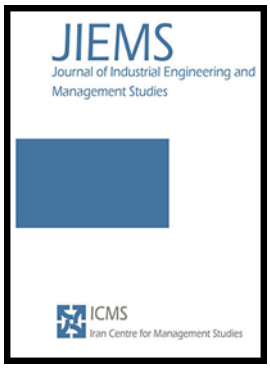
JIEMS
Journal of Industrial Engineering and Management Studies

Vol. 7, No. 1, 2020, pp. 191-219

DOI: 10.22116/JIEMS.2020.110249

www.jiems.icms.ac.ir



Simulated annealing and artificial immune system algorithms for cell formation with part family clustering

Sara Firouzian[1], Iraj Mahdavi[1,*], Mohammad Mahdi Paydar[2], Mahdi Saadat[1]

Abstract
This study proposed a new model for cellular manufacturing systems to group parts and machines in dedicated cells using a part-machine incidence matrix to minimize the voids. After identifying the exceptional elements, the machines required for processing the remained operations of corresponding parts which are not processed in the dedicated cells are specified. This results in a new matrix called part family-machine. Then, by clustering the part family-machine incidence matrix, the part families that should be assigned to a specific cell to achieve the highest similarity can be determined. The similarity can be translated to sharing machines required for completing the processes and form new cells called shared cells to minimize the number of exceptional elements and voids. Unlike previous models in which the similarity is considered only in the dedicated cells, here, the similarity would be observed in the entire production process. Due to the model NP-hardness, two meta-heuristics including artificial immune system (AIS) and simulated annealing (SA) are proposed. The efficiency of the algorithms is compared to that of exact solutions. Also, the algorithms are compared regarding the quality of solutions. Finally, according to grouping efficacy measure, SA has a superior performance in comparison with AIS by spending less CPU time.

Keywords: cell formation; part family clustering; cellular manufacturing system; dedicated cell; shared cell.

Received: November 2019-26

Revised: April 2020-15

Accepted: April 2020-16

1. Introduction

Cellular manufacturing (CM) is one of the well-known production methods used in most major manufacturing centers with high product diversity and multi-purpose facilities as a suitable production approach. In CM, every cell consists of several production machines and equipment and is capable to process a group of parts called family with similar production processes (Ayough et al., 2015; Golpîra and Tirkolaee, 2019). Main advantages of CM reported by previous conducted studies include, but are not limited to, reducing the setup and production

* Corresponding author; irajarash@rediffmail.com
[1] Mazandaran University of Science and Technology, Department of Industrial Engineering, Babol, Iran.
[2] Babol Noshirvani University of Technology, Department of Industrial Engineering, Babol, Iran.

time and material handling cost and increasing the quality level and inventory control (Heragu, 1994; Wemmerlöv and Hyer, 1989). One of the main problems in cellular manufacturing systems is cell formation problem (CFP) which is mainly the determination of parts and machine layout in the dedicated cells. The main objectives of the cell formation problem are to minimize the exceptional elements and voids. Binary part-machine incidence matrix is one of the commonly used tools to display how parts require the machines. The number of rows and the number of columns in the incidence matrix are equal to the number of parts and the number of machines, respectively. If the element related to a specific row and column is 1, then the part associated with that row requires the related machine in the corresponding column and if it is 0, then the machine associated with the corresponding machine is not required for processing the part associated with the corresponding row. In this study, part-machine incidence matrix (Mahdavi et al., 2012) and part family-machine incidence matrix are used.

In fact, different kinds of assignments can be established by displacing the columns and rows using block diagonal form in which the zeroes inside the block represent the voids and ones outside the blocks are exceptional elements that impose intercellular movements. In an ideal layout, there should be no voids or exceptional elements. Practically, it is desirable to minimize the number of voids and exceptional elements. This is because the voids inside the cell mean the absence of a link between the machine and the part and the exceptional elements translate to intercellular movements. In other words, voids are undesirable because all machines in a cell should be dedicated to the assigned parts and the number of exceptional elements should be minimized because the production operations of a part should be done inside its cell as far as possible. Figure 1 presents different configurations for a problem. Figure 1(A) has eleven exceptional elements without voids, Figure 1(B) has nine exceptional elements and four voids and Figure 1(C) has fifteen exceptional elements and four voids. Therefore, layout (A) is preferred to other configurations in terms of machines and intercellular parts.

