CrossMark

ORIGINAL RESEARCH

A stochastic model for the cell formation problem considering machine reliability

Bahman Esmailnezhad[1] · Parviz Fattahi[1] · Amir Saman Kheirkhah[1]

Abstract This paper presents a new mathematical model to solve cell formation problem in cellular manufacturing systems, where inter-arrival time, processing time, and machine breakdown time are probabilistic. The objective function maximizes the number of operations of each part with more arrival rate within one cell. Because a queue behind each machine; queuing theory is used to formulate the model. To solve the model, two metaheurstic algorithms such as modified particle swarm optimization and genetic algorithm are proposed. For the generation of initial solutions in these algorithms, a new heuristic method is developed, which always creates feasible solutions. Both metaheurstic algorithms are compared against global solutions obtained from Lingo software's branch and bound (B&B). Also, a statistical method will be used for comparison of solutions of two metaheurstic algorithms. The results of numerical examples indicate that considering the machine breakdown has significant effect on block structures of machine-part matrixes.

Keywords Cell formation · Queuing theory · Metaheurstic algorithm · Reliability

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) is an application of GT in a manufacturing system. CM involves processing a collection of similar parts (part families) on a dedicated cluster of machines or manufacturing processes (cells). The cell formation (CF) problem in CM systems is the decomposition of the manufacturing systems into cells (Singh and Rajamani 1996). Cellular manufacturing problems can be under static or dynamic conditions. In static conditions, CF is done for a single-period planning horizon. In real problems, some input parameters such as costs, demands, processing times, and setup times are uncertain and this uncertainty can affect the results. In the static stochastic problems, it is assumed that our information on model parameters is incomplete. 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. In the following, a review of previous studies about static stochastic cell formation problem is presented in four parts such as processing time, the mix product, the demand, and the reliability, respectively.

Saidi-Mehrabad and Ghezavati (2009) assumed the processing time and the time between two successive arrivals to cell described by exponential distribution in the CF problem. For analyzing this problem, they used queuing theory in which machine has been considered as a server and the part as a customer. The aim of this model is to minimize the summation of three costs: (1) the idleness costs for machines (2) total cost of sub-contracting for exceptional elements, and (3) the cost of resource underutilization. Ghezavati and Saidi-Mehrabad (2010) proposed a mathematical model for the CM problem integrated with group scheduling in an uncertain space. Within this model, CF and scheduling decisions are optimized concurrently. It is assumed that processing time of

✉ Parviz Fattahi
Fattahi@basu.ac.ir

[1] Faculty of Industrial Engineering, Bu-Ali Sina University, Hamedan, Iran

parts on machines is stochastic and described by discrete scenarios. Their model minimizes the expected total cost including maximum tardiness cost among all parts, the cost of sub-contracting for exceptional elements, and the cost of resource underutilization. Egilmez and Suer (2011a, b) presented a mathematical model for CF which minimized the number of tardy jobs and total probability of tardiness. They assumed that processing time of each job has a normal distribution. Ghezavati and Saidi-Mehrabad (2011) assumed that each machine works as a server and each part is a customer where servers should provide service to 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 (2011) evaluated the 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 with respect to the suppliers network in a supply chain in order to outsource exceptional operations. Fardis et al. (2013) examined the CF problem considering that the arrival rate of parts into cells and machine service rate are stochastic parameters, which have described by an exponential distribution. The objective function of presented model minimized summation of the idleness cost of machines, the sub-contracting cost for exceptional parts, non-utilizing machine cost, and the holding cost of parts in the cells.

The mix product is defined as the set of part types in a factory which can be produced and each factory looks for the best mix product. The mix product and value of demand for each product in the mix product due to customized products, shorter product life cycles, and unpredictable patterns of demand, are not known exactly at the time of designing the manufacturing cells. The composition of the product mix is determined by demand and is probabilistic in nature. Seifoddini (1990) proposed a stochastic CF model in which for each mix product a probability had been attributed. He calculated the expected intercellular material handling cost for each machine cell arrangement under all possible product mixes. Madhusudanan Pillai and Chandrasekharan (2010) evaluated the CF problem under probabilistic product mix. Each product mix is specified with a scenario and to each scenario has been attributed probability of occurrence. They minimized inter-cell material handling. Jayakumar and Raju (2011) presented a mathematic model for the CF problem in which for each scenario probability of occurrence has been attributed. The objective function of this model is to minimize the total 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.

A review of studies done in the demand area is provided in the following. Cao and Chen (2005) offered CF with scenarios for product demand. In this model, an occurrence probability had been assigned for each scenario. The objective function of this model minimized machine cost and the expected inter-cell material handling cost. Tavakkoli-Moghaddam et al. (2007) 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-cell 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 (2011a, b) 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 the objective function of the production rate maximization. Then, the manpower is assigned to the manufacture cells to minimize number of labors. In both the models, the processing time and demand have the normal distribution. Ariafar et al. (2011) purposed the model for cell layout in the shop and machines in the machine cells. The demand has been considered as stochastic and with a uniform distribution. This model minimizes the inter-cell and intra-cell material handling costs. Egilmez et al. (2012) considered processing times and customer demand uncertain with the normal distribution. The objective is to design a CM system with product families that are formed with the most similar products and minimum number of cells and machines for a specified risk level. Ariafar et al. (2012) 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 the normal distribution. Rabbani et al. (2012) 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 and in the second objective minimizes the expected total cell loading variation. 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 the normal distribution.

Jabal Ameli et al. (2008) 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 CM system. Jabal Ameli and Arkat (2008) have 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 matrixes. Chung et al. (2011) found that machine reliability has meaningful effects on reducing the total system cost in the CF problem. Arkat et al. (2011) presented the CF problem in general state and considering the reliability. The generalized CF problem follows selecting the best process plan for each part and assigning machines to the cells. In this model, it has been 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.

Many articles investigate CM system 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 for investigation for making more accurate decisions. In this paper, uncertainty in processing time, time between two successive arrivals to cell, and reliability are considered to fill this gap in literature. The structure of this paper is as follows. In Sect. "Problem formulation", the problem formulation is described. The MPSO algorithm and GA are described in Sect. "Notation". The computational results and conclusion are reported in Sects. "Mathematical model" and "The proposed algorithms", respectively.

