it causes machine idleness, or, alternatively, capacity shortage. They described a mixed integer programming model via typical examples and concluded that using traditional approaches designers do not obtain optimal solutions and might make decisions on the basis of wrong results. Tavakkoli-Moghaddam et al. (2007) examined a mathematical model to solve a facility layout problem in CMSs with stochastic demands. The main purpose of their study was to minimize the total costs of inter-cell and intra-cell movements in both machine and cell layout problems in CMS simultaneously. They considered part demand as an independent variable with the normal probability distribution. Egilmez et al. (2012) proposed a non-linear mathematical model to solve the stochastic CMS design problem. In their paper processing times and customer demand were viewed uncertain with the normal distribution. The study was meant to design a CMS with product families that are formed with the most similar products and minimum number of cells and machines for a specified risk level. Egilmez and Suer (2014) 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 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 normal distribution. Concerning the machine reliability, JabalAmeli et al. (2008) investigated the effects of machine breakdowns on 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 CMS. JabalAmeli and Arkat (2008) conducted a study on the configuration of machine cells considering production volumes and process sequences of parts. Further, they studied 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 matrixes. Chung et al. (2011) found that machine reliability has meaningful effects on the reducing of the total system cost. Regarding probability distributions in the form of queuing theory, Kannan and Palocsay (1999) examined cellular versus process layouts based on the impact of learning on shop performance. They modeled mean flow time for the shops, in manufacturing process and cellular by queuing theory and showed a cellular shop needs only achieve a marginally higher learning rate than a job shop in order to perform at a comparable level. Pitchuka et al. (2006) carried out a study to explore whether splitting the part population into part families can offset the effect of partitioning the machine population on queue time. Their study recognizes certain situations where a cellular system without getting benefits in factor settings outperforms a functional system. Ghezavati and Saidi-Mehrabad (2011) assumed that each machine works as a server and each part is a customer where servers should give service to customers. Accordingly, they defined the formed cells as a queue system which can be optimized by queuing theory. By maximizing the probability that a server is busy, the optimal cells and part families formed. Fardis et al. (2013) examined CF problem considering stochastic parameters the arrival rate of parts into cells and machine service rate describing by an exponential distribution. The objective function of presented model minimized the summation of the idleness cost of machines, the subcontracting cost for exceptional parts, non-utilizing machine cost, and the holding cost of parts in the cells. After presenting the research conducted in the static stochastic problems with the probabilistic approach, a review of the investigations conducted on the static stochastic problems with the scenario planning approach is presented. A mathematical model for cellular manufacturing problem integrated with group scheduling in an uncertain space was proposed by Ghezavati and Saidi-Mehrabad (2010). In this model, cell formation 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 the expected cost consisting of maximum tardiness cost among all parts, the cost of subcontracting for exceptional elements and the cost of resource underutilization.

Finally, the studies carried out on dynamic stochastic problems in the frameworks of stochastic demand and the reliability of machine are presented, respectively. Asgharpour and Javadian (2004) presented a mathematical model for designing CMSs based on dynamically generated, stochastic demand, routing flexibility, and machine flexibility. Three distributions normal, binomial and beta for demand considered and the total sum of machine purchase cost, operating cost, inter-cell and intra-cell material handling costs, machine relocation cost and the absolute sum of the demand deviation from mean for part types over the planning horizon minimized. Das et al. (2007) proposed a preventive maintenance planning model for the performance improvement of CMSs in the terms of the machine reliability, and resource utilization. Considering machine failure times follow 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. In another study, Das et al. (2007) examined a new approach to the design of CMS by considering the machine reliability within a multi-objective optimization framework which seeks to strike a balance between the costs and reliability goals. The CMS design problem comprises assigning the machines to cells, and selecting, for each part type, the process route with the highest overall system reliability while minimizing the total costs of manufacturing operations, machine under-utilization, and inter-cell material handling. It has been assumed that machine failure and repair times are exponentially distributed. Das (2008) in another model, based on the Weibull distribution and the exponential distribution approach suggested designer or user to select 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. Rafiee et al. (2011) proposed the integrated approach to analyze better CMS, since different aspects of the manufacturing system are interrelated. The Weibull distribution, assigned to machine failure time distribution and to conquer the breakdowns, preventive and corrective actions had been considered. Table 1 indicates the summery of the presented literature review.

In this paper, a stochastic mathematical programming model for CF in CMS is proposed. The main aim of the proposed model is optimization of the mean waiting time for processing parts on machines to have optimal grouping of machines and parts. However, mathematical models of this type may impose computational difficulties and may not be solvable using commercial optimization software for medium-to-large sized problems. Thus, two algorithms based on particle swarm optimization (PSO) algorithm and the genetic algorithm (GA) is proposed to solve the model. The remainder of this paper is organized as follows. The problem formulation is described in Section 2. The MPSO algorithm and GA are described in Sections 3. The computational results and conclusion are reported in Section 7 and 1, respectively.

## 2. Problem Formulation

As pointed out in the above literature review, one of the solution methods for CF is the mathematical programming. In this section, a new mathematical model is presented in which the processing time and the time of inter-arrival parts into the cells, are considered random variables. To formulate the problem, a queuing model is used. In this queuing model, the part as customer and the machine as server are considered. The M/M/1 model is used in this queuing model. Service discipline is based on first come, first service. In this model, the time between two successive arrival customers and service time is exponentially distributed. The queuing system is shown in Fig. **1**.