		Machine					
		1	2	3	4	5	6
Part	1	1	1	1	0	0	1
	6	1	0	1	1	0	1
	2	0	1	1	1	1	0
	4	0	0	1	1	1	0
	5	1	0	0	1	1	1
	3	1	0	1	0	1	1

(A)

		Machine					
		1	2	3	4	5	6
Part	1	1	1	1	0	0	1
	2	0	1	1	1	1	0
	3	1	0	1	0	1	1
	4	0	0	1	1	1	0
	5	1	0	0	1	1	1
	6	1	0	1	1	0	1

(B)

		Machine					
		1	2	3	4	5	6
Part	1	1	1	1	0	0	1
	2	0	1	1	1	1	0
	3	1	0	1	0	1	1
	4	0	0	1	1	1	0
	5	1	0	0	1	1	1
	6	1	0	1	1	0	1

(C)

Figure 1. Different kinds of diagonal form of part-machine assignment

An initial layout of machines within the cells is needed in cellular manufacturing systems due to the costs of machine reconfiguration inside the cells and impossibility of displacement. One of the ways to prevent machine duplicate in cellular manufacturing systems is the movement of parts to do operation outside their dedicated cells. By emerging the movement concept in cellular manufacturing systems, parts for which operation is not done in their dedicated cells are moved to other cells that process completely different parts. In the classic cell formation in the literature review, the similarity of part operations is used to cluster the dedicated cells. Our technique in this study is to realize the similarity by the end of the operation parts as far as possible. For this reason, dedicated cells are formed by a part-machine incidence matrix to minimize the number of voids. After identifying the exceptional elements, a new matrix is formed based on the part families' operation requirements that call for their movements among machines. The matrix is called part family-machine incidence matrix. Part family-machine incidence matrix is a binary matrix in which 1 means a part family needs a bottle-neck machine (a machine to process exceptional elements). The part family-machine incidence matrix is

introduced for the first time in this study. In addition, through clustering of part family-machine matrix, an accurate planning is performed to deal with the parts whose operations are not done in their dedicated cells. This can lead to the determination of the part families which are assigned to a cell and therefore, the highest similarity in term of need for shared machines to complete their process and form some cells called shared cells is achieved. For example, the final configuration of dedicated cells related is shown in Figure 2(A). Also, part family-machine incidence matrix of this example is according to Figure 2(B). By clustering this matrix, one shared cell is formed and is shown in Figure 2(C). The objective function of the proposed model for this example is 3; in other words, this example has zero voids related to dedicated cells and 2 voids and an exceptional element for third part family related to the shared cells.

		Machine					
		1	6	2	3	4	5
Part	1	1	1	1	1	0	0
	3	1	1	0	1	0	1
	5	1	1	0	0	1	1
	6	1	1	0	1	1	0
	2	0	0	1	1	1	1
	4	0	0	0	1	1	1

(A)

		Part Family		
		PF_1	PF_2	PF_3
Machine	2	1	0	0
	3	1	0	1
	4	1	1	0
	5	1	1	0

(B)

		Part Family	
		PF_1	PF_2
Machine	2	1	0
	3	1	0
	4	1	1
	5	1	1

(C)

Figure 2. Illustration of dedicated cells, part family-machine incidence matrix, and formed shared cell

CFP is an NP-hard problem (Ballakur and Steudel, 1987). Several methods have been introduced to solve this problem. The methods include both heuristics and meta-heuristics. In this study, we formulated a mathematical model to form the part family-machine incidence matrix and cluster it to establish shared cells. In addition, artificial immune system and simulated annealing algorithms are utilized to minimize the number of voids in the dedicated cells and number of voids and exceptional elements in the shared cells.

Organization of other sections is as follows. Section 2 represents the literature review. Section 3 explains the problem formulation. Section 4 is to provide validation of the model. The proposed algorithms are presented in sections 5 and 6. In section 7, the parameters of algorithms are adjusted. Numerical experiments and analyses are presented in section 8 and finally, Section 9 concludes the paper.

2. Literature Review

Many studies have been conducted in the field of cell formation in classic cellular manufacturing. The approaches followed by these researches include but are not confined to the some categories such as method of coding and classification, production flow analysis, the similarity coefficient, methods based on fuzzy set and graph theory, array method, mathematical programming, and heuristic and meta-heuristic methods.