Problem formulation

In this section, a new mathematical model is proposed for static stochastic cell formation problem. The presented model forms manufacturing cells considering three stochastic parameters including the processing time, the time between two successive arrival parts to the cell, and the

machine availability, simultaneously. In this model, cells are assigned to the parts according to priority of arrival rate, i.e., at first, the cell is assigned to the part which has the most arrival rate. In fact, exceptional elements (EEs) are minimized in this way. EEs are defined as parts which must be processed in different cells, and therefore they have intercellular movements. In this paper, EEs will be outsourced to be operated. M/G/1 queuing model is used to formulate the problem and the machine is assumed as a server and the part as a customer. In the M/G/1 model, the time between two successive arrival customers is exponentially distributed and service time is generally distributed. Service discipline is based on first come, first service. 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 which have been pointed out by these researchers in the article presented by the Cruz et al. (2010). Since the arrival rate to each queuing system is less than the service rate in the presented model, the arrival rate is equal to the output rate. Besides, the arrival time or output time of parts is exponentially distributed. The queuing system is shown in Fig. 1.

Notation

Indexing sets

- i index for parts $i = 1, ..., P$
- j index for machines $j = 1, ..., M$
- k index for cells $k = 1, ..., C$

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
- MTBFj Mean time between failures for machine j
- MTTRj Mean time to repair for machine j

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

Decision variables

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

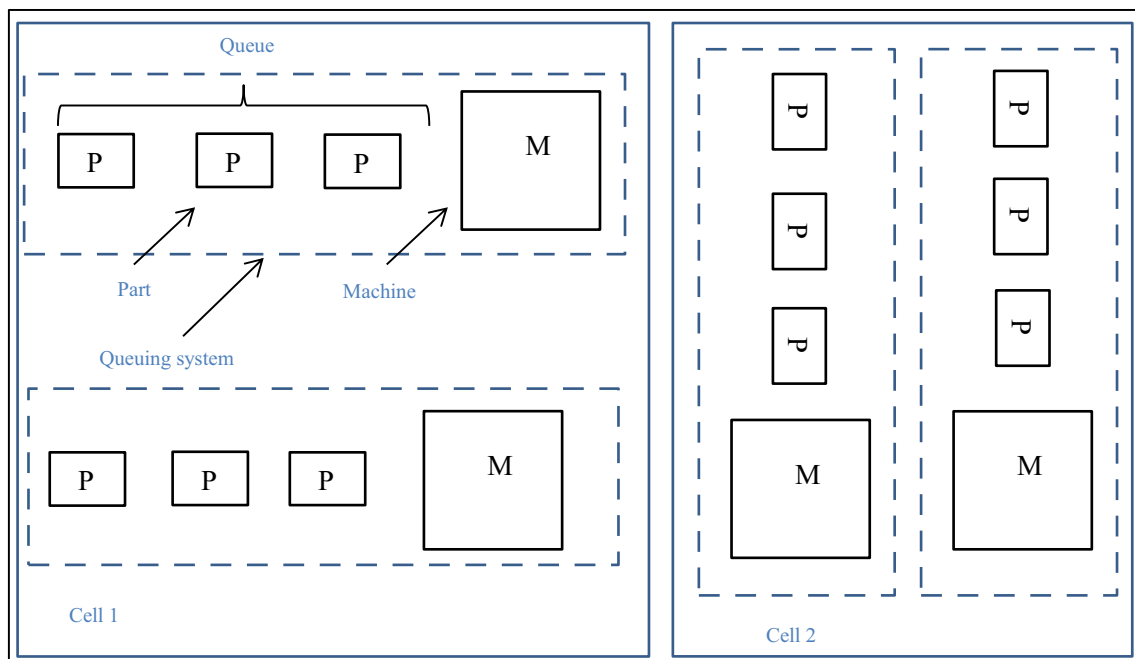$y_{jk} = \begin{cases} 1 & \text{if machine } j \text{ is assigned to cell } k \\ 0 & \text{otherwise} \end{cases}$

Fig. 1 An example of queuing system in the CM system

Mathematical model

Machine reliability has a probabilistic nature. It is assumed that the breakdown time for each machine follows a general distribution with known mean time to repair (MTTR) and known mean time between failures (MTBF). In this section, the reliability of machines into the CF model is discussed. The approach presented by Ameli et al. (2008) has been used for considering the reliability. For investigation of the effect of the reliability on the CF problem, two definitions are given. The number of machine breakdowns for machine j, $N_j(t)$, can be acquired by dividing the production time for machine j, t_j, by the MTBF_j.

$$N_j(t) = \frac{t_j}{\text{MTBF}_j}. \tag{1}$$

By multiplying the MTTR_j by the number of breakdowns calculated in Eq. (1), the total repair time for machine j, $T_j(t)$, can be obtained as follows:

$$T_j(t) = \frac{t_j \times \text{MTTR}_j}{\text{MTBF}_j}. \tag{2}$$

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

mean of total time for machine j

$$= \frac{E_j(t) \times \text{MTTR}_j}{\text{MTBF}_j} + E_j(t) = \frac{\text{MTTR}_j}{\mu_j \times \text{MTBF}_j} + \frac{1}{\mu_j},$$

mean of total time for machine $j = \frac{1}{\mu_j}\left(\frac{\text{MTTR}_j + \text{MTBF}_j}{\text{MTBF}_j}\right) \tag{3}$

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

the production rate for machine $j = \mu_j\left(\frac{\text{MTBF}_j}{\text{MTTR}_j + \text{MTBF}_j}\right) \tag{4}$