Table 1
The summary of the literature review

| Author(s) | stochastic problems | | | |
|---|---|---|---|---|
| Arzi et al. (2001) | demand | Mathematical programming | Probabilistic | Static stochastic problems |
| Tavakkoli-Moghaddam et al. (2007) | | | | |
| Egilmez et al. (2012) | | | | |
| Egilmez, and Suer (2014) | | | | |
| JabalAmeli et al. (2008) | machine reliability | | | |
| JabalAmeli and Arkat (2008) | | | | |
| Chung et al. (2011) | | | | |
| Kannan and Palocsay (1999) | Queuing theory | | | |
| Pitchuka et al. (2006) | | | | |
| Ghezavati and Saidi-Mehrabad (2011) | | | | |
| Fardis et al. (2013) | | | | |
| Ghezavati and SaidiMehrabad (2010) | scenario planning | | | |
| Asgharpour and Javadian (2004) | Demand | Dynamic stochastic problems | | |
| Das et al. (2007) | machine reliability | | | |
| Das et al. (2007) | | | | |
| Das (2008) | | | | |
| Rafiee et al. (2011) | | | | |
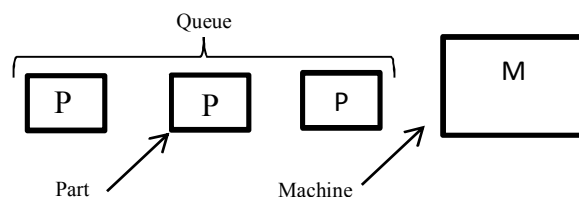


Fig. 1. Queuing system for the proposed model

In this model, the mean waiting time of part processing time is considered to be optimized. Optimizing average waiting time leads to increased number of parts processed in intra-cell and minimal inter-cell transportations of parts. In the optimization of the mean waiting time, the first idea that comes to mind is minimizing the mean waiting time. However, within the cellular manufacturing framework, for forming cells, each part must be assigned only to one cell and each machine also must be assigned only to one cell. This is an important barrier to minimize the mean waiting time.

Minimizing the mean waiting time within cellular manufacturing framework makes parts and machines have a tendency not to be in a cell and thus decreases the waiting time. But it should be noted that this locating will not form the manufacturing cells (see Fig. 2). As a result, maximizing the mean waiting time in the CF problem decreases the number of inter-cell movements and finally leads to the formation of the optimal cells and part families.

Part

|  |  | 2 | 3 | 5 | 1 | 4 |
|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 1 | 1 |
| | 3 | 0 | 0 | 0 | 1 | 0 |
| Machine | 5 | 0 | 0 | 0 | 0 | 1 |
| | 2 | 1 | 1 | 1 | 0 | 0 |
| | 4 | 1 | 1 | 0 | 0 | 0 |

Fig. 2. The forming cells in minimizing average waiting time

According to the queuing model and Fig. 1, the part inter-arrival time for processing on a particular machine is equal to the most minimization of the part arrival time for processing and it has the exponential distribution with parameter $\lambda_{eff}$ (effective arrival rate) (Hillier and Lieberman, 1995). The $\lambda_{eff}$ can be computed as follows:

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

In which $\lambda_i$ is arrival rate for part $i$ and $n$ is a number of parts that are processed on the same machine. Hence, the mean waiting time of part, in order to be processed on the machine j, is calculated as follows:

$$W_j = \frac{1}{\mu_j - \lambda_{eff_j}} \tag{2}$$

Based on the presented description, the proposed model can be formulated as follows:

*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*

$\lambda_i$: mean arrival rate for part i (mean number of parts entered per unit time).
$\mu_j$: 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.
$a_{ij} = \begin{cases} 1 & \text{if part i is to be processed on machine j} \\ 0 & \text{otherwise} \end{cases}$

*Decision variables*

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

*Mathematical model*

$$Max\ Z = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{\mu_j - \sum_{k=1}^{C}\sum_{i=1}^{P} \lambda_i a_{ij} x_{ik} y_{jk}} \tag{3}$$

$$s.t: \quad \sum_{k=1}^{C} x_{ik} = 1 \quad \forall i \tag{4}$$

$$\sum_{k=1}^{C} y_{jk} = 1 \quad \forall j \tag{5}$$

$$\sum_{j=1}^{M} y_{jk} \leq M_{max} \quad \forall k \tag{6}$$

$$\sum_{k=1}^{C}\sum_{i=1}^{P} \lambda_i a_{ij} x_{ik} y_{jk} < \mu_j \quad \forall j \tag{7}$$

$$x_{ik}, y_{jk} \in \{0,1\} \quad \forall i,j,k \tag{8}$$

The objective function (3) maximizes the average of the mean waiting time of parts for processing on different machines. The main point in the objective function is that the arrival rate for each part is added to the effective arrival rate for each machine when part needs to be operated on the machine, and the part and the machine are located in the same cell as well. The objective function increases the waiting time of the parts behind the machines, that is, it increases the parts in the cells as the machines and parts within the same cell are taken into

account in the waiting time. Constraint (4) guarantees that each part must be allocated to one cell only. Constraint (5) guarantees that each machine must be allocated to one cell only. Constraint (6) guarantees that, number of the machines to be allocated to each cell, should be less than the maximum number of the machines allowed in each cell. Constraint (7) avoids instability of the queuing system, that is, the effective arrival rate will be necessarily less than service rate. Note that the effective arrival rate must be less than the service rate; otherwise the objective function is unclear.

