Scientia Iranica E (2019) 26(4), 2541–2560

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

A bi-objective bi-level mathematical model for cellular manufacturing system applying evolutionary algorithms

B. Behnia[a], I. Mahdavi[a,*], B. Shirazi[a], and M.M. Paydar[b]

a. *Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, P.O. Box 4716685635, Iran.*
b. *Department of Industrial Engineering, Babol Noshirvani University of Technology, Babol, P.O. Box 4714871167, Iran.*

Received 6 December 2017; received in revised form 18 March 2018; accepted 9 June 2018

KEYWORDS
Cellular manufacturing system;
Bi-level programming approach;
Workers' interest;
Bi-objective optimization;
Goal programming;
Evolutionary algorithms;
TOPSIS method.

Abstract. The present study aims to design a bi-objective bi-level model for a multi-dimensional Cellular Manufacturing System (CMS). Minimization of the total number of voids and balancing of the workloads assigned to cells are regarded as two objectives at the upper level of the model. However, at the lower level, attempts are made to maximize the workers' interest to work together in a particular cell. To this end, two Nested Bi-Level metaheuristics, including Particle Swarm Optimization (NBL-PSO) and a Population-Based Simulated Annealing algorithm (NBL-PBSA), were implemented to solve the model. In addition, the goal programming approach was utilized at the upper level of these algorithms. Further, nine numerical examples were applied to verify the suggested framework, and the TOPSIS method was used to find a better algorithm. Furthermore, the best weights for upper-level objectives were tuned by using a weight sensitivity analysis. Based on computational results of all of the three objectives, when decisions about inter- and intra-cell layouts as well as cell formation were simultaneously made in order to balance the assigned workloads by considering voids and workers' interest, making the problem closer to the real world, the outcomes were found different from their ideal. Finally, NBL-PBSA could perform better than NBL-PSO, which confirmed the efficiency of the proposed framework.

1. Introduction

Nowadays, managers should seek a new and efficient approach to make plans about their facilities to ensure success in the competitive environment. For this purpose, the industrialized countries spend a large sum of money annually just for facility planning. In addition, about 20%-50% of the overall production cost is associated with the transportation of materials. Thus, effective planning can reduce these costs from 10% to 30% [1]. Further, manufacturing systems should be able to produce final products of the highest quality with lower production costs to ensure the timely delivery of products to customers. Furthermore, these systems should adapt or respond quickly to the changes in demand and production without any significant investment [2].

Cellular manufacturing is regarded as one of the basic rules of the group technology to construct the production. Based on this new technology, every cell consists of several machines and production equipment, which can process a group of parts as a family of parts including a similar manufacturing process. Compared to other production systems, cellular manufacturing can play a significant role in reducing transport volume

*. *Corresponding author. Tel.: +98 113 2191205*
E-mail addresses: bardia.behnia@gmail.com (B. Behnia); irajarash@rediffmail.com (I. Mahdavi); shirazi_b@yahoo.com (B. Shirazi); paydar@nit.ac.ir (M.M. Paydar)

doi: 10.24200/sci.2018.5717.1440

and cost, setup time, production time, package size, and the amount of inventory in the manufacturing process [3].

Generally, identification of a set of part types appropriate for producing a group of machines as one of the highly used strategic levels of cell design has attracted much attention. However, many other strategic level issues, including the types and number of tools and fixture, cell layout, machine flexibility level, kind of material handling equipment, human and worker issue, idle time, etc., exist that must be considered and measured as a part of cell design problem [4]. To this end, various approaches are implemented for evaluating CMS such as machine component group analysis [5,6], similarity coefficient-based approaches [7], genetic-algorithm-based approaches [8], neural network-based approaches [6], goal programming [9], dynamic programming [5,10], heuristic-based approaches [11], fuzzy clustering [12], and artificial intelligence approaches [13].

Thus, the present study aims to provide a framework for multi-dimensional CMS. Herein, three dimensions, including part, machine, and worker, are considered, and a bi-objective bi-level model is proposed for modeling the related framework. At the upper level, attempts are made to minimize the total number of voids [14] and balance the workloads assigned to cells [15]. However, at the lower level, attempts are made to maximize workers' interest in working together in a particular cell. To solve the model, two metaheuristics, including NBL-PSO and NBL-PBSA, are implemented. In addition, the goal programming approach is used for these higher-level algorithms. Further, all parameters of these algorithms are tuned by Taguchi method to achieve better performance. Further, numerical examples are applied to demonstrate the efficiency of the approach. Furthermore, a weight analysis is utilized to achieve the optimal weight of each objective function based on the goal programming approach. Finally, the results are presented.

The remainder of this paper is presented as follows. Section 2 gives a brief literature review about the related works. Section 3 provides problem definition and model formulation. Section 4 suggests the solution approaches including encoding and decoding as well as an explanation of algorithms. The numerical examples, parameter tuning, and results are presented in Section 5. Finally, conclusion and future directions are emphasized in Section 6.

2. Literature review

This section presents a brief review of the studies relating to cellular manufacturing as well as a review of bi-level programming. Next, the related gaps are highlighted.

2.1. Cellular manufacturing

Compared to other manufacturing systems, a large number of studies on cellular manufacturing have focused on enhancing the firms' performance to fulfill different purposes such as minimizing costs, decreasing voids and exceptional elements, maximizing the total profit, etc. Some research studies underscored the importance of a particular review of these studies, the reviewing study of Offodile et al. [16], to identify the related gaps. They provided a strong review of the CMS by focusing on some related researches and, then, categorized them based on features, assumptions, and key properties. In addition, they highlighted various models by using a tubular framework to provide directions for future research. In another study, Singh [17] provided a brief literature review of cell formation aspects of CMS in order to identify the effectiveness and limitations of existing methodologies.

Further, Joines et al. [18] proposed an integer programming model for a manufacturing cell design, which was regarded as the case study for others. They considered a unique illustrated structure for the part or machine partitions that reduces the size of the cell formation problem and increases the scale of problems, which can be solved. Furthermore, they used genetic algorithms as a solution approach and proved its efficiency by using several problems based on the literature. Mohammadi and Forghani [19] developed a new layout framework of CMS, called "S-shaped layout", upon which an integrated bi-objective cell formation and layout problem for this suggested layout was formulated. They considered some parameters including the demands of parts, machine dimensions, sequences of operations, and passageway widths. In addition, they attempted to minimize the total intercell handling costs of material and maximize the total similarity between the machines. Further, a weighted sum method, a hybrid simulated annealing, and dynamic programming method as the solution approaches were suggested.

Bootaki et al. [20] proposed two different criteria: workers-experts and relations. They formulated a bi-objective mathematical model for minimizing the voids of worker-machine, as well as those of worker-worker, and used the ε-constraint method and NSGAII algorithm for finding optimal Pareto fronts. Aalaei and Davoudpour [21] proposed a robust mathematical model for minimizing the total costs. The demands of products under uncertainty were also pointed out, upon which three scenarios were suggested for this purpose. Additionally, a case study was reported in typical equipment manufacturing for the parameter setting of the model. In summary, a brief review of the recent related work is provided in Table 1 to identify the research gaps.

As shown in Table 1, worker allocation is dis-

Table 1. A brief review of CMS.

Ref.	**Year**	**Objective(s)**	**Solution approach**	**Model type**	A[a]			B[e]		C[h]	D[i]		**# of attributes**
					$A1$[b]	$A2$[c]	$A3$[d]	$B1$[f]	$B2$[g]		$D1$[j]	$D2$[k]	
Mahdavi et al. [14]	2012	Minimizing voids and exceptional elements	Branch and bound via LINGO	SO[l]	✓	✓	✓	✓			✓	✓	6
Hosseini et al. [15]	2016	Minimizing the total cost and balancing the assigned workloads of cells	Multi-choice goal programming, genetic algorithm	BO[m]	✓	✓		✓			✓	✓	5
Imran et al. [22]	2017	Minimizing the value-added work in process	Hybrid genetic algorithm	SO	✓	✓		✓				✓	4
Aalaei [21]	2017	Minimizing the total cost	Robust optimization	SO	✓	✓	✓		✓			✓	5
Delgoshaei and Gomes [23]	2016	Minimizing the total cost	Metaheuristics, Branch and bound	SO	✓	✓			✓			✓	4
Bootaki et al. [20]	2016	Minimizing the voids of worker-machine, minimizing the voids of worker-worker	ε-constraint, NSGAII algorithm	BO		✓	✓	✓				✓	4
Aljuneidi and Bulgak [24]	2017	Minimizing the total cost	CPLEX	SO	✓	✓		✓				✓	4
Jawahara and Subhan [25]	2017	Minimizing the total cost	Metaheuristics	SO	✓	✓		✓				✓	4
Kuo and Liu [26]	2017	Minimizing the total required workers	LINGO	SO	✓	✓	✓	✓				✓	5
Forghani and Mohammadi [8]	2014	Minimizing total material handling cost	Genetic algorithm	SO	✓	✓		✓				✓	4
Rabbani et al. [27]	2016	Minimizing total cost maximizing labor utilization	Genetic algorithm	BO	✓	✓		✓		✓		✓	5
Rabbani et al. [28]	2017	Minimizing costs and production waste	Ant colony optimization	SO	✓	✓		✓				✓	4
Mahootchi et al. [29]	2017	Minimizing the expected total variable cost	GAMS	SO	✓	✓			✓			✓	4
This study	—	Minimizing voids and exceptional elements, balancing the workloads assigned to cells, and maximizing workers' interest to work together	Goal programming, particle swarm optimization and simulated annealing	BO-BL[n]	✓	✓	✓	✓		✓	✓	✓	7

[a]A: Considered dimension; [b]A1: Part; [c]A2: Machine; [d]A3: Worker; [e]B: Environment type; [f]B1: Deterministic; [g]B2: Stochastic; [h]C: Processing time; [i]D: Movement; [j]D1: Intera-cell; [k]D2: Inter-cell; [l]SO: Single-Objective; [m]BO: Bi-Objective; [n]BL: Bi-Level.

regarded in most of the recent studies. Allocation of workers to cells and various machines is a tactical decision, while the cell formation is regarded as a strategic decision. Therefore, these two different problems should not be considered as centralized planning. Accordingly, a bi-level model is presented to make decentralized decisions. To the best of our knowledge, no study has focused on the bi-level concept so far. Due to the importance of reducing voids and exceptional element in the cellular manufacturing problem, a leader is considered at the first level, and the allocation of human resources as followers is regarded at the second level.