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

According to the queuing model and the Fig. 1, the inter-arrival time of parts is uncertain and it is described by the exponential distribution. Also, as each machine processes different parts with different arrival rates, according to this property, the minimum of independent exponential random variables with the arrival rate of $\lambda_1, \lambda_2, \ldots, \lambda_P$ is also exponential with rate $\sum_{i=1}^{P} \lambda_i$ (Frederick and Hillier 2001). Hence, the minimum of the inter-arrival times has an exponential distribution with parameter λ_{eff} (effective arrival rate). The λ_{eff} can be computed as follows:

$$\lambda_{\text{eff}} = \sum_{i=1}^{P} \lambda_i,$$

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

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

Fig. 2 The sample of particle (chromosome) structure

$$\text{Max } z = \frac{1}{M}\sum_{k=1}^{C}\sum_{j=1}^{M}\sum_{i=1}^{P}\lambda_i a_{ij} x_{ik} y_{jk} \tag{5}$$

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

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

$$\sum_{j=1}^{M} y_{jk} \leq M_{\max} \quad \forall k \tag{8}$$

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

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

The objective function (5) maximizes the average effective arrival rate. Maximizing the average effective arrival rate increases processing operations of the part with more arrival rate within one cell. Hence, the number of intercellular movement becomes lower. 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 allowed in each cell. 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 each cell should be less than the maximum number of machines allowed in each cell. Constraint (9) avoids instability of the queuing system, that is, the effective arrival rate will be necessarily less than the 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:

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

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

$$V_{ijk} - x_{ik} - y_{jk} + 1.5 \geq 0 \quad \forall i,j,k \tag{12}$$

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

The proposed algorithms

The CF problem is NP-hard problem (King and Nakornchai 1982). Therefore, precise solution procedures and commercial optimization softwares are unable to reach global optimum in an acceptable amount of time for medium- and large-scale problems. To deal with this deficiency, two algorithms based on MPSO and GA metheuristics have been developed in this paper.

Particle or chromosome structure

The particle representation or the chromosome representation involves two sections: the first section indicates the cells assigned to machines and the second section represents the cells assigned to parts. The used particle or the used chromosome for the proposed model has been presented in Fig. 2.

The proposed generating initial population

To present a qualified initial population, a heuristic method is proposed that always produces a feasible solution. The heuristic method has been presented in Fig. 3. In the first step, machines are allocated to cells based on the capacity of cells, and in the second step parts are allocated to cells considering constraint (9) for all machines.

The MPSO algorithm

Particle swarm optimization (PSO) algorithm by Kennedy and Eberhart (1995), (Eberhart and Kennedy 1995) 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 particle ($\vec{x}$). Particles move
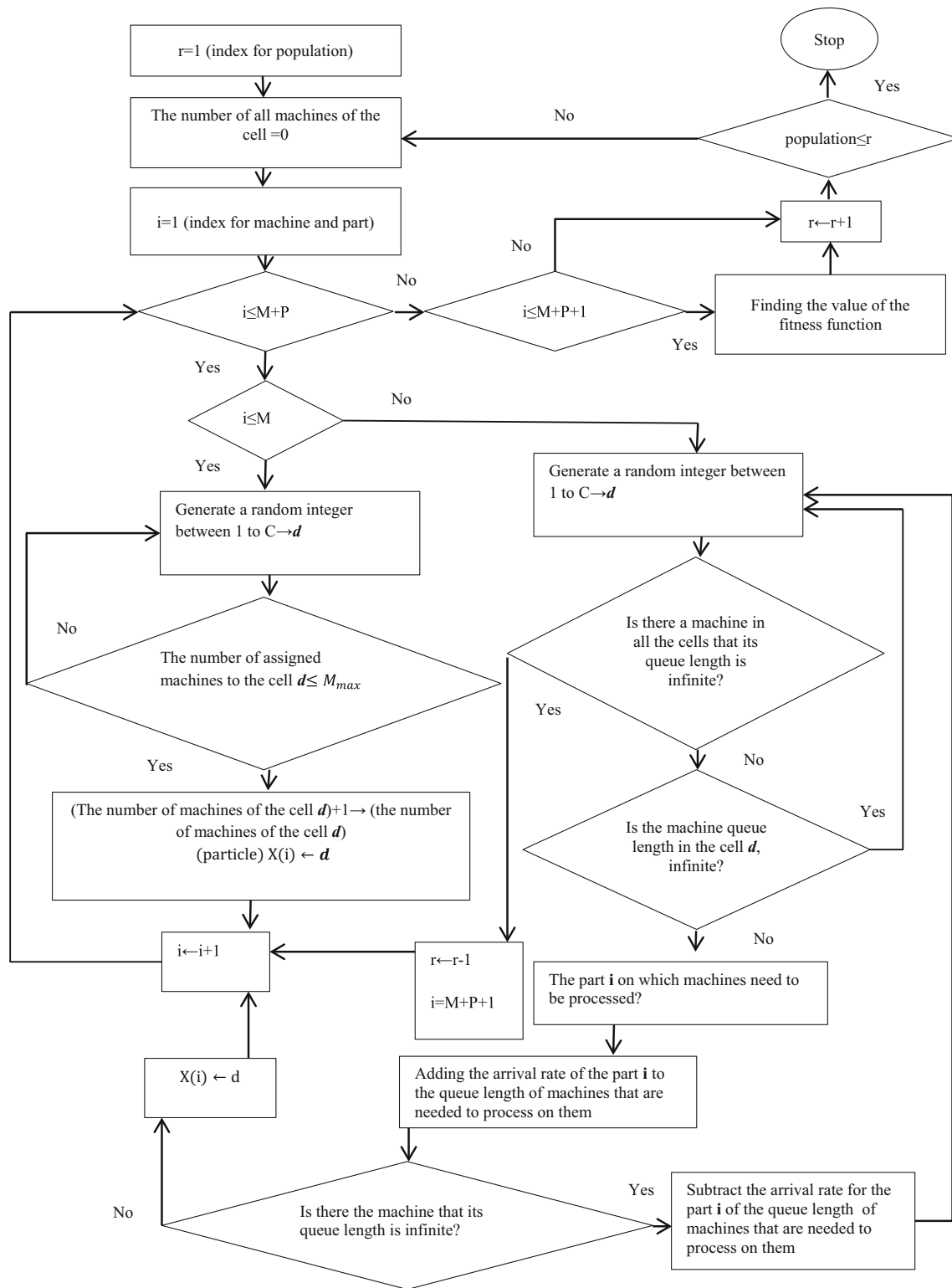
Fig. 3 Heuristic algorithm to generate a feasible initial population