In the last decades of research, many approaches have mainly adopted part-machine incidence matrix as the input data for the cell formation problem. Classification and summarizing of them can be seen in some of studies (Papaioannou and Wilson, 2010; Yin and Yasuda, 2006). In this study, some recent researches that have addressed the CFP as a binary part-machine incidence matrix are reviewed. Chen and Cheng (1995) used a neural network algorithm to determine the formation of cells. They used adaptive resonance theory and neural network in cellular manufacturing systems. Cheng et al. (1998) formulated a cell formation problem as Traveling Salesman Problem (TSP) and proposed a solution methodology based on genetic algorithms. Onwubolu and Mutingi (2001) proposed a genetic algorithm to determine cell formation in cellular manufacturing systems. Gonçalves and Resende (2004) presented an approach to determine family of productions and machine cells. Their method was a combination of

heuristic local research and genetic algorithm. Albadawi et al. (2005) presented a mathematical model to form manufacturing cells. Their approach includes two steps: in the first step, machine cell is determined using a similarity coefficient matrix factor and in the second step, an integer mathematical model is used to assign parts to cells. In order to solve the CFP, James et al. (2007) proposed a hybrid genetic grouping algorithm which combines a local research algorithm with a standard genetic algorithm. Computational results were established with grouping efficiency standard by a set of data related to cell formation problem gathered through literature review. Hybrid approach of genetic algorithm had a better objective function value compared to that of standard genetic algorithm.

Mahdavi et al. (2009) proposed a mathematical model to solve CFP based on cell application concept in cellular manufacturing systems. The purpose of their model was to minimize the number of exceptional elements and intercellular voids. Also, they proposed an efficient genetic algorithm to solve the mathematical model. Anvari et al. (2010) developed a particle swarm optimization algorithm to determine the part families using similarity coefficient and to form machine groups to minimize the number of voids and exceptional elements. Pailla et al. (2010) applied two algorithms to solve CFP. Firstly, an evolutionary algorithm was suggested that improves the efficiency of the standard genetic algorithm by considering cooperation with a local search around some of the solutions it visits. Secondly, a simulated annealing algorithm was implemented that uses the same representation scheme of a feasible solution. Paydar et al. (2011) formulated a model for simultaneous determination of part families and machine groups in CFP to minimize the exceptional elements and the number of voids in cells. One of the advantages of their model is that the number of manufacturing cells are considered as a decision variable whose value is determined by the model. Elbenani et al. (2012) proposed a local research method to solve cell formation problem assuming that at least a part and a machine should be located in any cell. Their proposed method used resonance strategy to improve the local solution and utilized destruction strategy to create a new solution according to previously obtained solution. By solving 35 problems brought from the literature and comparing the solutions of the proposed method with the best-obtained solution, they demonstrated the efficiency of their method in terms of grouping efficacy. Banerjee et al. (2012) proposed predator-prey genetic algorithm to solve grouping technology and cellular manufacturing problems. Their algorithm is focused on local search strategy to maintain balance between hunting and hunter avoiding premature convergence. Paydar and Saidi-Mehrabad (2013) proposed a hybrid genetic-neighborhood research algorithm to solve the CFP to minimize the exceptional elements and voids. They compared their computational results with grouping efficacy of 35 various methods proposed in the literature and justified the performance superiority of their hybrid method.

Brown (2015) in a study developed mathematical models to obtain a strategic long-term layout for the cells in a CMS to minimize the exceptional elements to reduce costs in terms of intercellular movement, machine duplication and subcontracting while taking machine capacities into account to avoid capacity violations require a specific machine. Kia et al. (2015) proposed a mixed-integer mathematical model to incorporate some new features such as: (i) manufacturing cells with variable numbers and shapes, (ii) machine depot for keeping idle machines, (iii) machines of unequal areas and (iv) manufacturing cells with regular rectangular shapes set on the continuous shop floor and in the design and configuration of the dynamic CMS. Erenay et al. (2015) in a study compared different methods of layer designing of a cellular manufacturing system. In their study, three kinds of cells, i.e., dedicate, shared and remainder cells are introduced and it is explained that each part family has a dedicated cell and if needed has a shared cell and a remainder cell to perform the required operations which are not done in the corresponding dedicated cell. Two part families with similar operations form a shared cell and three part families and most of form a remainder cell. Martins et al. (2015) proposed a new