As shown in Table 1, metaheuristic algorithms used in some studies are applied according to the NP-hardness of these field models. In addition, maximization of workers' interest and balancing of the workloads assigned to cells have been hardly considered by different studies. Thus, no study has focused on adopting a bi-objective bi-level model for minimizing voids and exceptional elements, balancing the workloads assigned to cells, and maximizing the workers' interest together.

2.2. Bi-level programming

Bi-level programming has been planned for hierarchical decision levels and interaction between two decision-makers (DMs). The DM at the upper level (the leader) attempts to optimize the objective function for a set of constraints by considering the optimal solution of the DM (the follower) at the lower level. The general formulation of the bi-level programming is presented as follows:

$$(U)\min_{x} F(x, y(x)),$$

$$\text{s.t.} \quad G(x, y(x)) \leq 0,$$

where $y(x)$ is computed by:

$$(L)\min_{y} f(x, y),$$

$$\text{s.t.} \quad g(x, y) \leq 0.$$

As already mentioned, the solution of the lower level is considered as a constraint on the upper level, which is regarded as a key feature of the bi-level programming problem. In addition, in this model, $F(x, y(x))$, $f(x, y)$, $G(x, y)$, and $g(x, y)$ represent the upper-level objective functions, lower-level objective functions, upper-level constraint sets, and lower-level constraint sets, respectively. Further, x and y represent the decision variables of two levels.

So far, a large body of research has emphasized the bi-level programming approach in their model formulation. For example, Saranwong and Likasiri [30] analyzed a bi-level model via a layer iterative method for minimizing the costs as the objective function of each level. Then, five algorithms were suggested based on the layer iterative method and solved by using CPLEX software. Furthermore, a case study in the municipal waste system was reported in this study. In another research, Calvete et al. [31] offered a bi-level mathematical model to plan a distribution network. Similar to the study of Saranwong and Likasiri, each level was used to minimize the related costs. However, some evolutionary algorithms were implemented as the solution approaches in this study. Ma et al. [32] proposed a hybrid priority-based algorithm composed of a nested genetic algorithm and a fuzzy logic controller for the purpose of minimizing the costs at each level. Some other studies, such as [33-35,36], used the bi-level approach in different fields of study.

2.3. The contributions of the study

The main contributions of the present study are as follows:

- ✓ To the best of our knowledge, no study has considered a bi-level bi-objective programming in cellular manufacturing so far;
- ✓ Here, three significant objectives are considered simultaneously. The first level is related to reducing the number of voids, exceptional elements, workloads, and cell balancing. At the second level, promoting a sense of cooperation between the workers is emphasized to maintain an innovative and dynamic organization in the long term;
- ✓ Two novel goal programming-based metaheuristic algorithms are highlighted for solving due to NP-hardness of the model;
- ✓ ANOVA is used to analyze algorithms' behavior;
- ✓ TOPSIS method is applied to find a better algorithm;
- ✓ Finally, a weight sensitivity analysis is performed for presenting two goals at the upper level.

3. Problem description

In this section, a bi-objective bi-level mathematical model for multi-dimensional CMS is formulated to achieve an optimal solution. In this study, this model is conducted on cellular manufacturing based on a three-dimensional matrix containing part, machine, and worker. First, the number of cells, machines, parts, and workers are determined; accordingly, the machines assigned to each cell, the workers assigned to each cell, and the assigned parts of each cell are bounded. Then, the assigned machines to each cell are fixed; however, the parts and workers can be shipped to various cells. In addition, a binary parameter is provided to represent the workers' interest in working together. Further, the daily production capacity of each part and the working

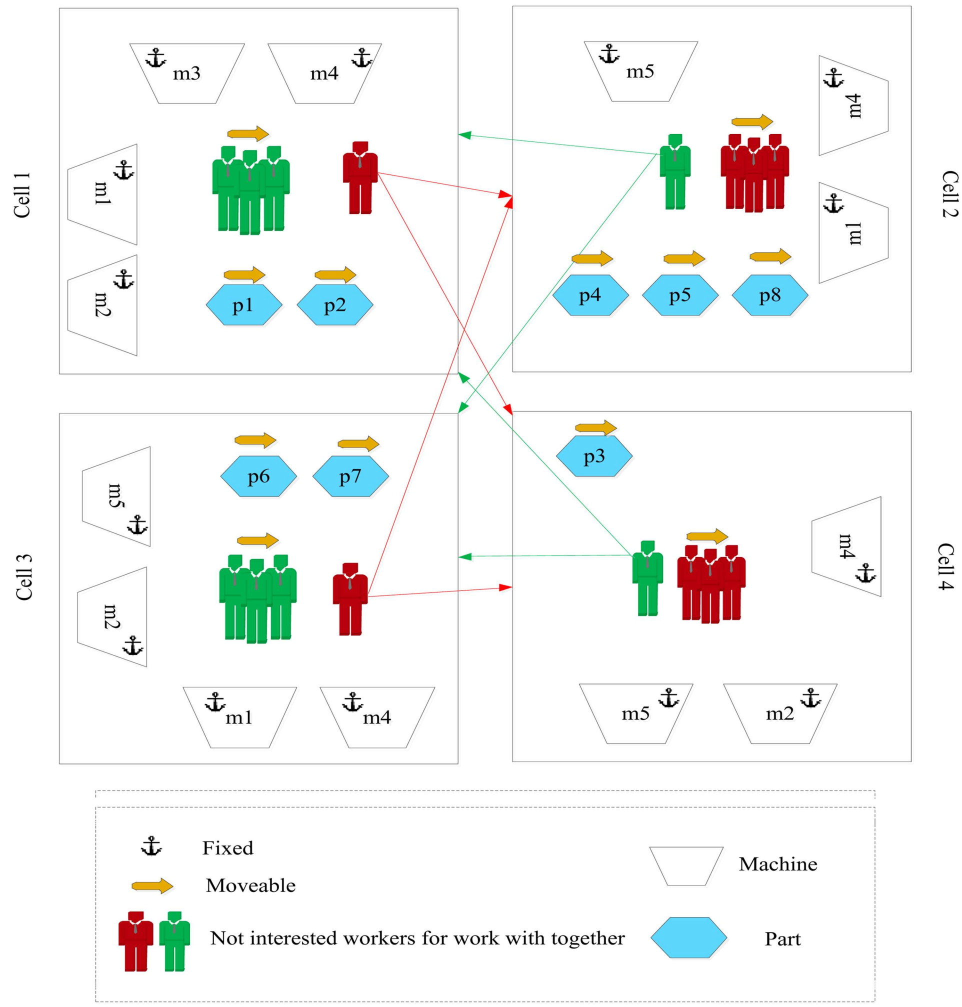

Figure 1. Main structure of the proposed CMS.

time of each worker for each machine are considered to process each part. Furthermore, it is assumed that the time required to produce parts should not exceed the available time of the machines. It is worth noting that these mentioned times are finite. Figure 1 illustrates the main structure of the proposed multi-dimensional CMS.

As shown in Figure 1 and due to the high cost of moving, machines are deployed in a cell, although the parts and workers can move between cells, if required. Then, the parts are allocated to the cells to reduce the voids and exceptional elements. Cell balancing is regarded as another issue, which should be emphasized in this allocation. Workers are assigned to the cells based on their ability as well as their interest to work with others. Finally, the workload of workers and cell balancing are effective in these assignments.

3.1. Proposed mathematical model

Herein, at first, related definitions of the model's parameters and decision variables are provided; then, the suggested bi-objective bi-level mathematical model is formulated.

Subscripts

i Index for part type ($i = 1, 2 \cdots, P$);

w Index for worker ($w = 1, 2, \cdots, W$);

m Index for machine type ($m = 1, 2, \cdots, M$);

k Index for cell ($k = 1, 2, \cdots, C$).

Parameters

A_{im} 1 if part type i needs machine type m; 0 otherwise;

B_{imw} 1 if part type i can be processed on machine type m with worker w; 0 otherwise;

LM_k Minimum number of machine types to be assigned to each cluster;

LP_k Minimum number of part types to be assigned to each cluster;

LW_k Minimum number of workers to be assigned to each cluster;

UW_k Maximum number of workers to be assigned to each cluster;

$R_{ww'}$ 1 if worker w is interested in working with worker w'; 0 otherwise;

T_{imw} Working time of worker w on machine m for processing of part i;

AT_m Available time of machine m;

PN_i The daily production capacity of part i.

Decision variables

x_{mk} 1 if machine type m is assigned to cell k; 0 otherwise;

y_{ik} 1 if part i is assigned to cell k; 0 otherwise;

z_{wk} 1 if worker w is assigned to cell k; 0 otherwise;

d_{imwk} 1 if part i is processed by machine type m with worker w in cell k; =0 otherwise.

In general, based on the nature of the bi-level programming, the levels should be related to each other by using some variables. As illustrated in Figure 2, z_{wk} represents the only variable generated at the lower level and is used at the upper level of formulation, while d_{imwk} is generated at the upper level and is utilized at the lower level of formulation. Thus, these two variables are considered as two dependent factors between the two levels.