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 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

Fig. 4 An example for the first stage of the new population generation of the MPSO algorithm

$$x_i = [3\ 1\ 1\ 2\ 2]\} \rightarrow x_i - p_i = [1\ 0\ -1\ -1\ 0] \rightarrow \begin{Bmatrix} P = [1\ 3\ 4] \\ Q = [1\ 0\ 1] \end{Bmatrix}$$
$$p_i = [2\ 1\ 2\ 3\ 2]$$

Positions

The selected elements

$$p_i = [2\ 1\ 2\ 3\ 2] \\ x_i = [3\ 1\ 1\ 2\ 2] \end{Bmatrix} \rightarrow x'_i = [2\ 1\ 1\ 3\ 2]$$
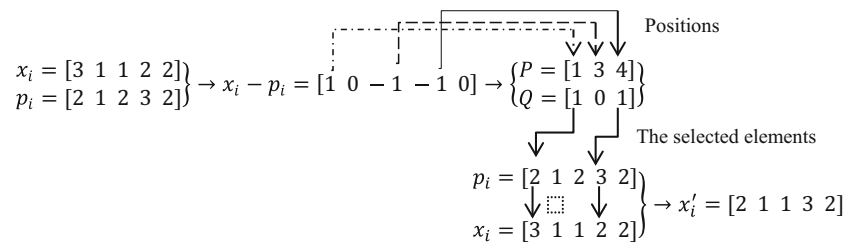
to the current location $\vec{x}_i$ in multidimensional space. Value of the best function result so far for all particles is called gbest, and its location is assigned to p_g.

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 operate on a solution. The MPSO algorithm uses this concept for optimizing.

The pseudocode main steps of the MPSO algorithm are as follows:

1. Initial population is generated using the proposed heuristic algorithm (Fig. 3).
2. The fitness value of all particles is calculated by the linearization objective function.
3. The fitness value of each particle is assigned to pbest$_i$ and its location to $\vec{p}_i$. Identification of the particle in whole swarm with the best success so far, and assignment of its fitness value to gbest and its location to $\vec{p}_g$.
4. Producing a new population is based on the repetition of the following steps:

    4.1. A new vector P is generated to record the positions where the $\vec{x}_i$ and $\vec{p}_i$ elements are not equal. A vector Q is defined with the same length with the vector P. Binary elements for the vector Q are generated randomly.

    4.2. In each position of the vector Q, if the element is 0, the change is not made, 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 the vector $\vec{x}_i$. Then, the feasibility of constraints (8) and (9) is 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 the vector Q (see Fig. 4).

    4.3. For the new location ($\overrightarrow{x'}_i$) a new vector P is generated to record the positions where the $\overrightarrow{x'}_i$ and $\vec{p}_g$ elements are not equal. A new vector Q, is defined with the same length with vector P.

Binary elements for vector Q are generated randomly.

    4.4. In each position of vector Q, if the element is 0, the change is not made, 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 $\vec{p}_g$ which should be copied in vector $\overrightarrow{x'}_i$. Then, the feasibility of constraints (8) and (9) is 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.

5. Comparing particle's fitness value with its pbest$_i$. If current value is better than pbest$_i$, then set pbest$_i$ equal to the current value, and set $\vec{p}_i$ equal to the current location $\vec{x}_i$. Comparing previous gbest with current gbest. If current value is better than previous gbest, then set gbest equal to the current value, and assign its location to $\vec{p}_g$.
6. Check stopping criteria (number of iterations).
7. If the stopping condition is not met, go to step four.

The proposed genetic algorithm

Genetic algorithms (GAs) are search algorithms based on mechanics of the natural selection and the natural genetics. GA exploits the idea of the survival of the fittest and the interbreeding population to create a novel and innovative search strategy. A population of the strings representing solution to the specified problem is maintained by GA, which then iteratively creates the new population from the old by ranking the strings and interbreeding the fittest to create the new strings, which are closer to the optimum solution to a specified problem (Venkata 2011).

The pseudocode main steps of the proposed GA are as follows:

1. Initial population is generated using the proposed heuristic algorithm (see Fig. 3).
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 chromosomes 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 parents' chromosome is selected. All data beyond that point in either chromosome are 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 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 the source (Mahdavi et al. 2009) is used for the mutation. After mutation, the feasibility of constraints (8) and (9) are 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, 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.

Computational results

For finding high-quality solutions, some computational experiments are implemented to validate and verify the proposed model and evaluate the efficiency and performance of the proposed GA and MPSO algorithms. For this purpose, 19 sample problems are defined and then solved by Lingo software's B&B algorithm, MPSO, and GA. Finally, the generated solutions will be compared with each other using the criteria of solution quality and solving time. The proposed model is coded in the LINGO 8.0 optimization software and the proposed metaheurstic 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, it is allowed to run for 5400 s (1.5 h). In the B&B algorithm (obtained by Lingo software package), if the problem can be solved in less than 5400 s (1.5 h), it is categorized as small-, medium-sized problems; otherwise, it is categorized as large-sized problems. This procedure is similar to Safaei et al. (2008). Since the efficiency of the metaheurstics 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 have the lowest variation, which aims 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 a full factorial design and Taguchi technique design, respectively. A summary of all obtained MPSO and GA parameters are given in Tables 1 and 2, respectively.