### 2.1. *Linearization of the mathematical model*

In the proposed mathematical model, the objective function (3) and constraint (7) are nonlinear. For linearization, new binary integer variable $V_{ijk}$ is defined which is computed by the following equation:

$$V_{ijk} = x_{ik} \times y_{jk} \quad \forall i,j,k \qquad (9)$$

Equivalent of the objective function is given in appendix A and for linearization constraint (7), following equations should be added to the proposed model by enforcing these two linear inequalities simultaneously:

$$V_{ijk} - x_{ik} - y_{jk} + 1.5 \geq 0 \quad \forall i,j,k \qquad (10)$$
$$1.5V_{ijk} - x_{ik} - y_{jk} \leq 0 \quad \forall i,j,k \qquad (11)$$

Overall, it is implied that proposed linearization model seeks to put those kinds of the parts in the cells that have higher arrival rates (larger numbers of them are needed to produce), because the arrival rate of each part as coefficient in the linearization objective function is appeared.

## 3. The Proposed Algorithms

It is well-known that cellular manufacturing problems are NP-hard (King and Nakornchai, 1982). Therefore, precise solution procedures and commercial optimization soft wares are unable to reach global optimum in an acceptable amount of time for medium and large size scale problems. To deal with this deficiency, two algorithms based on MPSO and GA meta-heuristics have been developed in this paper.

### 3.1. *The MPSO algorithm*

#### 3.1.1 *The Original PSO*

Kennedy and Eberhart (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995) introduced PSO as an optimization tool that provides a population-based search method. In PSO, solutions are called as particles that modify their locations in each iteration. Particles fly around in a multidimensional search space, and during

flight, each particle adjusts its location based on its own past, and the experience of neighbor particles. The original process for implementing PSO is as follows:

1: Initialize a population array of particles ($\vec{x}_i$) with random locations and velocities ($\vec{v}_i$) on D-dimensions in the search space.

2: loop

3: For each particle, evaluate the desired optimization fitness function in D variables.

4: Compare particle fitness evaluation with its $pbest_i$ (value of the best function result so far, for particle i). If existing value is better than $pbest_i$, then set $pbest_i$ equal to the current value, and $p_i$ equal to the current location $\vec{x}_i$ in D-dimensional space.

5: Recognize the particle in the neighborhood with the best success so far (gbest), and assign its index to the variable g.

6: Modify the velocity and location of the particle based on the following equation (see notes below):

$$\begin{cases} \vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0, \emptyset_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0, \emptyset_2) \\ \qquad \otimes (\vec{p}_g - \vec{x}_i) \\ \vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \end{cases} \qquad (12)$$

7: If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.

8: end loop

Notes:

– $\vec{U}(0, \emptyset_i)$ indicates the vector of random numbers uniformly distributed in $[0, \emptyset_i]$ which is randomly produced at each iteration and for each particle.

– $\otimes$ is component-wise multiplication.

– In the original version of PSO, each component of $\vec{v}_i$ is kept within the range $[-V_{max}, +V_{max}]$ (Poli et al., 2007).

The original PSO algorithm was developed for continuous domains. A discrete version of the PSO algorithm was developed by Durán et al. (2010). In this paper, a modified PSO algorithm by Duran et al. (2010) is presented. The main modification compared to the original PSO algorithm consists in not using the vector of velocities that the original PSO algorithm does. This algorithm utilizes the concept of proportional likelihood with modifications, a technique applied in data mining applications.

#### 3.1.2 *Particle structure*

The particle representation involves two sections: the first section indicates the cells assigned to machines and the second section represents the cells assigned to parts. The particle used for the proposed model is presented in Fig. **3**.

| | Machine1 | Machine2 | ⋯ | Machine M | Part1 | Part2 | ⋯ | Part P |
|---|---|---|---|---|---|---|---|---|
| number of cells | 1 | 2 | ⋯ | 1 | 3 | 2 | ⋯ | 3 |

Fig. 3. Sample of particle structure

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

### 3.1.3 The proposed generating initial population

To present a qualified initial population, a heuristic method that may always produce a feasible solution is proposed. The heuristic method has been presented in Fig. 4. In the first step, machines are allocated to cells based on capacity of cells and in the second step parts are allocated to cells considering constraint (7) for all machines.

### 3.1.4 Improvement procedure

In this phase, the linearization objectives function is used as the fitness function of the MPSO algorithm. 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, each particle of MPSO algorithm, ( $\vec{x}_i - \vec{p}_i$) and ($\vec{x}_i - \vec{p}_g$) indicates the necessary movements to modify 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 the $\vec{x}_i$ and $\vec{p}_i$ elements are not equal. The vector Q, is defined with the same length with the vector P. Binary elements for the vector Q is randomly generated. In any position of the 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 the 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 (6) and (7) is evaluated. The procedure continues, if it is true, otherwise, the applied changes return and the next element of the vector P will be tested, which is specified by the vector Q (see Fig. 5). 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 (6) and (7) are examined, and 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. 6.
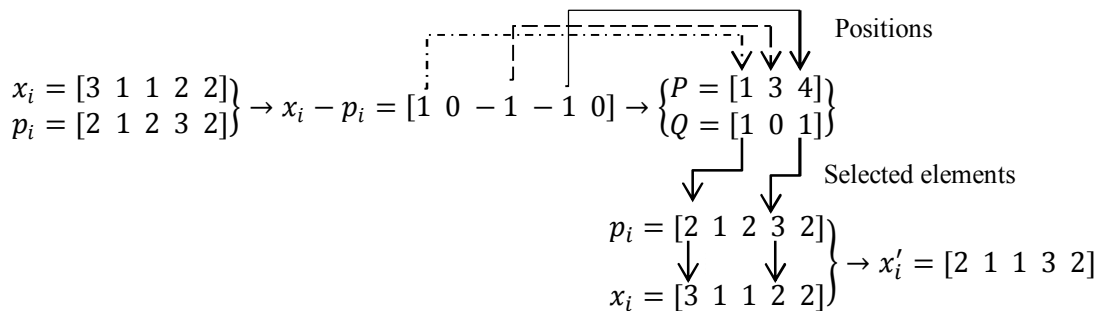