The suggested bi-objective bi-level mathematical model is first formulated after describing the related subscripts, parameters, and decision variables. At the upper level of this model, two minimization objective functions are available, while only one maximization objective function is included at the lower level. The proposed bi-level model is presented as follows:

Upper level

$$\min \quad Z_1 = \sum_{k=1}^{C}\left[\sum_{i=1}^{P}\sum_{m=1}^{M}\sum_{w=1}^{W} y_{ik}x_{mk}z_{wk} - \sum_{i=1}^{P}\sum_{m=1}^{M}\sum_{w=1}^{W} y_{ik}x_{mk}z_{wk}d_{imwk}\right] \tag{1-1}$$

$$+\sum_{i=1}^{P}\sum_{k=1}^{C}\sum_{m=1}^{M}\sum_{w=1}^{W}[y_{ik}x_{mk}(1-z_{wk})d_{imwk}] \tag{1-2}$$

$$+\sum_{i=1}^{P}\sum_{k=1}^{C}\sum_{m=1}^{M}\sum_{w=1}^{W} [2\times x_{mk}(1-y_{ik})(1-z_{wk})d_{imwk}] \tag{1-3}$$

$$+\sum_{i=1}^{P}\sum_{k=1}^{C}\sum_{m=1}^{M}\sum_{w=1}^{W}[x_{mk}(1-y_{ik})z_{wk}d_{imwk}], \tag{1-4}$$

$$\min \quad Z_2 = \sum_{k=1}^{C}\left|\sum_{i=1}^{P}\sum_{m=1}^{M}\sum_{w=1}^{W} d_{imwk}T_{imw}PN_i - \frac{1}{C}\sum_{k=1}^{C}\sum_{i=1}^{P}\sum_{m=1}^{M}\sum_{w=1}^{W} d_{imwk}T_{imw}PN_i\right|. \tag{2}$$

Constraints:

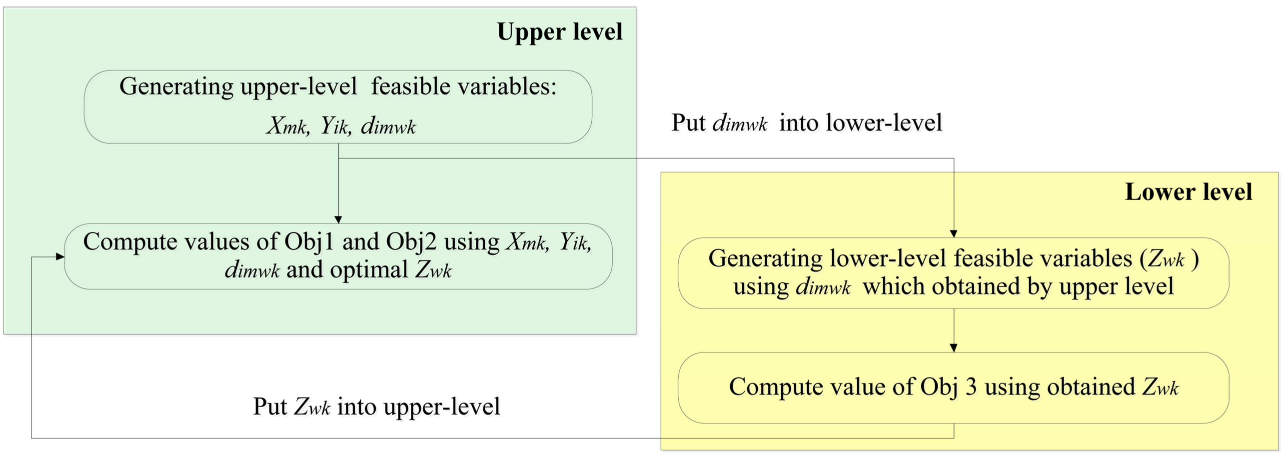

Figure 2. The relationship between the two levels.

$$\sum_{k=1}^{C} y_{ik} = 1 \qquad \forall\, i, \tag{3}$$

$$\sum_{m=1}^{M} x_{mk} \geq LM_k \qquad \forall\, k, \tag{4}$$

$$d_{imwk} \leq B_{imw} x_{mk} \qquad \forall\, i, m, w, k, \tag{5}$$

$$\sum_{k=1}^{C}\sum_{w=1}^{W} d_{imwk} = A_{im} \qquad \forall\, i, m, \tag{6}$$

$$\sum_{i=1}^{P} y_{ik} \geq LP_k \qquad \forall\, k, \tag{7}$$

$$\sum_{i=1}^{P}\sum_{w=1}^{W}\sum_{k=1}^{C} d_{imwk} PN_i T_{imw} \leq AT_m \qquad \forall\, m, \tag{8}$$

$$x_{mk}, y_{ik}, z_{wk}, d_{imwk} \in \{0, 1\} \qquad \forall\, i, m, w, k. \tag{9}$$

The first objective function of the upper level consists of four segments. Segment (1-1) attempts to minimize the total number of voids. In addition, the number of exceptional elements is considered in Segments (1-2), (1-3), and (1-4). It is worth noting that these exceptional elements are related to the availability of relevant workers and machines.

The second objective function (2) that attempts to balance the workloads assigned to all cells by minimizing the difference between the total workload of each cell and the average workload of all the related cells includes two goals. First, the assigned workload of each cell is calculated; second, the average workload of all cells is computed.

According to Eq. (3), each part is assigned to one cell only. Based on Inequality (4), the total number of specific machines assigned to cells is limited, which is equal to or more than the minimum number of machines to be assigned to each cluster. Inequality (5) shows that when machine m is not in cell k, then $d_{imwk} = 0$. Constraint (6) ensures that if part i is required to be processed by machine m, there is only one cell like k that includes this machine and worker w, who worked for processing part i in this cell. According to Eq. (7), the minimum number of parts that is processed in each cell is equal to or more than the minimum number of part types, which is assigned to each cluster. Based on Inequality (8), the required time to produce the parts should not exceed the available time of the machines. Finally, Eq. (9) shows binary decision variables.

The lower level aims to maximize the workers' interest to work together in a particular cell.

Lower level

$$\max \quad Z_3 = \sum_{w=1}^{W}\sum_{w'=1}^{W}\sum_{k=1}^{C} R_{ww'} z_{wk} z_{w'k}. \tag{10}$$

Constraints:

$$\sum_{k=1}^{C} z_{wk} = 1 \qquad \forall\, w, \tag{11}$$

$$\sum_{w=1}^{W} z_{wk} \leq UW_k \qquad \forall\, k, \tag{12}$$

$$\sum_{w=1}^{W} z_{wk} \geq LW_k \qquad \forall\, k, \tag{13}$$

$$\sum_{i=1}^{P}\sum_{m=1}^{M} d_{imwk} \geq z_{wk} \qquad \forall\, w, k. \tag{14}$$

According to Eq. (11), a worker is assigned to one cell only. Inequalities (12) and (13) ensure that the assigned workers to each cell should be between the proposed minimum and maximum bound. Finally, Inequality (14) is defined to ensure that there is at least one part like i in this cell if worker w is assigned to cell k, which is processed by machine type m by worker w working on that machine.

3.2. Final model (goal programming)

In this study, two upper-level objective functions and one deviation-based objective were used to solve the proposed model via applied evolutionary algorithms. To this end, the offered model should be rewritten based on the goal programming procedure. Leung and Chan [37] described the main explanation of this well-known method. The GP version of the model is as follows:

Upper level

$$\min \quad \text{TotalDev} = \omega_1 \left(\frac{(d_1^+ + d_1^-)}{\text{goal}_1} \right) + \omega_2 \left(\frac{(d_2^+ + d_2^-)}{\text{goal}_2} \right). \tag{15}$$

Subject to:

Constraints (3) to (9)

$$Z_1 - d_1^+ + d_1^- = \text{goal}_1, \tag{16}$$

$$Z_2 - d_2^+ + d_2^- = \text{goal}_2, \tag{17}$$

$$d_1^+, d_1^-, d_2^+, d_2^- \geq 0. \tag{18}$$

Lower level
Second-level objective function (10)
Subject to:
Constraints (11) to (14).

Goal_1 and goal_2 indicate the aspiration levels of the first and second goals at the upper level, respectively. In addition, parameters ω_1 and ω_2 are considered as the weights of these two targets; d_i^+ and d_i^- are defined as the positive and negative deviation variables. Here, Eq. (15) attempts to minimize the total deviation of the targets at two upper levels. Further, each of these deviations is divided by their aspiration to normalize these two different kinds of deviation. Constraints (16) and (17) calculate the amount of positive and negative deviations of objectives with respect to their aspiration level. Eq. (18) demonstrates the positive variables. It is worth noting that other constraints, along with the lower-level objective function, are similar to those in the proposed model.

4. Solution approaches

The solution approaches used in this study include encoding and decoding, nested bi-level population-based simulated annealing, and nested bi-level particle swarm optimization. The NP-hardness of the bi-level programming problems was confirmed [36], which is regarded as the main reason for using these metaheuristics instead of exact methods.

4.1. Encoding and decoding

Different methods are available for encoding mathematical models such as priority-based encoding [38], Michalewicz et al.'s matrix [39], and Prüfer number [40]. In the present study, each vector was first generated by random numbers between [0, 1] and, accordingly, the sequence was achieved by the priority-based encoding method. Figure 3 displays the proposed chromosome of this study. As shown, the written number in each gene of the chromosome represents the cell number to which the machine, part, or worker ought to be assigned.

In addition, two procedures were provided to find the required vectors of **X**, **Y**, **d**, **Z**. Due to the similarity in allocating **X**, **Y**, **Z** vectors, their main allocation procedure is presented in Figure 4, and the allocation of **d** is provided in Figure 5.

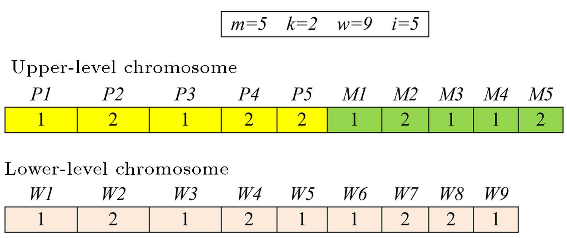

Figure 3. The proposed chromosome of this research.

Inputs: I: Set of sources
K: Set of cells
V_I: Encode solution (chromosome)
Outputs: Xaloc (I, k): Relation between source and cell (0 or 1)
For 1 to i
Step 1: Xaloc $(I, V(I)) = 1$
Step 2: Other Xaloc $(I, k) = 0$
End for

Figure 4. The allocation procedure of $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ vectors.

4.2. NBL-PBSA

Simulated annealing is known as one of the strong solution-based metaheuristic algorithms. This efficient algorithm was developed by Kirkpatrick et al. [41] to solve the optimization problem. Different studies [42] have implemented the hybrid version of this algorithm. In addition, some researchers [43] developed the population-based version of the simulated annealing algorithm. In this paper, an NBL-PBSA algorithm was developed to solve the proposed mathematical model. In this algorithm, some of the well-known mutation operators were utilized for local search such as swap operator, displacement operator, insertion operator, and reversion operator [44]. Finally, Figures 6 and 7 illustrate the pseudocode of two levels in the developed NBL-PBSA.