According to the Lingo software's documents, the 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 $F_{best} \leq F^* \leq F_{bound}$. Table 3 and Table 4 indicate the comparison of the Lingo software's B&B algorithm results with MPSO and GA corresponding to 19 test problems for states of ignoring reliability and considering reliability, respectively. Each problem is ran 10 times and

Table 1 The obtained values for MPSO parameters

Size parameter	8×11	9×18	16×30
Population	450	1050	4000
Iteration	10	60	50

Table 2 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

Table 3 Comparison of B&B, MPSO, and GA results for state of ignoring machine reliability

Problem No.	No. of parts	No. of machines	No. of cells	Mmax	B&B				MPSO
					F_{best}	F_{bound}	TB&B(s)	Z_{ave}	Z_{best}
1	4	4	2	3	15.25	15.25	0	15.25	15.25
2	5	5	2	3	21.00	21.00	0	21.00	21.00
3	7	6	2	3	20.50	20.50	0	20.50	20.50
4	8	6	2	4	25.17	25.17	1	24.83	24.83
5	9	7	3	4	22.71	22.71	3	21.74	22.71
6	11	8	3	4	22.00	22.00	13	21.13	22.00
7	12	9	3	4	27.44	27.44	70	25.47	26.33
8	18	8	3	5	28.00	28.00	588	26.11	26.63
9	17	10	3	5	26.80	27.30	5400	26.40	26.60
10	18	9	3	5	28.11	28.11	1766	25.79	26.44
11	19	9	3	5	28.78	28.78	997	27.10	27.33
12	20	9	3	5	28.44	28.44	3560	27.21	27.56
13	19	10	3	5	27.70	28.20	5400	26.25	26.60
14	24	11	4	5	27.91	29.64	5400	26.43	27.36
15	24	14	4	5	25.14	27.57	5400	23.84	24.29
16	30	16	4	5	25.56	29.77	5400	26.81	27.50
17	35	20	4	7	21.60	29.60	5400	26.02	27.05
18	37	20	5	7	25.40	30.10	5400	26.86	27.70
19	43	22	5	7	22.18	27.68	5400	22.89	23.68
Average									

TMPSO(s)	$G_{ave}(\%)$	$G_{best}(\%)$	Z_{ave}	GA				MPSO & GA comparison (%)		
				Z_{best}	$T_{GA}(s)$	$G_{ave}(\%)$	$G_{best}(\%)$	Ga-ave	Ga-best	R
0	0	0	15.25	15.25	0	0	0	0	0	–
0	0	0	21.00	21.00	0	0	0	0	0	–
0	0	0	20.50	20.50	0	0	0	0	0	–
0	−1.32	−1.32	24.83	24.83	0	−1.32	−1.32	0	0	–
0	−4.28	0	22.34	22.71	0	−1.64	0	−2.76	0	–
0	−3.98	0	21.16	21.63	0	−3.81	−1.7	−0.18	1.7	–
4.3	−7.21	−4.05	25.53	26.33	7	−6.96	−4.05	−0.26	0	−38.57
4.7	−6.74	−4.91	26.20	26.50	7	−6.43	−5.36	−0.34	0.47	−32.86
18.3	−1.49	−0.75	26.39	26.60	80.6	−1.53	−0.75	0.04	0	−77.3
4.8	−8.26	−5.93	26.14	26.67	7	−7	−5.14	−1.38	−0.84	−31.43
6.1	−5.83	−5.02	26.87	27.22	7	−6.64	−5.41	0.86	0.41	−12.86
19	−4.34	−3.13	26.93	27.56	7	−5.31	−3.13	1.02	0	171.43
30.2	−5.23	−3.97	26.41	26.50	77.2	−4.66	−4.33	−0.61	0.38	−60.88
21.9	−5.31	−1.95	26.27	26.82	84.1	−5.86	−3.91	0.58	1.99	−73.96
23.6	−5.2	−3.41	23.65	24.07	90.4	−5.94	−4.26	0.78	0.88	−73.89
34.5	4.89	7.58	26.29	27.19	89.1	2.84	6.36	1.96	1.14	−61.28
40.7	20.46	25.23	25.55	26.25	96.9	18.29	21.53	1.81	2.96	−58
40.9	5.73	9.06	26.47	27.10	101.75	4.21	6.69	1.43	2.17	−59.8
42.7	3.18	6.76	22.53	23.27	113.8	1.56	4.92	1.57	1.73	−62.48
	−1.31	0.75				−1.59	0.01	0.24	0.68	−36.3

the average of OFV (Z_{ave}), the best OFV (Z_{best}), and also average of run time (T_{MPSO}) are represented in this two tables. The relative gap between the best OFV found by Lingo (F_{best}) and Z_{ave} that is found by the metaheurstic 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

Table 4 Comparison of B&B, MPSO, and GA results for state of considering machine reliability

Problem No.	No. of parts	No. of machines	No. of cells	Mmax	B&B				MPSO
					F_{best}	F_{bound}	TB&B (s)	Z_{ave}	Z_{best}
1	4	4	2	3	15.25	15.25	0	15.25	15.25
2	5	5	2	3	19.20	19.20	0	19.20	19.20
3	7	6	2	3	20.50	20.50	0	20.50	20.50
4	8	6	2	4	22.67	22.67	1	22.67	22.67
5	9	7	3	4	20.86	20.86	5	20.23	20.86
6	11	8	3	4	22.00	22.00	12	20.59	22.00
7	12	9	3	4	24.89	24.89	24	23.88	24.89
8	18	8	3	5	26.25	26.25	250	25.14	25.75
9	17	10	3	5	25.50	25.50	1517	24.77	25.00
10	18	9	3	5	25.67	25.67	2154	25.08	25.56
11	19	9	3	5	26.11	26.11	1811	25.53	26.11
12	20	9	3	5	27.11	27.11	3855	26.27	26.78
13	19	10	3	5	26.20	26.20	3870	25.74	26.20
14	24	11	4	5	25.64	26.82	5400	25.04	25.64
15	24	14	4	5	22.57	26.00	5400	22.02	22.57
16	30	16	4	5	23.44	27.31	5400	25.91	26.69
17	35	20	4	7	23.55	26.55	5400	24.39	25.00
18	37	20	5	7	24.05	27.85	5400	25.46	26.55
19	43	22	5	7	20.00	25.32	5400	22.15	23.32
Average									