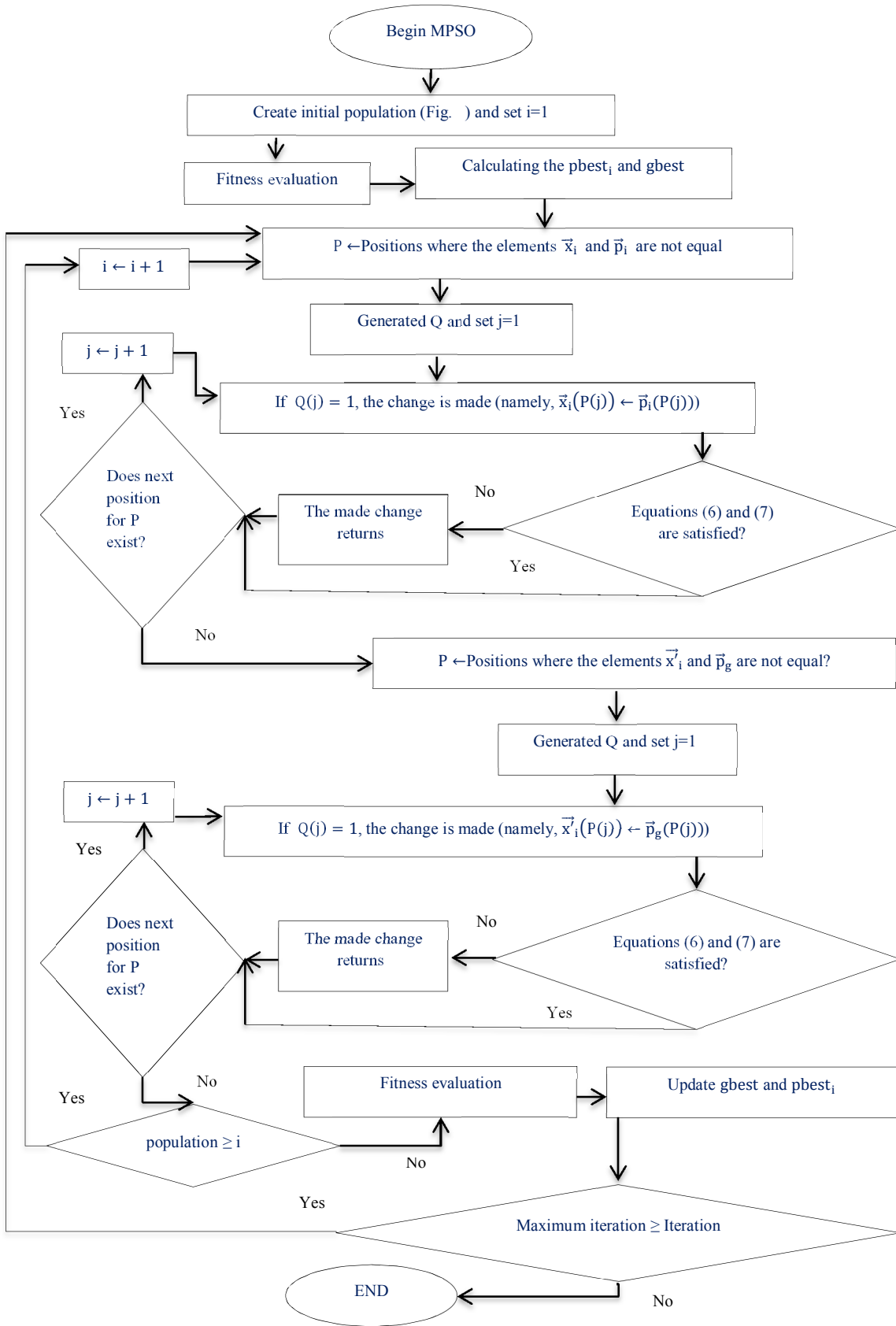Fig. 5. An example of how doing the first stage for the MPSO algorithm

Fig. 6. Flowchart of the 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, with regard to evaluation function, which is called fitness function. Members with higher fitness can be selected by the high probability. So, members, being less fit, are substituted by the better ones. This procedure is repeated to reach to a certain number of iterations (Mahdavi et al., 2009).

GA chromosome structure for this model is like particle structure for MPSO. The pescudo code 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 objectives function.
3.  Producing a new population is based on the repetition of the following steps:

## 3.3.  *Crossover operator*

3.3.1.  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.

3.3.2. 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 has been selected. All data beyond that point in either chromosome is swapped between the two parent chromosomes. The derived combinations are the children (see Fig. 7). After crossover, the feasibility of constraints (6) and (7) are evaluated. The procedure continues, if it is true, otherwise, the made change returns.

3.4.  The fraction of the initial population is selected with a probability 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 Mahdavi et al. (2009) is used for the mutation. After mutation, the feasibility of constraints (6) and (7) are evaluated. The procedure continues, if it is true, otherwise, the made change returns.

4.  The size of the next population is as the same as the previous one, that 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.