As shown in Figure 5, Goal_1 and Goal_2 are obtained via running each objective of the upper level separately. For this purpose, two PBSAs with a single objective function were developed. In addition, the algorithm parameters consist of $Maxiter$, $Npop$, T_0, and α that should be tuned to achieve better performance.

4.3. NBL-PSO

Particle swarm optimization algorithm, as a well-known metaheuristic model, was employed, and its results were compared with those of the suggested NBL-PBSA. This evolutionary algorithm was proposed by Eberhart and Kennedy [45] and modeled based on the social behavior of birds' flocks. The main formulas for this algorithm are presented as follows:

$$V_{ij}(t+1) = W \times V_{ij}(t) + c_1 r_{1j}(t)[p_{ij}(t) - x_{ij}(t)]$$
$$+ c_2 r_{2j}(t)[g_j(t) - x_{ij}(t)], \tag{19}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \tag{20}$$

where Eq. (19) computes the velocity of each particle, and the position of each particle can be updated via Eq. (20). More details are presented in [44,45].

Inputs: I: Set of parts
M: Set of machines
W: Set of workers
K: Set of cells
A_{im}: If part type i needs to machine type m is 1, otherwise is 0
B_{imw}: If part type i can be processed on machine type m with worker w, otherwise is 0
X_{mk}: 1 if machine type m is assigned to cell k, otherwise is 0
Outputs: d_{imwk}: Relation between part, machine, worker and cell (0 or 1)
For $i \in I$, $k \in K$, $m \in M$, $w \in W$
If $X_{mk} = 1$ && $B_{imw} = 1$ && $A_{im} = 1$
Step 1: $d_{imwk} = 1$
End if
Step 2: Other $d_{imwk} = 0$
End for
Do a feasibility check for singularity of assigned workers

Figure 5. The allocation procedure of $\boldsymbol{d}$ vectors.

Input: $\boldsymbol{Maxiter}$, $\boldsymbol{Npop}$, $\boldsymbol{T_0}$ **and** $\boldsymbol{\alpha}$
Population = Generate initial variables vectors (**X, Y, d) for each solution**
While the feasibility conditions are not satisfied
Check the feasibility of vectors
If not feasible→Correct vectors
End while
Perform lower level algorithm for calculating **Z** and $\mathbf{F_2}$
For each solution
Compute $\mathbf{F_{11}}$ and $\mathbf{F_{12}}$ by using **Z**, **X**, **Y**, **d**
Compute $\boldsymbol{GP} = ((\boldsymbol{W_1} \times \boldsymbol{F_{11}})/\mathbf{Goal_1}) + ((\boldsymbol{W_2} \times \boldsymbol{F_{12}})/\mathbf{Goal_2})$
End for
Best$_{sol}$=The solution among $\boldsymbol{Npop}$ solutions with minimum $\boldsymbol{GP}$
$\boldsymbol{T = T_0}$
For $i = 1$ to $\boldsymbol{Maxiter}$
For each solution
New population = Create neighbor population (local search)
Update variables vectors (**X**, **Y**, **d**) for each solution
While the feasibility conditions are not satisfied
Check the feasibility of vectors
If not feasible→Correct vectors
End while
End for
Perform **lower-level algorithm** for calculating **Z** and $\mathbf{F_2}$
For each solution
Compute $\mathbf{F_{11}}$ and $\mathbf{F_{12}}$ by using (**Z**, **X**, **Y**, **d**)
Compute $\boldsymbol{GP}_{\mathrm{New}} = ((\boldsymbol{W_1} \times \boldsymbol{F_{11}})/\mathbf{Goal_1}) + ((\boldsymbol{W_2} \times \boldsymbol{F_{12}})/\mathbf{Goal_2})$
If $\boldsymbol{GP}_{\mathrm{New}} \leq \boldsymbol{GP}$
Population=New population
Else if rand $\sim (0,1) <$ EXP $((\mathbf{GP}_{\mathrm{New}} - \mathbf{GP})/T)$
Population = New population
End if
End for
Update $\mathbf{Best_{\mathit{sol}}}$
$\boldsymbol{T = T \times a}$
End for
End.

Figure 6. Pseudocode of the upper-level NBL-PBSA.

Herein, an NBL-PSO algorithm was suggested. Figure 8 illustrates more information about the proposed algorithm.

5. Computational results

5.1. Numerical examples

In the present study, to verify the proposed model and suggested approaches, nine different test problems were designed where p, m, c, and w are defined, as shown in Table 2.

The values of A_{im}, B_{imw}, and $R_{ww'}$ are used randomly. The values of LM_c, LP_c, LW_c, and UW_c for all cells are similar and are assigned based on Table 3. In addition, the amount of T_{imw} and PN_i follows a uniform distribution, and AT_m is determined by experts.

Table 2. Specifications of test problems.

Problem	Part type (p)	Machine type (m)	Cell number (c)	Worker (w)
1	5	5	2	9
2	8	10	3	12
3	9	7	3	10
4	10	15	4	18
5	15	12	3	14
6	18	11	3	15
7	20	12	4	15
8	25	15	4	20
9	30	20	4	20

```
Input: Maxiter, Npop, T0 and α
Population = Generate initial variable vector (Z) for each solution
While the feasibility condition are not satisfied
Check the feasibility of vector Z
If not feasible→Correct vector
End while
For each solution
Compute F2 by using (Z,d)
End for
Best_sol = The solution among Npop solution maximum F2
T = T0
For i = 1 to Maxiter
For each solution
New population = Create neighbor population (local search)
Update variable vector (Z) for each solution
While the feasibility conditions are not satisfied
Check the feasibility of vector Z
If not feasible→Correct vectors
End while
Compute New F2 by using (Z,d)
If New F2 ≥ F2
Population = New population
Else if rand ~ (0,1) < Exp ((F2 − NewF2)/T)
Population=New population
End if
End for
Update Best_sol
T = T × α
End for
End.
```

Figure 7. Pseudocode of the lower-level NBL-PBSA.

5.2. Parameter tuning

Taguchi experiment is implemented to tune the parameter setting [46]. This approach is applied, instead of the full factorial experiment. In general, in a single-objective optimization problem, only the objective function can be used as a response of Taguchi. Further, the combination of standard measurement metrics as the response of Taguchi can be implemented in a multi-objective optimization problem. However, the present study utilizes a bi-level model with two objective functions at the upper level that changed the single objective problem via Goal Programming (GP) approach. Finally, the upper-level objective function (GP) was only applied to the response of Taguchi experiment based on the research of Kuo et al. [47]. Since the first-level objective function (GP) is regarded as the minimization problem, "the less is better" ratio is used among the proposed Taguchi relationships as illustrated below:

Table 3. The value of the model's parameters.

Parameter	Value	Unit
LM_c	2	Machine
LP_c	2	Part
LW_c	3	worker
UW_c	6	worker
A_{im}	0 or 1 randomly	—
B_{imw}	0 or 1 randomly	—
$R_{ww'}$	0 or 1 randomly	—
T_{imw}	Uniform $\sim (1,3)$	Minute
AT_m	500 or 600	Minute
PN_i	Uniform $\sim (400,600)$	part

$$S/N = -10\log\left(\frac{\sum Y^2}{n}\right). \tag{21}$$

In this regard, the proposed level of each algorithm should be first defined in order to find the proposed level of each parameter. The studies of [43,44] were used for PBSA and PSO, respectively, to find a proper level of each parameter. The selected levels are presented in Table 4.

After performing the Taguchi experiment via Minitab software, orthogonal arrays L^9 and L^{27} were designed for NBL-PBSA and NBL-PSO, respectively. Thus, the parameters of each algorithm were tuned for each test problem for nine test problems, separately. In addition, each problem was executed 10 times, the mean of which was used. Tables 5 and 6 indicate the results.

Finally, the proper level of parameters for each test problem is obtained, as shown in Table 7, which are adapted to those in Tables 5 and 6. Thus, the obtain proper values can be used as indicators of parameters in the process of implementing each test problem.

5.3. Results

To solve these test problems, the two metaheuristic algorithms were coded in the environment of

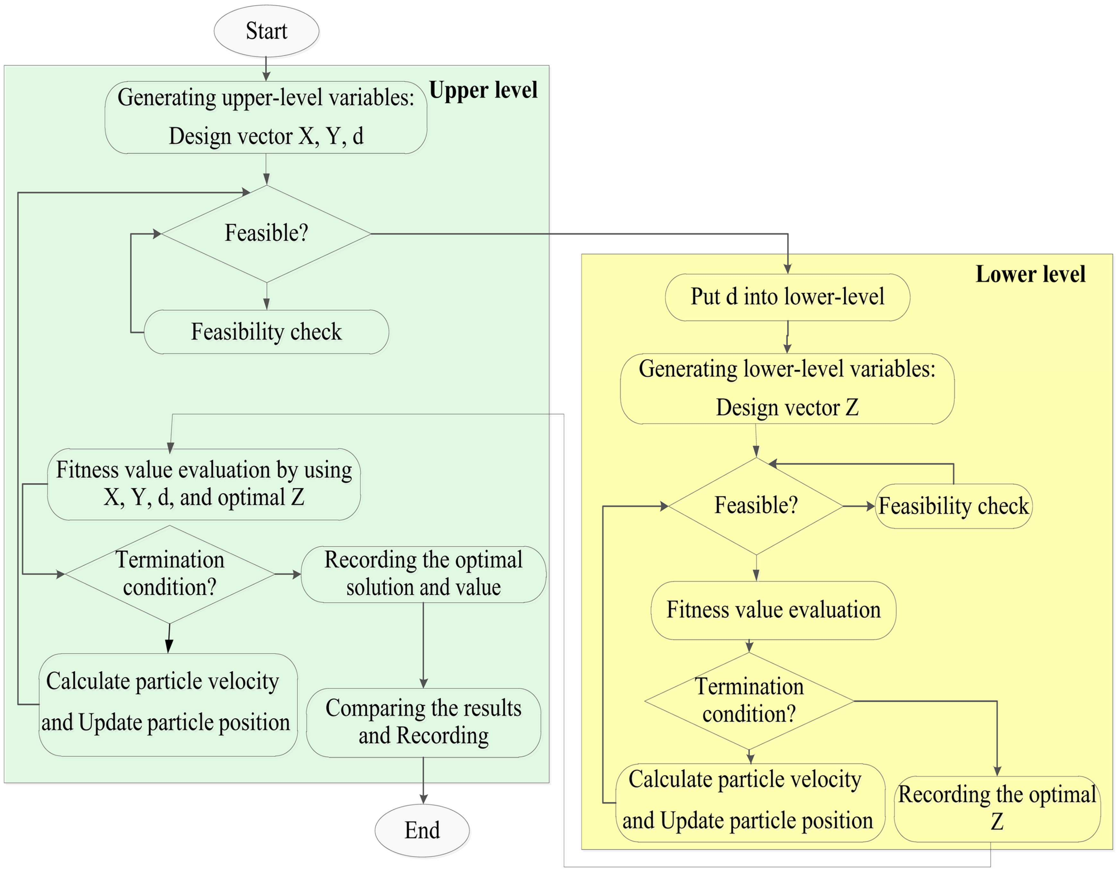

Figure 8. Flowchart of the proposed NBL-PSO.