TMPSO (s)	G_{ave} (%)	G_{best} (%)	Z_{ave}	GA				MPSO & GA comparison (%)		
				Z_{best}	TGA (s)	G_{ave} (%)	G_{best} (%)	Ga-ave	Ga-best	R
0	0	0	15.25	15.25	0	0	0	0	0	–
0	0	0	19.20	19.20	0	0	0	0	0	–
0	0	0	20.50	20.50	0	0	0	0	0	–
0	0	0	22.67	22.67	0	0	0	0	0	–
0	−3.01	0	20.70	20.86	0	−0.75	0	−2.33	0	–
0	−6.42	0	21.10	22.00	0	−4.09	0	−2.49	0	–
0	−4.06	0	24.44	24.89	7	−1.79	0	−2.37	0	−100
0.4	−4.24	−1.9	25.58	26.00	7	−2.57	−0.95	−1.74	−0.97	−94.29
6.3	−2.86	−1.96	24.71	25.00	7	−3.1	−1.96	0.24	0	−10
18.7	−2.29	−0.43	25.03	25.33	7	−2.47	−1.3	0.18	0.87	167.14
7.2	−2.21	0	25.71	26.11	7	−1.53	0	−0.7	0	2.86
24.2	−3.11	−1.23	25.84	26.33	7	−4.67	−2.87	1.61	1.66	245.71
29.5	−1.76	0	25.19	26.10	7	−3.85	−0.38	2.14	0.38	321.43
22	−2.34	0	25.09	25.45	85.08	−2.13	−0.71	−0.21	0.71	−74.14
23.2	−2.44	0	22.00	22.50	89.2	−2.53	−0.32	0.1	0.32	−73.99
37.7	10.56	13.87	25.38	25.69	88.6	8.27	9.6	2.07	3.75	−57.45
40.9	3.55	6.16	23.86	24.35	94.9	1.3	3.4	2.17	2.6	−56.9
42.5	5.86	10.4	24.73	25.60	100.4	2.83	6.44	2.87	3.58	−57.67
42	10.73	16.59	21.55	22.77	117.2	7.73	13.86	2.71	2.34	−64.16
	−0.21	2.18				−0.49	1.31	0.22	0.8	11.43

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 3 and 4, in some cases, Z_{ave} and Z_{best} are between F_{bound} and F_{best} that shows a feasible better
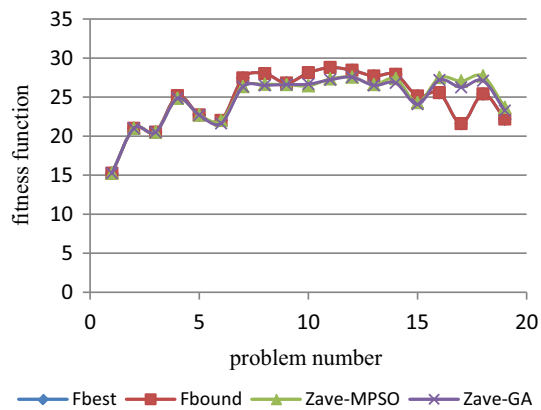
Fig. 5 Comparison of B&B, MPSO, and GA results (Table 3) for state of ignoring machine reliability
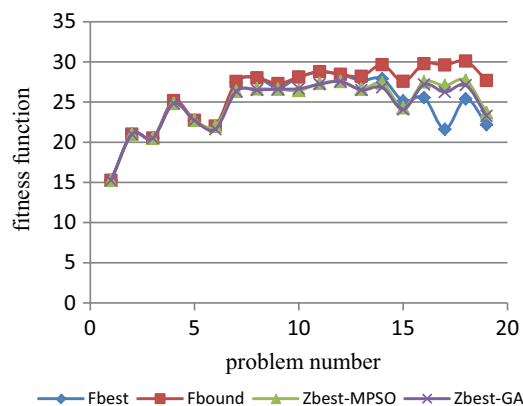

Fig. 6 Comparison of B&B, MPSO, and GA results (Table 3) for state of ignoring machine reliability
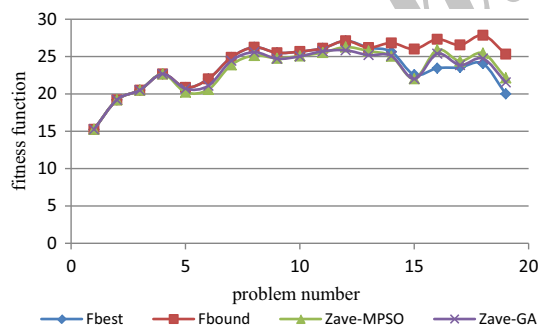

Fig. 7 Comparison of B&B, MPSO, and GA results (Table 4) for state of considering machine reliability

solution, under this condition G_{ave} and G_{best} are positive. But in cases where Z_{ave} and Z_{best} are out of the domain of $[F_{\text{best}}, F_{\text{bound}}]$, G_{ave} and G_{best} will be negative numbers. For comparing MPSO and GA, some columns are defined as Ga-ave, Ga-best, and R that are formulated as follows: $\text{Ga}^{\text{ave}} = \left(Z_{\text{ave}}^{\text{MPSO}} - Z_{\text{ave}}^{\text{GA}}\right)/Z_{\text{ave}}^{\text{MPSO}}$, $\text{Ga}^{\text{best}} = \left(Z_{\text{best}}^{\text{MPSO}} - Z_{\text{best}}^{\text{GA}}\right)/Z_{\text{best}}^{\text{MPSO}}$, and $R = (T_{\text{MPSO}} - T_{\text{GA}})/T_{\text{GA}}$, respectively.
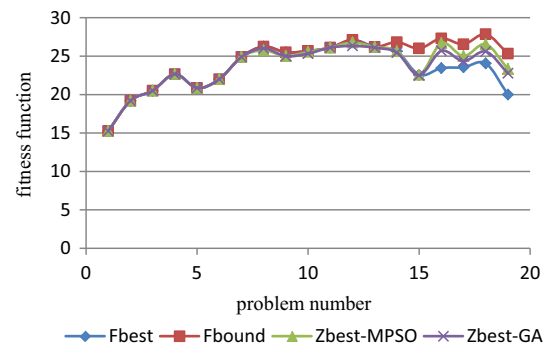
Fig. 8 Comparison of B&B, MPSO, and GA results (Table 4) for state of considering machine reliability
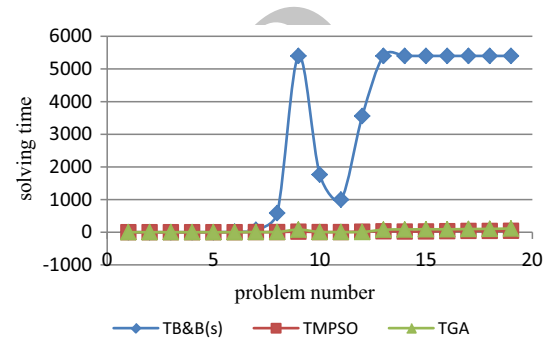