## 7.  **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 B&B algorithm, MPSO and GA. Finally, the generated solutions will be compared with each other according to the criteria of solution quality and solving time. The proposed model is coded in LINGO 8.0 optimization software and the proposed meta-heuristic algorithms are coded in MATLAB 2010a on a computer with 2.99 GB RAM and core i3with 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 was solved in less than 5400 seconds (1.5 hour), it is categorized as small-medium size problems; otherwise, it is categorized as large size problems. This procedure is similar to Safaei et al. (2008). Since the efficiency of the meta-heurstics algorithms depends strongly on the operators and the parameters, the design of experiments is done to set parameters. Design of experiments finds the combination of control factors that has the lowest variation, aiming for robustness in solutions. To cover different sizes, problems 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 the full factorial design and Taguchi technique design, respectively. A summary of all obtained MPSO and GA parameters are given in Tables 2 and 3, respectively.
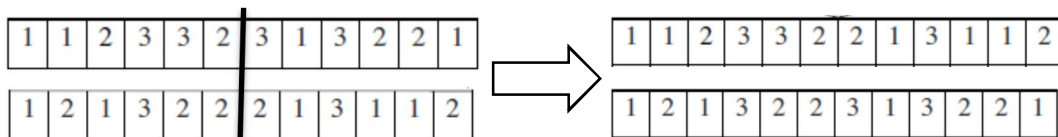
Fig. 7. An example for one-point crossover

Table 2
The obtained values for MPSO parameters

| Size / parameter | 8×11 | 9×18 | 16×30 |
|---|---|---|---|
| population | 450 | 1050 | 4000 |
| Iteration | 10 | 60 | 50 |

Table 3
The obtained values for GA parameters

| Size / parameter | 8×11 | 9×18 | 16×30 |
|---|---|---|---|
| Population | 450 | 1100 | 4000 |
| Iteration | 20 | 60 | 70 |
| Probability of crossover | 0.7 | 0.7 | 0.6 |
| Probability of mutation | 0.4 | 0.3 | 0.1 |
| Number of members competing in the tournament | 3 | 2 | 3 |

According to the Lingo software documents, $F_{best}$ 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 the objective function ($F^*$) is limited between $F_{bound} \geq F^* \geq F_{best}$.

indicates the comparison of the Lingo software B&B algorithm results with MPSO and GA corresponding to 19 test problems. Each problem is run 10 times and the average of OFV ($Z_{ave}$), the best OFV ($Z_{best}$), and also average of run time ($T_{MPSO}$) are represented in. 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 the Lingo software B&B algorithm, if $F_{bound} = F_{best}$, the optimal solution is achieved. In, 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 positive. 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 negative numbers. To compare MPSO and GA some columns are defined as: Ga-ave, Ga-best and R that are formulated as: $Ga^{ave} = (Z_{ave}^{MPSO} - Z_{ave}^{GA})/Z_{ave}^{MPSO}$, $Ga^{best} = (Z_{best}^{MPSO} - Z_{best}^{GA})/Z_{best}^{MPSO}$ and $R = (T_{MPSO} - T_{GA})/T_{GA}$, respectively.

As mentioned above, in small-medium size examples, a limited run time (1.5 h) is considered for Lingo solver to find optimal solutions. Therefore, as it can be concluded from, the percent error of optimal solution is very low when different problems are selected. Also, in large size examples, MPSO and GA perform better than the Lingo software B&B algorithm in most problems in a limited time. It implies that MPSO and GA algorithms are so effective in solving the proposed model in all class of problems. Also, performance of MPSO and GA has been indicated in Fig. 8 and Fig.9. With regard to Fig. 8 and Fig. 9, it can be gathered that $Z_{ave}$ and $Z_{best}$ solutions for small and medium size problems are so close to $F_{best}$ and even coincided. In large size problems, meta-heuristic algorithms which have been used, generate better solutions from the Lingo software B&B algorithm or solve problems with negligible error. Fig. 10 represents the time for solving meta-heuristic algorithms and the Lingo software B&B algorithm. It is obvious that the solving time for meta-heuristic algorithms while increasing the size of the problem is much less than the Lingo software B&B algorithm. A paired t-test was conducted to analyze the significant difference between the obtained solutions of the algorithms. The statistical details are shown in Table 4. This test shows that there is a statistically significant difference between solutions obtained by MPSO and GA. Considering this table, it can be gathered that the obtained solutions by MPSO are apparently better than GA.

Table 4
Detailed statistics of paired t test.

| | Paired Differences | | | | | t | df | Sig. (2-tailed) |
|---|---|---|---|---|---|---|---|---|
| | Mean | Std. Deviation | Std. Error Mean | 95% Confidence Interval of the Difference | | | | |
| | | | | Lower | Upper | | | |
| Pair1 MPSO-GA | 3.47421E1 | 70.29767 | 16.12739 | .85971 | 68.62450 | 2.154 | 18 | .045 |