Table 4. Ranges of algorithms parameters along with their levels.

Nested bi-level algorithms	Parameters	Parameter levels		
		Level 1	Level 2	Level 3
NBL-PBSA	α	0.7	0.8	0.98
	T_0	50	100	150
	N-pop	50	100	150
	Max-iteration	100	250	350
NBL-PSO	$C1$	0.5	1	2
	$C2$	0.5	1	2
	W	0.5	0.75	1
	N-pop	50	100	150
	Max-iteration	100	250	350

MATLAB$^{\text{TM}}$ software, and all computational experiments were executed by an operating system equipped with Windows 10, Intel Core i7, 2.80 GHz, and 16 GB RAM. Table 8 indicates the results.

As shown, the upper-level objective functions (Objective 1 and Objective 2) are defined as the minimization functions, and their minimum values are considered properly. In addition, the upper-level unifier objective function (GP) is related to the minimization function, while the lower-level objective function (Objective 3) is related to the maximization function. In addition, the minimum value of CPU time has a better performance. It is worth noting that the best values of each test problem are shown in Table 8.

As illustrated in Figure 9, the NBL-PBSA has a better performance in terms of Objective 1, while one of these algorithms cannot be fully realized based on Objective 2.

Figure 10 displays the result of goal programming at the upper level. As illustrated, the NBL-PBSA has

Table 5. The value of GP in various test problems for NBL-PSO.

Exp.	$C1$	$C2$	W	N-pop	Max-Iter	Test problems								
						1	2	3	4	5	6	7	8	9
1	0.5	0.5	0.5	100	100	0.200	23.349	8.255	345.151	12.615	45.868	55.604	251.552	80.323
2	0.5	0.5	0.5	100	250	0.162	8.411	3.985	102.323	4.930	13.813	323.844	34.752	41.220
3	0.5	0.5	0.5	100	350	0.181	9.612	4.400	115.781	5.630	15.776	55.639	40.031	95.223
4	0.5	1	0.75	250	100	0.186	14.168	9.057	23.458	8.174	31.908	29.740	49.729	51.702
5	0.5	1	0.75	250	250	0.148	9.445	2.653	39.401	5.721	9.951	69.865	26.361	29.617
6	0.5	1	0.75	250	350	0.156	4.284	4.089	51.684	5.662	6.323	8.875	39.635	52.295
7	0.5	2	1	350	100	0.161	19.316	4.758	52.662	6.211	10.073	26.073	67.414	33.916
8	0.5	2	1	350	250	0.160	9.820	3.267	45.641	4.439	19.052	74.125	30.056	24.576
9	0.5	2	1	350	350	0.134	4.342	2.672	42.481	3.920	35.651	95.719	33.936	65.620
10	1	0.5	0.75	350	100	0.147	14.016	5.173	96.591	10.386	33.467	50.906	130.858	87.361
11	1	0.5	0.75	350	250	0.122	11.795	4.154	33.208	4.488	17.527	19.104	104.234	115.093
12	1	0.5	0.75	350	350	0.127	5.132	1.793	17.506	1.009	7.580	64.083	27.875	49.031
13	1	1	1	100	100	0.201	33.168	7.117	67.896	8.436	19.083	134.705	49.274	64.583
14	1	1	1	100	250	0.178	10.008	6.064	117.493	4.644	11.007	25.841	233.740	99.072
15	1	1	1	100	350	0.152	7.474	3.814	52.435	5.748	7.065	116.448	32.292	42.311
16	1	2	0.5	250	100	0.168	17.986	5.956	57.000	7.117	15.630	20.198	18.906	38.252
17	1	2	0.5	250	250	0.165	7.864	4.796	4531.000	4.416	6.736	65.021	49.917	41.116
18	1	2	0.5	250	350	0.140	4.729	2.604	39.427	4.153	7.279	81.354	20.490	72.794
19	2	0.5	1	250	100	0.158	19.187	7.255	244.828	8.724	12.969	184.594	404.525	73.925
20	2	0.5	1	250	250	0.148	10.701	2.702	144.268	5.457	10.573	85.729	78.729	37.531
21	2	0.5	1	250	350	0.131	12.336	2.739	51.521	6.698	15.468	86.875	51.385	36.598
22	2	1	0.5	350	100	0.167	24.125	3.809	88.537	6.649	21.723	195.956	78.906	137.906
23	2	1	0.5	350	250	0.143	5.956	2.753	292.590	6.754	20.841	20.781	226.988	24.031
24	2	1	0.5	350	350	0.116	4.666	2.497	105.067	4.486	13.087	72.490	82.677	43.961
25	2	2	0.75	100	100	0.193	12.872	11.249	326.666	21.781	19.302	106.250	214.010	56.638
26	2	2	0.75	100	250	0.152	23.741	5.245	40.689	5.453	19.417	119.948	54.885	120.634
27	2	2	0.75	100	350	0.153	4.877	6.926	51.245	6.666	6.506	40.583	131.406	172.128

Table 6. The value of GP in various test problems for NBL-PBSA.

Exp.	α	T	N-pop	Max-Iter	Test problems								
					1	2	3	4	5	6	7	8	9
1	0.7	50	50	100	0.228	2.762	1.398	9.308	6.110	0.238	8.591	15.128	15.265
2	0.7	100	100	250	0.200	2.333	1.258	6.061	5.626	0.186	7.157	13.223	13.578
3	0.7	150	150	350	0.168	2.018	1.225	6.518	5.236	0.182	5.882	12.168	13.163
4	0.8	50	100	350	0.200	2.099	1.098	5.473	5.569	0.185	6.963	12.422	15.369
5	0.8	100	150	100	0.248	2.672	1.561	7.661	5.631	0.209	6.982	14.545	18.503
6	0.8	150	50	250	0.234	2.422	1.287	7.474	5.947	0.188	8.336	13.658	14.080
7	0.98	50	150	250	0.154	2.194	1.273	6.243	5.558	0.160	5.638	12.877	15.560
8	0.98	100	50	350	0.225	2.208	1.338	7.510	5.466	0.198	6.894	13.820	12.645
9	0.98	150	100	100	0.266	2.666	1.461	10.948	6.211	0.250	9.043	17.318	21.383

Table 7. The proper level of parameters for each test problem.

Algorithm	Parameter	Test problem number								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
NBL-PBSA	α	0.7	0.98	0.7	0.8	0.7	0.8	0.8	0.7	0.7
	T	50	50	50	50	100	50	100	50	100
	N-pop	150	150	100	150	150	150	150	150	50
	Max-Iteration	250	350	350	350	350	250	350	350	350
NBL-PSO	C1	2	0.5	1	0.5	1	1	2	0.5	2
	$C2$	0.5	2	0.5	1	0.5	2	0.5	2	2
	W	0.75	0.75	0.5	0.75	1	1	0.5	0.5	0.75
	N-pop	150	150	150	150	150	100	50	100	50
	Max-Iteration	350	350	350	350	350	350	250	350	350

Table 8. The obtained results of two algorithms in various test problems.

Test problem	Objective 1		Objective 2		GP		Objective 3		CPU time	
	PBSA	**PSO**	**PBSA**	**PSO**	**PBSA**	**PSO**	**PBSA**	**PSO**	**PBSA**	**PSO**
1	**59**	**59**	**0.08**	1.08	**0.139**	0.184	15	**27**	1053.39	**739.82**
2	**155**	174	4.85	**4.21**	**1.548**	2.919	32	**39**	6015.88	**3570.13**
3	**112**	118	**1.33**	5.33	**0.862**	2.321	21	**27**	2445.57	**2291.15**
4	**278**	292	92	**22.86**	**7.470**	10.116	50	**66**	17658.42	**10651.91**
5	**310**	376	4.16	**3.33**	**3.478**	5.483	44	**57**	11844.55	**7530.43**
6	**353**	400	**8.67**	13.42	**4.241**	8.859	41	**62**	9983.17	**6125.73**
7	**336**	360	**69.25**	291	**7.058**	91.550	34	**48**	22206.19	**3548.37**
8	**577**	664	**49.44**	53.03	**9.080**	24.592	65	**84**	53674.99	**21099.31**
9	**803**	860	**104**	138	**14.633**	52.942	71	**82**	31729.23	**17826.88**

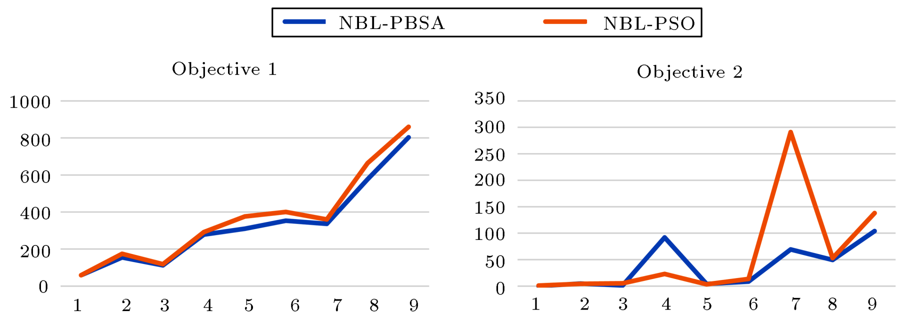

Figure 9. Result of the upper-level objective functions.

a better performance in terms of the upper level in the mathematical model.