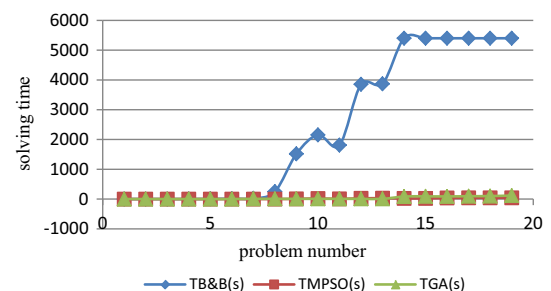**Fig. 9** Comparison of solving time for state of ignoring machine reliability

Fig. 10 Comparison of solving time for state of considering machine reliability

Fig. 11 Optimal solution of example 7 for state of ignoring machine reliability

Table 5 Detailed statistics of paired t test for state of 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	0.06947	0.27840	0.06387	−0.06471	0.20366	1.088	18	0.291

Table 6 Detailed statistics of paired t test for state of 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	0.06789	0.40029	0.09183	−0.12504	0.26083	0.739	18	0.469

As mentioned above in small, medium-sized examples, a limited run time (1.5 h) is considered for Lingo solver to find optimal solutions. Therefore, as it can be concluded from Tables 3 and 4, 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 most problems in limited time. It implies that MPSO and GA algorithms are so effective to solve the proposed model in all class of problems. Also, performance of MPSO and GA in two states of ignoring reliability and considering reliability has been indicated in Figs. 5, 6, 7, 8, and 9. From Figs. 6, 7, 8, and 9, it can be gathered that Z_{ave} and Z_{best} solutions for small- and medium-sized problems are so close to F_{best} and even coincided. In large-sized problems, the metaheurstic algorithms which have been used, generate better solutions from lingo software's B&B algorithm or solve problems with negligible error. Figures 10 and 11 represent the time for solving the metaheurstic algorithms and the Lingo software's B&B algorithm for two states of ignoring reliability and considering reliability, respectively. It is obvious that the solving time for the metaheurstic algorithms with the increasing size of the problem is much less than the Lingo software's B&B algorithm. Two series of paired t test were conducted to analyze significant difference between the obtained solutions of the metaheurstic algorithms for two states of ignoring reliability and considering reliability, respectively. The statistical details are shown in Tables 5 and 6. Tests show that there is no statistically significant difference between solutions obtained by MPSO and GA in both states.

In Table 7, the best solution obtaining from three algorithms for each problem has been used for comparison of effect of machine failure. Based on the constraint (9), the solution space of the proposed model in the state of considering machine reliability is less than the state of ignoring machine reliability. The small solution space is the reason that some values given for state of considering machine reliability are less than that for state of ignoring machine reliability. Because the objective function of this model is maximization, the values for reduction percent mentioned in Table 7 are positive. The optimal solutions of example 7 for two states of ignoring reliability and considering reliability have been given to evaluate the effect of reliability (see Figs. 11, 12). The results of solving numerical examples show that the reliability consideration has significant impacts on the final block diagonal form of machine-part matrixes. Besides, the reduction of reliability reduces the right side of the constraint (9) (see appendix A). Therefore, some operations of some parts do not process within the cell as there is an unreliable machine, although they need that machine for processing. Thus, an unreliable machine tends to form a smaller cell. The optimal solutions for example 2 for three states of MTBF of the machine 1 are presented in Figs. 13, 14, and 15. At the state MTBF = 1.9, the machine 1 cannot process any part within the cell 2 because the processing time of parts is more than the service capacity of the machine 1. Thus, those operations of parts are outsourced. Indeed, machine 1 can be deleted at this state.

Conclusion and future work

In this paper, a new stochastic nonlinear model to solve CF problem within the queuing theory framework with random variables such as time between two successive arrival parts, processing time, and machine availability has been

Table 7 Comparison of effect of machine failure for two states of 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	15.25	21.00	20.50	25.17	22.71	22.00	27.44	28.00	26.80	28.11	28.78	28.44	27.70	27.91	25.14	27.50	27.05	27.70	23.68
Considering machine reliability	15.25	19.20	20.50	22.67	20.86	22.00	24.89	26.25	25.50	25.67	26.11	27.11	26.20	25.64	22.57	26.69	25.00	26.55	23.32
Reduction percent	0.00	8.57	0.00	9.93	8.18	0.00	9.31	6.25	4.85	8.70	9.27	4.69	5.42	8.14	10.23	2.95	7.58	4.15	1.54

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

Fig. 13 Optimal solution of example 2 for state of MTBF = 60

Fig. 14 Optimal solution of example 2 for state of MTBF = 4.1

Fig. 15 Optimal solution of example 2 for state of MTBF = 1.9

presented. To find out the optimal solution in a reasonable time, the proposed nonlinear model was linearized using auxiliary variable. The time between two successive arrival customers had exponential distribution and service time is distributed generally. Numerical examples showed that the reliability consideration has meaningful effects on the final block diagonal form of machine-part matrixes. Because of complexity class of this problem that was categorized as NP-hard, two metaheurstic algorithms based on genetic and MPSO algorithms were developed to solve problems. Also, since the efficiency of metaheurstic algorithms depends strongly on the operators and the parameters, design of experiment was done for set parameters. Deterministic

method of the Lingo software's B&B algorithm was used to evaluate the results of both metaheurstic algorithms. The results indicated that proposed metaheurstic algorithms have better performance in quality of final answer and solving time against the method of Lingo software's B&B. For future research, considering machine capacity and costs in stochastic CF problem are offered.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix A