Table 5
Comparison of B&B, MPSO and GA runs

| Problem No. | No. of parts | No. of machines | No. of cells | Mmax | B&B | | | MPSO | | | | | GA | | | | | MPSO & GA comparison | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 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 | 244 | 244 | 0 | 244 | 244 | 0 | 0.00 | 0.00 | 244 | 244 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | - |
| 2 | 5 | 5 | 2 | 3 | 525 | 525 | 0 | 525 | 525 | 0 | 0.00 | 0.00 | 525 | 525 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | - |
| 3 | 7 | 6 | 2 | 3 | 738 | 738 | 0 | 738 | 738 | 0 | 0.00 | 0.00 | 738 | 738 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | - |
| 4 | 8 | 6 | 2 | 4 | 906 | 906 | 1 | 894 | 894 | 0 | -1.32 | -1.32 | 894 | 894 | 0 | -1.32 | -1.32 | 0.00 | 0.00 | - |
| 5 | 9 | 7 | 3 | 4 | 1113 | 1113 | 3 | 1065.4 | 1113 | 0 | -4.28 | 0.00 | 1094.8 | 1113 | 0 | -1.64 | 0.00 | -2.76 | 0.00 | - |
| 6 | 11 | 8 | 3 | 4 | 1408 | 1408 | 13 | 1352 | 1408 | 0 | -3.98 | 0.00 | 1354.4 | 1384 | 0 | -3.81 | -1.70 | -0.18 | 1.70 | - |
| 7 | 12 | 9 | 3 | 4 | 2223 | 2232 | 70 | 2062.8 | 2133 | 4.3 | -7.21 | -4.05 | 2068.2 | 2133 | 7 | -6.96 | -4.05 | -0.26 | 0.00 | -38.57 |
| 8 | 18 | 8 | 3 | 5 | 1792 | 1792 | 588 | 1671.2 | 1704 | 4.7 | -6.74 | -4.91 | 1676.8 | 1696 | 7 | -6.43 | -5.36 | -0.34 | 0.47 | -32.86 |
| 9 | 17 | 10 | 3 | 5 | 2680 | 2730 | 5400 | 2640 | 2660 | 18.3 | -1.49 | -0.75 | 2639 | 2660 | 80.6 | -1.53 | -0.75 | 0.04 | 0.00 | -77.30 |
| 10 | 18 | 9 | 3 | 5 | 2277 | 2277 | 1766 | 2088.9 | 2142 | 4.8 | -8.26 | -5.93 | 2117.7 | 2160 | 7 | -7.00 | -5.14 | -1.38 | -0.84 | -31.43 |
| 11 | 19 | 9 | 3 | 5 | 2331 | 2331 | 997 | 2195.1 | 2214 | 6.1 | -5.83 | -5.02 | 2176.2 | 2205 | 7 | -6.64 | -5.41 | 0.86 | 0.41 | -12.86 |
| 12 | 20 | 9 | 3 | 5 | 2304 | 2304 | 3560 | 2204.1 | 2232 | 19 | -4.34 | -3.13 | 2181.6 | 2232 | 7 | -5.31 | -3.13 | 1.02 | 0.00 | 171.43 |
| 13 | 19 | 10 | 3 | 5 | 2770 | 2820 | 5400 | 2625 | 2660 | 30.2 | -5.23 | -3.97 | 2641 | 2650 | 77.2 | -4.66 | -4.33 | -0.61 | 0.38 | -60.88 |
| 14 | 24 | 11 | 4 | 5 | 3377 | 3586 | 5400 | 3197.7 | 3311 | 21.9 | -5.31 | -1.95 | 3179 | 3245 | 84.1 | -5.86 | -3.91 | 0.58 | 1.99 | -73.96 |
| 15 | 24 | 14 | 4 | 5 | 4928 | 5404 | 5400 | 4671.8 | 4760 | 23.6 | -5.20 | -3.41 | 4635.4 | 4718 | 90.4 | -5.94 | -4.26 | 0.78 | 0.88 | -73.89 |
| 16 | 30 | 16 | 4 | 5 | 6544 | 7622 | 5400 | 6864 | 7040 | 34.5 | 4.89 | 7.58 | 6729.6 | 6960 | 89.1 | 2.84 | 6.36 | 1.96 | 1.14 | -61.28 |
| 17 | 35 | 20 | 4 | 7 | 8640 | 11840 | 5400 | 10408 | 10820 | 40.7 | 20.46 | 25.23 | 10220 | 10500 | 96.9 | 18.29 | 21.53 | 1.81 | 2.96 | -58.00 |
| 18 | 37 | 20 | 5 | 7 | 10160 | 12040 | 5400 | 10742 | 11080 | 40.9 | 5.73 | 9.06 | 10588 | 10840 | 101.75 | 4.21 | 6.69 | 1.43 | 2.17 | -59.80 |
| 19 | 43 | 22 | 5 | 7 | 10736 | 13398 | 5400 | 11077 | 11462 | 42.7 | 3.18 | 6.76 | 10903.2 | 11264 | 113.8 | 1.56 | 4.92 | 1.57 | 1.73 | -62.48 |
| average | | | | | | | | | | | -1.31 | 0.75 | | | | -1.59 | 0.01 | 0.24 | 0.68 | -36.30 |
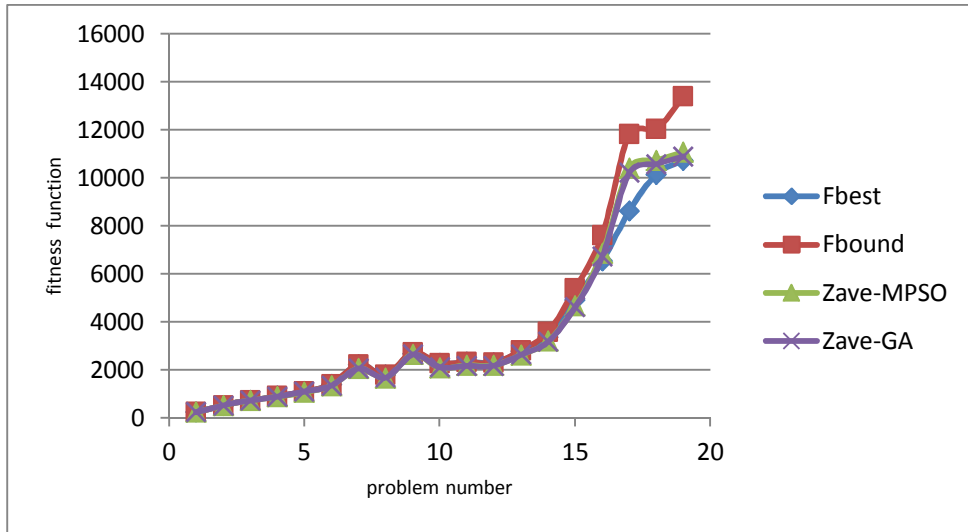
Fig. 8. Comparison of B & B,MPSO and GA results (Table 5):( $Z_{ave}$ found by meta heuristic algorithms) vs. ( $F_{bound}$ and $F_{best}$)
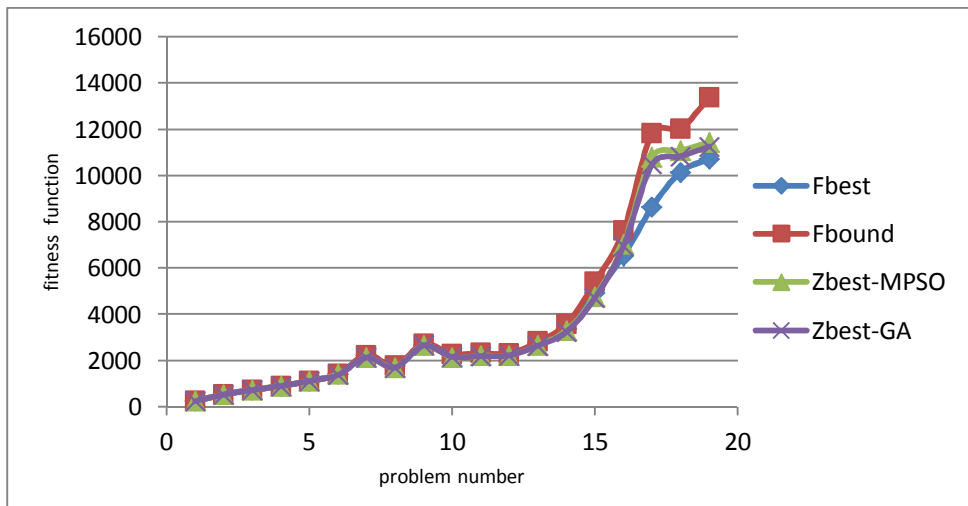