In addition, regarding CPU time and Objective 3, the NBL-PSO has a better performance than NBL-PBSA (Figure 11). Further, to compare these two metaheuristics more carefully, ANOVA test was used with a 95% confidence level. In general, ANOVA is used to compare metaheuristic performances, which is common in the literature [48]. To this end, it was used for nine tests in Table 8 based on five criteria including Objective 1, Objective 2, GP, Objective 3, and CPU time, as shown in Table 9. Finally, the mean and standard deviation of each algorithm were obtained for the nine tests with respect to all criteria (Table 10). As shown, no significant difference was observed between the two offered meta-heuristics based on Objective 1, Objective 2, GP, Objective 3, and CPU time. Thus, an MCDM method should be employed to find the best algorithm.

5.3.1. Ranking the algorithms

Selecting the best approach is a difficult task due to the existence of two approaches, including NBL-PSO and NBL-PBSA, and some various criteria, such

Table 9. The results of ANOVA.

Criterion's name	Source	DF	SS	MS	F-statistic	P-value	Test results
Objective 1	Factor	1	5689	5689	0.09	0.764	Null hypothesis is not rejected
	Error	16	972194	60762			
	Total	17	977883				
Objective 2	Factor	1	2189	2189	0.39	0.542	Null hypothesis is not rejected
	Error	16	90079	5630			
	Total	17	92268				
GP	Factor	1	1258	1258	2.58	0.128	Null hypothesis is not rejected
	Error	16	7789	487			
	Total	17	9047				
Objective 3	Factor	1	787	787	1.97	0.179	Null hypothesis is not rejected
	Error	16	6386	399			
	Total	17	7173				
CPU time	Factor	1	384824633	384824633	2.32	0.147	Null hypothesis is not rejected
	Error	16	2654139874	165883742			
	Total	17	3038964507				

as Objective 1, Objective 2, Objective 3, and CPU time. Thus, TOPSIS method, as a well-known MCDM approach, was used to find the best algorithm based on these criteria. The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is one of the most popular multi-criteria decision-making methods, which was proposed by Hwang and Yoon [49]. In addition, more improvements were made to enhance its performance [50]. The main steps used for this technique are presented as follows:

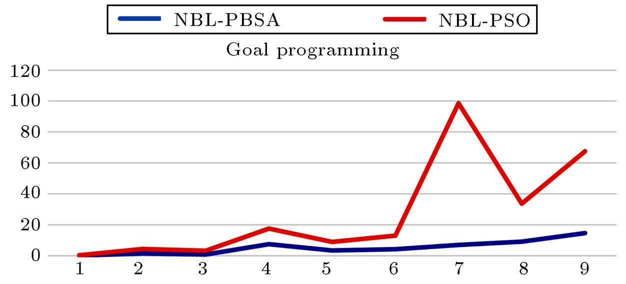

Figure 10. The result of goal programming at the upper level.

Step 1. Create the decision matrix;

Step 2. Calculate the normalized decision matrix by Eq. (22):

$$r_{ij} = \frac{f_{ij}}{\sqrt{\sum_{j=1}^{n} f_{ij}^2}}. \tag{22}$$

In this equation, r_{ij}, f_{ij}, and n represent the normalized decision matrix, the decision matrix, and the number of alternatives, respectively;

Step 3. Obtain the weighted normalized decision matrix by calculating the positive and negative ideal solutions. The positive ideal solution is the largest value of positive criteria and the smallest value of negative criteria, while the negative ideal solution is considered as the smallest value of positive criteria and the largest value of negative criteria;

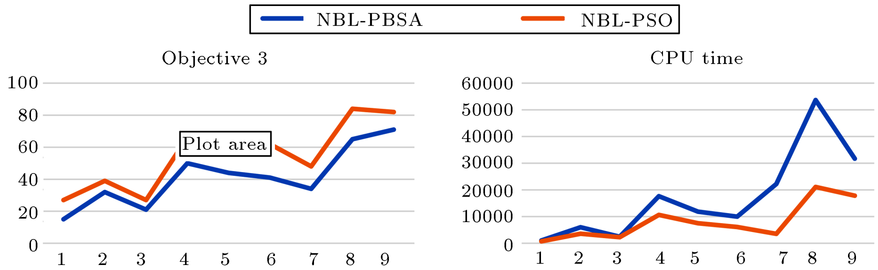

Figure 11. The results of lower-level objective function and CPU time.

Table 10. Mean and standard deviation for each algorithm in nine tests.

Test problem	Objective 1		Objective 2		GP		Objective 3		CPU time	
	PBSA	PSO	PBSA	PSO	PBSA	PSO	PBSA	PSO	PBSA	PSO
Mean	331.4	367.0	37.09	59.14	5.39	22.11	41.44	54.67	17401	8154
St. Dev.	234.5	257.9	42.22	97.35	4.66	30.85	18.61	21.26	16773	7102

Table 11. The average experimental outputs of two algorithms.

Problems	Algorithm	Objective 1	Objective 2	Objective 3	CPU time
Average 1-3	PBSA	108.67	2.09	22.67	3171.61
	PSO	117	3.54	31	2200.37
Average 4-6	PBSA	313.67	34.94	45	13162.05
	PSO	356	13.20	61.67	8102.69
Average 7-9	PBSA	572	74.23	56.67	35870.14
	PSO	628	160.68	71.33	14158.19
Criteria weight		0.4	0.3	0.2	0.1
Sign of criteria		–	–	+	–

Table 12. The result of TOPSIS method for the small-sized problems.

Algorithm	Normalized decision matrix				Weighted normalized decision matrix				d^+	d^-	C_i	Rank
	Obj 1	Obj 2	Obj 3	CPU time	Obj 1	Obj 2	Obj 3	CPU time				
PBSA	0.681	0.508	0.590	0.822	0.272	0.153	0.118	0.082	0.050	0.108	0.683	1
PSO	0.733	0.861	0.807	0.570	0.293	0.258	0.161	0.057	0.108	0.050	0.317	2

Step 4. Calculate the Euclidean distances of the alternatives from the positive and negative ideal solutions obtained in the previous step. These distances are calculated by using Eqs. (23) and (24), respectively:

$$d_i^+ = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^+)^2} \qquad \forall\, i = 1, 2, \cdots, m, \quad (23)$$

$$d_i^- = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_j^-)^2} \qquad \forall\, i = 1, 2, \cdots, m, \quad (24)$$

where d_i^+ and d_i^- represent the distances between the alternative to positive and negative ideal solutions, v_j^+ and v_j^- are regarded as the positive and negative ideal solutions, and v_{ij} indicates the weighted, normalized decision matrix;

Step 5. Compute the relative closeness of each alternative (C_i) by Eq. (25):

$$C_i = \frac{d_i^-}{d_i^- + d_i^+}. \quad (25)$$

Step 6. Rank the alternatives and select the best alternative with the largest C_i.

To this end, the means of the first three rows, the next three rows, and the three end rows as the small-, medium-, and large-sized problems, respectively, are used as the input data by the suggested TOPSIS method. These values are presented with the weights of criteria in Table 11. It is worth noting that the related weights of these criteria are proposed by the related experts.

Tables 12-14 represent the normalized decision matrix, the weighted normalized decision matrix, d^+, d^-, C_i, and the final rank of the algorithms for small-, medium-, and large-sized problems. The results indicated that the proposed NBL-PBSA algorithm is the best algorithm to simultaneously solve small- and large-sized problems in terms of all the criteria, while the proposed NBL-PSO has a better performance only in the case of medium-size problems.

5.3.2. Sensitivity analysis

In the present study, a weight sensitivity analysis was implemented since two goals were considered at the

Table 13. The result of TOPSIS method for the medium-sized problems.

Algorithm	Normalized decision matrix				Weighted normalized decision matrix				d^+	d^-	C_i	Rank
	Obj 1	Obj 2	Obj 3	CPU time	Obj 1	Obj 2	Obj 3	CPU time				
PBSA	0.661	0.935	0.589	0.852	0.264	0.281	0.118	0.085	0.183	0.036	0.163	2
PSO	0.750	0.353	0.808	0.524	0.300	0.106	0.162	0.052	0.036	0.183	0.837	1

Table 14. The result of the TOPSIS method for the large-sized problems.

Algorithm	Normalized decision matrix				Weighted normalized decision matrix				d^+	d^-	C_i	Rank
	Obj 1	Obj 2	Obj 3	CPU time	Obj 1	Obj 2	Obj 3	CPU time				
PBSA	0.673	0.419	0.622	0.930	0.269	0.126	0.124	0.093	0.065	0.149	0.697	1
PSO	0.739	0.908	0.783	0.367	0.296	0.272	0.157	0.037	0.149	0.065	0.303	2

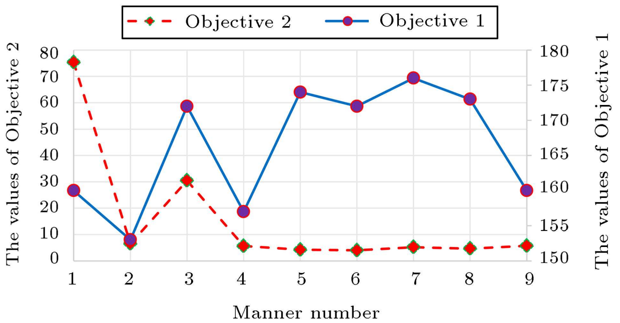

Figure 12. The weight sensitivity analysis of NBL-PSO.

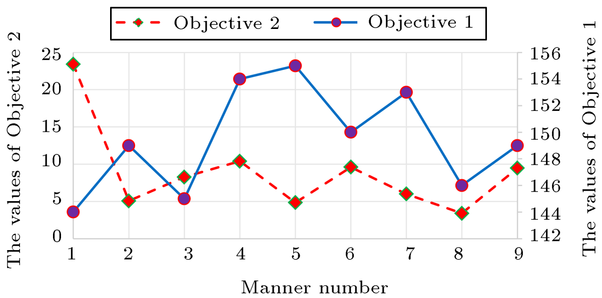

Figure 13. The weight sensitivity analysis of NBL-PBSA.

upper level, and two weights for these goals were used for the goal programming method. To this end, the second test problem and tuned parameters of the algorithm were utilized. The main results of this sensitivity analysis are shown in Table 15 and Figures 12 and 13.