The reliability function is expressed as

$$R(t) = P(T > t) = \int_t^\infty f(x)\mathrm{dx}$$

where $f(x)$ is the failure probability density function and t is the length of the period of time (which is assumed to start from time zero). On the other hand, the MTBF can be defined in terms of the expected value of the density function $f(x)$,

$$\mathrm{MTBF} = \int_0^\infty xf(x)\mathrm{dx}.$$

Thus,

$$\mathrm{MTBF} = \int_0^\infty xf(x)\mathrm{dx} = \int_0^\infty P(T > x)\mathrm{dx} = \int_0^\infty R(x)\mathrm{dx}.$$
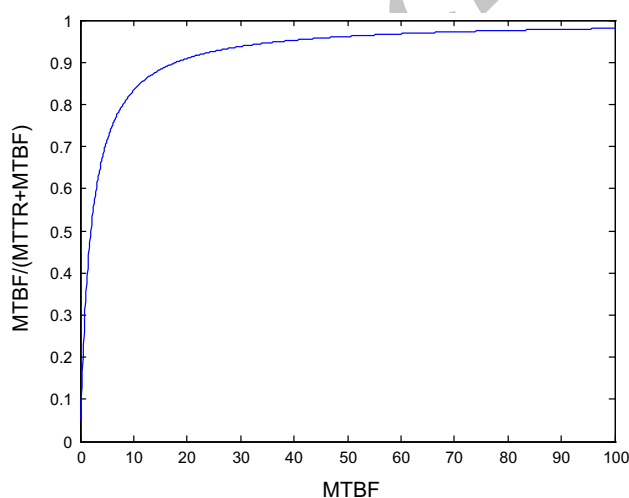

Fig. 16 The relationship between MTBF and $\frac{\mathrm{MTBF}}{\mathrm{MTTR} + \mathrm{MTBF}}$

The reliability has straight relationship with the mean time between failures. Hence, reduction of the reliability reduces the right side of the constraint (9). Figure 16 is presented for more illustration of the relationship between MTBF and coefficient $\frac{\mathrm{MTBF}}{\mathrm{MTTR}+\mathrm{MTBF}}$ of the constraint (9) in the constant value of MTTR.

References

Ameli MSJ, Arkat J (2008) Cell formation with alternative process routings and machine reliability consideration. Int J Adv Manuf Technol 35:761–768

Ameli MSJ, Arkat J, Barzinpour F (2008) Modelling the effects of machine breakdowns in the generalized cell formation problem. Int J Adv Manuf Technol 39:838–850

Ariafar Sh, Ismail N, Tang SH, Ariffin MKAM, Firoozi Z (2011) A stochastic facility layout model in cellular manufacturing systems. Int J Phys Sci 6:3666–3670

Ariafar Sh, Ismail N, Tang SH, Ariffin MKAM, Firoozi Z (2012) The reconfiguration issue of stochastic facility layout design in cellular manufacturing systems. Int J Serv Oper Manag 11:255–266

Arkat J, Naseri F, Ahmadizar F (2011) A stochastic model for the generalized cell formation problem considering machine reliability. Int J Comput Integr Manuf 24:1095–1102

Cao D, Chen M (2005) A robust cell formation approach for varying product demands. Int J Prod Res 43:1587–1605

Chung SH, Wu TH, Chang CC (2011) An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. Comput Ind Eng 60:7–15

Cruz FRB, Van Woense lT, Smith JMG (2010) Buffer and throughput trade-offs in M/G/1/K queuing networks: A bi-criteria approach. Int J Production Econom 125:224–234

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. The sixth international symposium on micro machine and human science, Nagoya, pp 39–43

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

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

Egilmez G, Suer GA (2014) The impact of risk on the integrated cellular design and control. Int J Prod Res 52:1455–1478

Egilmez G, Suer GA, Huang J (2012) Stochastic cellular manufacturing system design subject to maximum acceptable risk level. Comput Ind Eng 63:842–854

Fardis F, Zandi A, Ghezavati VR (2013) Stochastic extension of cellular manufacturing systems: a queuing-based analysis. J Ind Eng Int 9:1–8

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

Ghezavati VR (2011) A new stochastic mixed integer programming to design integrated cellular manufacturing system: A supply chain framework. Int J Ind Eng Comput 2:563–574

Ghezavati VR, Saidi-Mehrabad M (2010) Designing integrated cellular manufacturing systems with scheduling considering

stochastic processing time. Int J Adv Manuf Technol 48:701–717

Ghezavati VR, Saidi-Mehrabad M (2011) An efficient hybrid self-learning method for stochastic cellular manufacturing problem: A queuing-based analysis. Expert Syst Appl 38:1326–1335

Jayakumar V, Raju R (2011) A multi-objective genetic algorithm approach to the probabilistic manufacturing cell formation problem. S Afr J Ind Eng 22:199–212

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

King JR, Nakornchai V (1982) Machine-component group formation in group technology: review and extension. Int J Production Res 20:117–133

Madhusudanan Pillai V, Chandrasekharan MP (2010) Manufacturing cell formation under probabilistic product mix. Int Conf Ind Eng Eng Manag (IEEM) Macao, 580–584

Mahdavi I, Paydar MM, Solimanpur M, Heidarzade A (2009) Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. Expert Syst Appl 36:6598–6604

Rabbani M, Jolai F, Manavizadeh N, Radmehr F, Javadi B (2012) Solving a bi-objective cell formation problem with stochastic production quantities by a two-phase fuzzy linear programming approach. Int J Adv Manuf Technol 58:709–722

Rao RV (2011) Advanced Modeling and Optimization of Manufacturing Processes. Springer-Verlag, London

Safaei N, Saidi-Mehrabad M, Jabal-Ameli MS (2008) A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. Eur J Oper Res 185:563–592

Saidi-Mehrabad M, Ghezavati VR (2009) Designing Cellular Manufacturing Systems under Uncertainty. J Uncertain Syst 3:315–320

Seifoddini H (1990) A Probabilistic Model for Machine Cell Formation. J Manuf Syst 9:69–75

Singh N, Rajamani D (1996) Cellular manufacturing systems: design, planning and control, 1st edn. Chapman and Hall, India

Tavakkoli-Moghaddam R, Javadian N, Javadi B, Safaei N (2007) Design of a facility layout problem in cellular manufacturing systems with stochastic demands. Appl Math Comput 184:721–728