Fig. 9. Comparison of B &B, MPSO and GA results (Table 5): ( $Z_{best}$ found by meta heuristic algorithms) vs. ( $F_{bound}$ and $F_{best}$)
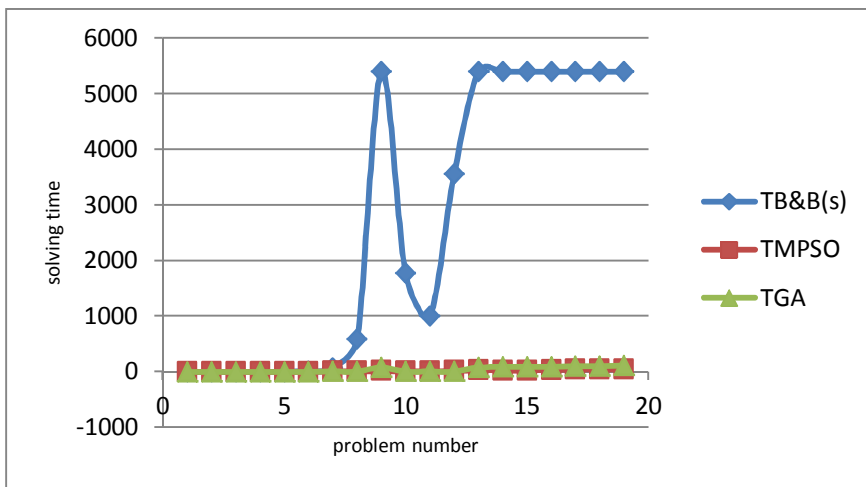


Fig. 10. Comparison of solving time

## 8. Conclusions

In this paper, a new nonlinear mathematical model for the stochastic cell formation problem was proposed. The queuing theory was used for modeling and the machine was assumed as a server and the part as a customer. Inter-arrival time and processing time were distributed exponentially. To find the optimal solution in a reasonable time, an equivalent procedure was used to linearize the nonlinear model. To study the proposed model, 19 sample problems with different sizes were used. The proposed model was solved by the Lingo software B & B algorithm. Because of the complexity class of this problem that is categorized as NP-hard, two meta-heuristic algorithms are developed to solve problems. Parameters setting for the MPSO algorithm have been done by the full factorial technique and for GA have been operated by the Taguchi technique. Finally, the generated solutions by MPSO, GA and the Lingo software B & B were compared with each other by considering solving times. These comparisons confirmed the high efficiency of the proposed meta-heuristic algorithms for large size problems compared with the Lingo software B&B algorithm. The solutions generated by the MPSO algorithm were apparently better in comparison with GA solutions. Some suggestions can be highlighted for future research: implementing this study by maintaining its conditions for the dynamic state, adding machine locating concept to this study for achieving real results, and solving this problem with other meta-heuristic approaches such as ant-colony algorithm and comparing results with this study.