approach for solving the cell formation problem using the group efficacy objective function. Their method was based on the iterated local search meta–heuristic coupled with a variant of the variable neighborhood descent method. Bootaki et al. (2015) presented a dynamic CFP with a robust design approach and objective function of minimizing the total intercellular movement and maximizing the machine and worker utilization. Buruk Sahin and Alpay (2016) proposed a genetic algorithm for the problem of part-machine-worker cell formation. The performance of their proposed GA on all of the test problems was satisfying. Mehdizadeh and Rahimi (2016) presented an integrated mathematical model to solve the dynamic cell formation problem considering operator assignment and inter/intra cell layouts problems with machine duplication. Due to NP-hardness of the problem, they proposed two meta-heuristics as solution methods: multi-objective simulated annealing (MOSA) and multi-objective vibration damping optimization (MOVDO).

Finally, Table 1 is provided to compare this work with other mentioned researches. Using this table, the research gaps can be identified. These researches are compared in terms of several model assumptions, type of solution method, and data sets for parameters.

Table 1. A brief review of related works to find research gaps

Reference	Model assumptions									Solution method		Data	
	Part	Machine	Cell	ShC	UBC	ExE	Voids	CFP	PFC	MtH	Exact	Case	Example
(Onwubolu and Mutingi, 2001)	×	×	×					×		×			×
(Gonçalves and Resende, 2004)	×	×	×					×	×	×			×
(Albadawi et al., 2005)	×	×	×					×			×		×
(James et al., 2007)	×	×	×					×	×	×		×	
(Duran et al., 2008)	×	×	×					×		×			×
(Wu et al., 2008)	×	×	×					×		×			×
(Mahdavi et al., 2009)	×	×	×			×	×	×		×		×	
(Anvari et al., 2010)	×	×	×			×	×	×		×			×
(Pailla et al., 2010)	×	×	×					×		×		×	
(Noktehdan et al., 2010)	×	×	×					×		×			×
(Paydar et al., 2011)	×	×	×			×	×	×	×		×	×	
(Elbenani et al., 2012)	×	×	×		×			×			×		×
(Mahdavi et al., 2012)	×	×	×			×	×	×			×		×
(Banerjee and Das, 2012)	×	×	×					×		×			×
(Paydar and S. Mehrabad, 2013)	×	×	×			×	×	×		×			×
(Jouzdani et al., 2014)	×	×	×		×			×		×		×	
(Brown, 2015)	×	×	×			×		×			×		×
(Erenay et al., 2015)	×	×	×	×				×	×		×		×
(Martins et al., 2015)	×	×	×					×		×		×	
(Bootaki et al., 2015)	×	×	×		×			×			×		×
(Niakan et al., 2016)	×	×	×		×			×		×			×
(Mehdizadeh et al., 2016)	×	×	×		×			×		×			×
(Alimoradi et al., 2016)	×	×	×					×			×		×
(Buruk Sahin and Alpay, 2016)	×	×	×					×		×			×
(Mehdizadeh and Rahimi, 2016)	×	×	×		×			×		×			×
(Goli et al., 2019)	×	×							×		×		×
(Farughi et al., 2019)	×	×	×					×			×	×	
(Bouaziz et al., 2020)	×	×	×		×			×		×	×		×
This research	×	×	×	×	×	×	×	×	×	×	×		×

UBC: Upper Bound for Components ExE: Exceptional Elements CFP: Cell Formation Problem
PFC: Part Family Clustering ShC: Shared Cell MtH: Meta-Heuristic

This table shows that shared cell is applied only in 3.5% of these 28 selected papers. Also, upper bound for components is considered only in 25% of these research. Besides, the exceptional elements and voids are used in 21% and 18% of these works, respectively. Also, part family clustering is noted only in 18% of these papers. On the other hand, considering these five mentioned assumptions have not been conducted in a single study simultaneously. Moreover, other information can be seen in this table such as type of solution method, and data sets for parameters.