Given that the minimum value of GP is regarded as a proper manner, manners 2 and 8 were considered as the best ones for NBL-MOPSO and NBL-PBSA, respectively (Table 15, Figures 12 and 13).

In addition, NBL-PBSA was implemented for the second test problem, and the achieved chromosome of the upper level and lower level are presented in Figures 14 and 15, respectively. As shown, the first and second rows indicate the obtained random key

Table 15. The weight sensitivity analysis.

Manner	(W_1, W_2)	NBL-MOPSO			NBL-PBSA		
		Objective 1	Objective 2	GP	Objective 1	Objective 2	GP
1	0.9-0.1	160	75.3333	9.5333	144	23.42	2.3932
2	0.8-0.2	**153**	**6.5267**	**2.8553**	149	5.0733	2.1195
3	0.7-0.3	172	30.4667	10.6483	145	8.2733	1.9427
4	0.6-0.4	157	5.6067	3.2052	154	10.4	2.002
5	0.5-0.5	174	4.2133	2.9192	155	4.8467	1.5478
6	0.4-0.6	172	4	2.83333	150	9.6067	1.5904
7	0.3-0.7	176	5.2	3.74	153	6	1.2005
8	0.2-0.8	173	4.6667	3.4542	**146**	**3.4**	**0.754**
9	0.1-0.9	160	5.6933	4.4573	149	9.4667	1.0995

I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
3	2.93	2.16	1.49	2.11	2.74	2.46	1.13	1.46	2.88	2.65	1.13	2.52	1.57	2.88	1.97	1.94	1.70
3	3	2	1	2	3	2	1	1	3	3	1	3	1	3	2	2	2

Figure 14. The obtained upper-level chromosome of NBL-PBSA for the second test problem.

W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}
3	2.91	1	1.72	1.69	1.12	2.55	2.58	1.51	2.67	2.95	2.85
1	3	1	2	2	1	3	3	2	3	3	3

Figure 15. The obtained lower-level chromosome of NBL-PBSA for the second test problem.

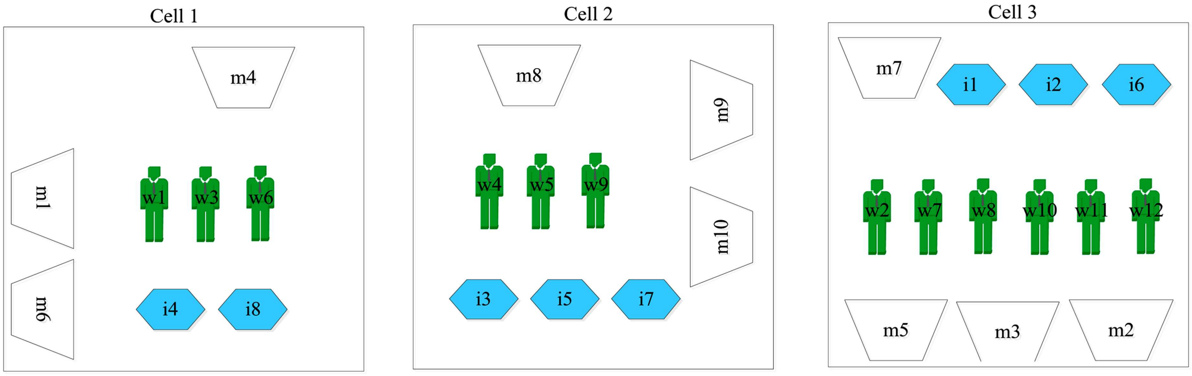

Figure 16. The location layout of parts, machines, and workers of NBL-PBSA for the second test problem.

Table 16. The values of three binary decision variables (x_{mk}, y_{ik}, and z_{wk}) of NBL-PBSA for the second test problem.

	z_{wk}				x_{mk}				y_{ik}		
	k_1	k_2	k_3		k_1	k_2	k_3		k_1	k_2	k_3
w_1	1	0	0	m_1	0	0	1	i_1	0	0	1
w_2	0	0	1	m_2	0	0	1	i_2	1	0	0
w_3	1	0	0	m_3	0	1	0	i_3	0	0	1
w_4	0	1	0	m_4	1	0	0	i_4	1	0	0
w_5	0	1	0	m_5	0	1	0	i_5	0	0	1
w_6	1	0	0	m_6	0	0	1	i_6	0	1	0
w_7	0	0	1	m_7	0	1	0	i_7	0	1	0
w_8	0	0	1	m_8	1	0	0	i_8	0	1	0
w_9	0	1	0	m_9	1	0	0				
w_{10}	0	0	1	m_{10}	0	0	1				
w_{11}	0	0	1								
w_{12}	0	0	1								

and sequence, respectively. Based on these obtained solutions, the location layout of parts, machines, and workers can be formed as in Figure 16. Further, the values of three binary decision variables (x_{mk}, y_{ik}, and z_{wk}) are presented in Table 16.

6. Discussion

After implementing the proposed model via the developed algorithms, the NBL-PSO achieved better performance in terms of CPU time and Objective 3 versus NBL-PBSA, while the NBL-PBSA achieved better performance with respect to GP and Objective 1 versus NBL-PSO (Table 6). However, based on Objective 2, the precedence of one of these algorithms was not fully realized. In other words, it is difficult to select the best approach amid the two approaches (i.e., NBL-PSO and NBL-PBSA) and some various criteria (e.g., Objective 1, Objective 2, Objective 3, and CPU time).

In addition, the TOPSIS method was applied to find the best algorithm in terms of these criteria since the precedence of these two algorithms by ANOVA method could not be fully realized. Based on the obtained results in Tables 12-14, the proposed NBL-

PBSA algorithm was regarded as the best approach to simultaneously solve small- and large-sized problems in terms of all criteria, while the proposed NBL-PSO could perform better only in the case of medium-size problems.

Furthermore, a weight sensitivity analysis was performed. To this end, the second test problem and tuned parameters of the algorithm were utilized. In addition, to report more information, NBL-PBSA is applied per the second test problem, and the achieved chromosome of the upper and lower levels, the location layout of parts, machines, and workers, and the values of three binary decision variables (x_{mk}, y_{ik}, and z_{wk}) are presented in Figures 14-16 and Table 16, respectively.

7. Conclusion

A high-level increase in product quality together with the reduced production costs has always been considered as an important factor in the case of CMS. Considering some issues such as manufacturing technology and machinery layout can ensure appropriate improvements in the cost and quality of products. However, human resources play an important role in enhancing creativity in the final product. Further, staff planning and their interest in cooperating and collaborating with each other is regarded as another important problem. Most problems in the CMS are considered as strategic problems such as determining the optimal facility layout for reducing the voids and exceptional elements, optimal production routing, etc. In the present study, a bi-objective bi-level model was planned for the multi-dimensional CMS. At the upper level, attempts were made to minimize the total number of voids and balance the workloads assigned to the cells. In addition, at the lower level, attempts were made to maximize the workers' interest in working together in a particular cell. Due to the importance of reducing voids and exceptional element in the cellular manufacturing problem, the bi-level concept has been considered as the leader at the first level, and the allocation of human resources as a follower at the second level. To solve the model, NBL-PSO and NBL-PBSA were employed. In this regard, nine test problems were examined using these algorithms based on the best parameters achieved by Taguchi experiments. The goal programming approach was used in the upper-level procedure of these algorithms. The results indicated the efficiency of the proposed approach. However, selecting the best approach was a difficult task, since the precedence of these two algorithms by ANOVA method could not be fully realized. To this end, the TOPSIS method was used to find the best algorithm. Further, a weight sensitivity analysis was performed since two goals were utilized at the upper level, and the goal programming method used two weights for these goals. Finally, the obtained results are useful for DMs and managers. Further work needs to be done for the idle time of machine minimization by considering job sequencing and scheduling through assigning machines to cells for cellular flexible manufacturing systems. In addition, stochastic and robust versions of the model can be used for future studies. Finally, merging cellular manufacturing with a sustainable supply chain is recommended.

References

1. Tompkins, J.A., White, J.A., Bozer, Y.A., and Tanchoco, J.M.A. "Facilities planning - 4th edition", *Int. J. Prod. Res.*, **49**(24), pp. 7519-7520 (2011).
2. Behnia, B., Mahdavi, I., Shirazi, B., and Paydar, M.M. "A bi-level mathematical programming for cell formation problem considering workers' interest", *Int. J. Ind. Eng. Prod. Res.*, **28**(3), pp. 267-277 (2017).
3. Dale, B.G., Burbidge, J.L., and Cottam, M.J. "Planning the Introduction of Group Technology", *Int. J. Oper. Prod. Manag.*, **4**(1), pp. 34-47 (1984).
4. Wemmerlöv, U. and Hyer, N.L. "Research issues in cellular manufacturing", *Int. J. Prod. Res.*, **25**(3), pp. 413-431 (1987).
5. Bajestani, M.A., Rabbani, M., Rahimi-Vahed, A.R., and Baharian Khoshkhou, G. "A multi-objective scatter search for a dynamic cell formation problem", *Comput. Oper. Res.*, **36**(3), pp. 777-794 (2009).
6. Sudhakara Pandian, R. and Mahapatra, S.S. "Manufacturing cell formation with production data usingneural networks", *Comput. Ind. Eng.*, **56**(4), pp. 1340-1347 (2009).
7. Yin, Y. and Yasuda, K. "Similarity coefficient methods applied to the cell formation problem: a comparative investigation", *Comput. Ind. Eng.*, **48**(3), pp. 471-489 (2005).
8. Forghani, K. and Mohammadi, M. "A genetic algorithm for solving integrated cell formation and layout problem considering alternative routings and machine capacities", *Sci. Iran. Trans. E*, **21**(6), pp. 2326-2346 (2014).
9. Shafer, S.M. and Rogers, D.F. "A goal programming approach to the cell formation problem", *J. Oper. Manag.*, **10**(1), pp. 28-43 (1991).
10. Mahdavi, I., Aalaei, A., Paydar, M.M., and Solimanpur, M. "Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment", *Comput. Math. with Appl.*, **60**(4), pp. 1014-1025 (2010).
11. Mahdavi, I., Javadi, B., Fallah-Alipour, K., and Slomp, J. "Designing a new mathematical model for cellular manufacturing system based on cell utilization", *Appl. Math. Comput.*, **190**(1), pp. 662-670 (2007).