## Appendix A

An equivalent ($\equiv$) objective function (3) is generated with respect to $W_j$ (the mean waiting time) that is always a positive quantity and defined binary integer variable in the equation (9), as follows:

$$Max\left(\frac{1}{M}\sum_{j=1}^{M} W_j\right) = Max\left(\frac{W_1}{M}+\frac{W_2}{M}+\cdots+\frac{W_M}{M}\right)$$

$$= Max\left(\frac{W_1}{M}\right) + Max\left(\frac{W_2}{M}\right) + \cdots$$

$$+Max\left(\frac{W_M}{M}\right) \equiv Min\left(\frac{M}{W_1}\right) + Min\left(\frac{M}{W_2}\right) + \cdots + Min\left(\frac{M}{W_M}\right)$$

$$= Min\left(\frac{M}{W_1}+\frac{M}{W_2}+\cdots+\frac{M}{W_M}\right)$$

$$= Min\left(M\sum_{j=1}^{M}\left(\mu_j - \sum_{k=1}^{C}\sum_{i=1}^{P}\lambda_i a_{ij}V_{ijk}\right)\right)$$

$$= Min\left(M\sum_{j=1}^{M}\mu_j - M\sum_{j=1}^{M}\sum_{k=1}^{C}\sum_{i=1}^{P}\lambda_i a_{ij}V_{ijk}\right)$$

The first term is always constant, thus it has no role in minimization:

$$Min\left(-M\sum_{j=1}^{M}\sum_{k=1}^{C}\sum_{i=1}^{P}\lambda_i a_{ij}V_{ijk}\right)$$

$$\equiv Max\left(M\sum_{j=1}^{M}\sum_{k=1}^{C}\sum_{i=1}^{P}\lambda_i a_{ij}V_{ijk}\right)$$

## References

[1] Ahi, A., Aryanezhad, M.B., Ashtiani, B. and Makui, A. (2009). A novel approach to determine cell formation, intracellular machine layout and cell layout in the CMS problem based on TOPSIS method. Computers & Operations Research, 36, 1478 – 1496.

[2] Arzi, Y., Bukchin, J. and Masin, M. (2001). An efficiency frontier approach for the design of cellular manufacturing systems in a lumpy demand environment. European Journal of Operational Research, 134, 346-364.

[3] Asgharpour, M.J. and Javadian, N. (2004). Solving a stochastic cellular manufacturing model by using genetic algorithms. International Journal of Engineering, 17(2), 145-156.

[4] Chung, S.H., Wu, T.H. and Chang, C.C. (2011). An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. Computers & Industrial Engineering, 60, 7–15.

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

[6] Das, K., Lashkari, R.S. and Sengupta, S. (2007). Machine reliability and preventive maintenance planning for cellular manufacturing systems. European Journal of Operational Research, 183, 162–180.

[7] Das, K., Lashkari, R.S. and Sengupta, S. (2007). Reliability consideration in the design and analysis of cellular manufacturing systems. International Journal of Production Economics, 105, 243–262.

[8] Durán, O., Rodriguez, N. and Consalter, L.A. (2010). Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. Expert Systems with Applications, 37, 1563–1567.

[9] Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science, IEEE Press, 39-43.

[10] Egilmez, G. and Suer, G. (2014). The impact of risk on the integrated cellular design and control. International Journal of Production Research, 52(5), 1455–1478.

[11] Egilmez, G., Süer, G.A. and Huang, J. (2012). Stochastic cellular manufacturing system design subject to maximum acceptable risk level. Computers & Industrial Engineering, 63, 842–854.

[12] Fardis, F., Zandi, A. and Ghezavati, V.R. (2013). Stochastic extension of cellular manufacturing systems: a queuing-based analysis. Journal of Industrial Engineering International, 9, 1-8.

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

[14] Ghezavati, V.R. and Saidi-Mehrabad, M. (2011). An efficient hybrid self-learning method for stochastic cellular manufacturing problem: A queuing-based analysis. Expert Systems with Applications, 38, 1326–1335.

[15] Hillier, F.S. and Lieberman, G.J. (1995). Introduction to operation research.7th edition, McGraw-Hill, New York.

[16] JabalAmeli, M.S. and Arkat, J. (2008). Cell formation with alternative process routings and machine reliability consideration. International Journal of Advanced Manufacturing Technology, 35, 761–768.

[17] JabalAmeli, M.S., Arkat, J. and Barzinpour, F. (2008). Modelling the effects of machine breakdowns in the generalized cell formation problem. International Journal of Advanced Manufacturing Technology, 39, 838–850.

[18] Kannan, V.R. and Palocsay, S.W. (1999). Cellular vs process layouts: an analytic investigation of the impact of learning on shop performance. Omega, 27(5), 583-592.

[19] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. Proceedings of international conference on neural networks IV, IEEE Press, 1942-1948.

[20] King, J.R. and Nakornchai, V. (1982). Machine-component group formation in group technology: review and extension. international journal of production research, 20 (2), 117-133.

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

[22] Papaioannou, G. and Wilson, J.M. (2010). The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research. European Journal of Operational Research, 206, 509–521.

[23] Pitchuka, L.N., Adil, G.K. and Ananthakumar, U. (2006). Effect of conversion of functional layout to a cellular layout on the queue time performance: some new insights. International Journal of Advanced Manufacturing Technology, 31, 594–601.

[24] Poli, R., Kennedy, J. and Blackwell, T. (2007). Particle swarm optimization. Swarm Intelligence, 1(1), 33-57.

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

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

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