12. Liao, T.W. "Classification and coding approaches to part family formation under a fuzzy environment", *Fuzzy Sets Syst.*, **122**(3), pp. 425-441 (2001).

13. Renzi, C., Leali, F., Cavazzuti, M., and Andrisano, A.O. "A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems", *Int. J. Adv. Manuf. Technol.*, **72**(1-4), pp. 403-418 (2014).

14. Mahdavi, I., Aalaei, A., Paydar, M.M., and Solimanpur, M. "A new mathematical model for integrating all incidence matrices in multi-dimensional cellular manufacturing system", *J. Manuf. Syst.*, **31**(2), pp. 214-223 (2012).

15. Hosseini, A., Paydar, M.M., Mahdavi, I., and Jouzdani, J. "Cell forming and cell balancing of virtual cellular manufacturing systems with alternative processing routes using genetic algorithm", *J. Optim. Ind. Eng.*, **9**(20), pp. 41-51 (2016).

16. Offodile, O.F., Mehrez, A., and Grznar, J. "Cellular manufacturing: A taxonomic review framework", *J. Manuf. Syst.*, **13**(3), pp. 196-220 (1994).

17. Singh, N. "Design of cellular manufacturing systems: An invited review", *Eur. J. Oper. Res.*, **69**(3), pp. 284-291 (1993).

18. Joines, J.A., Culbreth, C.T., and King, R.E. "Manufacturing cell design: an integer programming model employing genetic algorithms", *IIE Trans.*, **28**(1), pp. 69-85 (1996).

19. Mohammadi, M. and Forghani, K. "Designing cellular manufacturing systems considering S-shaped layout", *Comput. Ind. Eng.*, **98**, pp. 221-236 (2016).

20. Bootaki, B., Mahdavi, I., and Paydar, M.M. "New criteria for configuration of cellular manufacturing considering product mix variation", *Comput. Ind. Eng.*, **98**, pp. 413-426 (2016).

21. Aalaei, A. and Davoudpour, H. "A robust optimization model for cellular manufacturing system into supply chain management", *Int. J. Prod. Econ.*, **183**, pp. 667-679 (2017).

22. Imran, M., Kang, C., Lee, Y.H., Jahanzaib, M., and Aziz, H. "Cell formation in a cellular manufacturing system using simulation integrated hybrid genetic algorithm", *Comput. Ind. Eng.*, **105**, pp. 123-135 (2017).

23. Delgoshaei, A. and Gomes, C. "A multi-layer perceptron for scheduling cellular manufacturing systems in the presence of unreliable machines and uncertain cost", *Appl. Soft Comput.*, **49**, pp. 27-55 (2016).

24. Aljuneidi, T. and Bulgak, A.A. "Designing a cellular manufacturing system featuring remanufacturing, recycling, and disposal options: A mathematical modeling approach", *CIRP J. Manuf. Sci. Technol.*, **19**, pp. 25-35 (2017).

25. Jawahar, N. and Subhaa, R. "An adjustable grouping genetic algorithm for the design of cellular manufacturing system integrating structural and operational parameters", *J. Manuf. Syst.*, **44**, pp. 115-142 (2017).

26. Kuo, Y. and Liu, C.-C. "Operator assignment in a labor-intensive manufacturing cell considering intercell manpower transfer", *Comput. Ind. Eng.*, **110**, pp. 83-91 (2017).

27. Rabbani, M., Habibnejad-Ledari, H., Rafiei, H., and Farshbaf-Geranmayeh, A. "A bi-objective mathematical model for dynamic cell formation problem considering learning effect, human issues, and worker assignment", *Sci. Iran.*, **23**(5), pp. 2341-2354 (2016).

28. Rabbani, M., Keyhanian, S., Manavizadeh, N., and Farrokhi-Asl, H. "Integrated dynamic cell formation-production planning: A new mathematical model", *Sci. Iran.*, **24**(5), pp. 2550-2566 (2017).

29. Mahootchi, M., Forghani, K., and Abdollahi Kamran, M. "A two-stage stochastic model for designing cellular manufacturing systems with simultaneous multiple processing routes and subcontracting", *Sci. Iran.*, **25**(5), pp. 2824-2837 (2018).

30. Saranwong, S. and Likasiri, C. "Product distribution via a bi-level programming approach: Algorithms and a case study in municipal waste system", *Expert Syst. Appl.*, **44**, pp. 78-91 (2016).

31. Calvete, H.I., Galé, C., and Iranzo, J.A. "Planning of a decentralized distribution network using bilevel optimization", *Omega*, **49**, pp. 30-41 (2014).

32. Ma, Y., Yan, F., Kang, K., and Wei, X. "A novel integrated production-distribution planning model with conflict and coordination in a supply chain network", *Knowledge-Based Syst.*, **105**, pp. 119-133 (2016).

33. Safaei, A.S., Farsad, S., and Paydar, M.M. "Robust bi-level optimization of relief logistics operations", *Appl. Math. Model.*, **56**, pp. 359-380 (2018).

34. Safaei, A.S., Farsad, S., and Paydar, M.M. "Emergency logistics planning under supply risk and demand uncertainty", *Oper. Res.*, pp. 1-24 (2018) (In Press).

35. Mokhlesian, M. and Zegordi, S.H. "Pricing and advertising decisions in a dominant-retailer supply chain: A multi-follower bi-level programming approach", *Sci. Iran.*, **25**(4), pp. 2254-2266 (2018).

36. Talbi, E.G., *Metaheuristics for Bi-level Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg (2013).

37. Leung, S.C.H. and Chan, S.S.W. "A goal programming model for aggregate production planning with resource utilization constraint", *Comput. Ind. Eng.*, **56**(3), pp. 1053-1064 (2009).

38. Gen, M., Altiparmak, F., and Lin, L. "A genetic algorithm for two-stage transportation problem using priority-based encoding", *OR Spectr.*, **28**(3), pp. 337-354 (2006).

39. Michalewicz, Z., Vignaux, G., and Hobbs, M. "A non-standard genetic algorithm for the non-linear transportation problem.", *ORSA J. Comput.*, **3**, pp. 307-316 (1991).

40. Prüfer, H. "Neuer Beweis eines Satzes uber Permutationen.", *Arch. Der Math. Und Phys.*, **27**, pp. 142-144 (1918).

41. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. "Optimization by simulated annealing", *Science, New Series*, **220**(4598), pp. 671-680 (1983).

42. Mafarja, M.M. and Mirjalili, S. "Hybrid whale optimization algorithm with simulated annealing for feature selection", *Neurocomputing*, **260**, pp. 302-312 (2017).

43. Assadi, M.T. and Bagheri, M. "Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems", *Comput. Ind. Eng.*, **96**, pp. 149-161 (2016).

44. Maghsoudlou, H., Kahag, M.R., Niaki, S.T.A., and Pourvaziri, H. "Bi-objective optimization of a three-echelon multi-server supply-chain problem in congested systems: Modeling and solution", *Comput. Ind. Eng.*, **99**, pp. 41-62 (2016).

45. Eberhart, R. and Kennedy, J. "A new optimizer using particle swarm theory", *MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39-43 (1995).

46. Taguchi, G. *Introduction to Quality Engineering: Designing Quality into Products and Processes*, Illustrated The Organization, White Plains (1986).

47. Kuo, R.J., Lee, Y.H., Zulvia, F.E., and Tien, F.C. "Solving bi-level linear programming problem through hybrid of immune genetic algorithm and particle swarm optimization algorithm", *Appl. Math. Comput.*, **266**(43), pp. 1013-1026 (2015).

48. Sarrafha, K., Rahmati, S.H.A., Niaki, S.T.A., and Zaretalab, A. "A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA", *Comput. Oper. Res.*, **54**, pp. 35-51 (2015).

49. Hwang, C.L. and Yoon, K., *Multiple Attributes Decision Making Methods & Applications*, Springer, Berlin (1981).

50. Hwang, C.-L., Lai, Y.-J., and Liu, T.-Y. "A new approach for multiple objective decision making", *Comput. Oper. Res.*, **20**(8), pp. 889-899 (1993).

Biographies

Bardia Behnia is a PhD candidate of Industrial Engineering at Babol Mazandaran University of Science and Technology and Instructor of University College of Rouzbahan. He received his MSc degree from Iran University of Science and Technology. The title of his thesis is "Modeling of bi-level cellular manufacturing system considering human resource interest".

Iraj Mahdavi is a Full Professor of Industrial Engineering at Mazandaran University of Science and Technology, Babol, Iran. He received his PhD from India in Production Engineering. He is also a member in the editorial board of five journals and a scientific committee member of international conferences. He was awarded as the best researcher in the engineering area in Iran and among the best professors of Iran. He has published over 350 research papers. His research interests include production planning, supply chain management, fuzzy networks, digital management, and intelligent operations management. He also edited and authored 2 books in electronic supply network design and multi-objective optimization, respectively.

Babak Shirazi is an Assistant Professor of Industrial Engineering at Mazandaran University of Science and Technology, Babol, Iran. He studied Computer Engineering in Tehran University from 1996 to 2000. He has graduated in MSc and PhD of industrial engineering from Mazandaran University of Science and Technology, Babol, Iran, in 2005 and 2010, respectively. His research interests include simulation modelling in manufacturing system and supply chain management, Enterprise Architecture, and industrial enterprise development strategy setting. He has published over 70 research papers. He authored a book in enterprise resources planning in Persian.

Mohammad Mahdi Paydar is an Assistant Professor of Industrial Engineering at Babol Noshirvani University of Technology. He received his PhD in Industrial Engineering from Iran University of Science and Technology. His research interests are cellular manufacturing systems, supply chain design, and modeling of manufacturing applications. He has published articles in journals such as Computers and Industrial Engineering, Computers and Operations Research, Expert Systems with Applications, Computers and Chemical Engineering, International Journal of Advanced Manufacturing Technology, Journal of Manufacturing Systems, International Journal of Production Research, and International Journal of Operational Research and 25 papers in international conferences.