In this study, after grouping parts and machines and forming dedicated cells to minimize number of voids, part family-machine incidence matrix is formed through identifying the exceptional elements. By clustering this new matrix, some cells, called shared cells, are formed. In fact, the required operations that are not done in the dedicated cells and have similar process to other part families are done in their shared cell. Therefore, a new cell formation is established to minimize the number of voids and exceptional elements. Here, shared cells concept of Erenay et al. (2015) is adopted to cluster part family-machine incidence matrix. Unlike their research in which machines are purchased based on the number of dedicated, shared and remained cells, in this study, bottle-neck machines are duplicated only once again. The technique in this study is to realize the similarity through the operations in whole part processes as far as possible. It is noteworthy than, if bottle-neck machines are purchased initially and the shared cells are not established and dedicated cells are formed, then, the similarity is reduced and efficient cells are not formed. Figure 1 illustrates an example that clarifies the concept. In this figure, if machines 2, 3, 4 and 5 located in part family-machine incidence matrix related to Figure 2(B) are purchased initially, the number of voids is increased to 18. In this study, first, a mathematical model is developed to form dedicated cells and clustering of part family-machine incidence matrix to establish shared cells. Then, the proposed algorithms are analyzed using numerical experiments.

3. Problem formulation

In this section, the problem is formulated as a mixed non-linear mathematical model to solve CFP by clustering part family-machine incidence matrix. The proposed model deals with the minimization of voids in dedicated cells and voids and exceptional elements in shared cells. The assumptions are:

- The main cells are already known.
- Machine duplication is allowed.
- The number of shared cells is considered as a variable and is determined by the model.
- Each part has only one dedicated cell.
- Upper and lower limits of the number of machines in dedicated and shared cells are known.
- At least two parts families are in each shared cell.
- Machine requirements of parts are expressed as a binary part-machine incidence matrix.

The main focus of research is to form dedicated cells and shared cells are to maximize the similarity in the part process.

Sets:

I Number of part types
J Number of machine types
K Number of dedicated cells
C Number of shared cells

Indices:

i Index for part type (i=1, 2, …, I)
j Index for machine type (j=1, 2, …, J)
k, k' Index for dedicated cell (k, k'=1, 2, …, K)

c Index for shared cell (c=1, 2, …, C)

Input parameters:

r_{ij} 1 if part type i needs machine type j; =0 otherwise
L_k Minimum number of machine types for cell k
U_k Maximum number of machine types for cell k
Lf_k Minimum number of part types for cell k
Lc_c Minimum number of machine types for cell c
A The sufficiently large positive number

Decision variables:

Z_{ik} 1 if part i is assigned to dedicated cell k; =0 otherwise
y_{jk} 1 if machine j is assigned to dedicated cell k; =0 otherwise
v_{jc} 1 if machine j is assigned to shared cell c; =0 otherwise
x_{kc} 1 if part family k is assigned to shared cell c; =0 otherwise
M_c 1 if shared cell c is opened; =0 otherwise
PF_{kj} 1 if part family k needs machine j; =0 otherwise
Q_k 1 if part family k is selected for part family-machine incidence matrix; =0 otherwise
$f_m(i, k)$ The number of operations needed to be performed on part i in dedicated cell k
$f_{max}(i)$ The Maximum needed operations to be performed on part i in a dedicated cell

Objective function:

Min Z=

$$\sum_{k=1}^{K}\left(\sum_{i=1}^{I}\sum_{j=1}^{J} z_{ik} y_{jk} - \sum_{i=1}^{I}\sum_{j=1}^{J} z_{ik} y_{jk} r_{ij}\right) + \sum_{k=1}^{K}\sum_{j=1}^{J} PF_{kj} - \left(\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j=1}^{J} x_{kc} v_{jc} PF_{kj}\right) + \sum_{c=1}^{C}\left(\sum_{k=1}^{K}\sum_{j=1}^{J} x_{kc} v_{jc} - \sum_{k=1}^{K}\sum_{j=1}^{J} x_{kc} v_{jc} PF_{kj}\right) \quad (1)$$

Constraints:

$$\sum_{k=1}^{K} z_{ik} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{k=1}^{K} y_{jk} = 1 \quad \forall j \in J \quad (3)$$

$$\sum_{j=1}^{J} y_{jk} \geq L_k \quad \forall k \in K \quad (4)$$

$$\sum_{j=1}^{J} y_{jk} \leq U_k \quad \forall k \in K \quad (5)$$

$$\sum_{i=1}^{I} z_{ik} \geq Lf_k \quad \forall k \in K \quad (6)$$

$$f_m(i,k) = \sum_{j=1}^{J} y_{jk} \cdot r_{ij} \quad \forall i \in I, k \in K \quad (7)$$

$$f_{\max}(i) = \max\{f_m(i,k)\} \quad \forall i \in I \quad (8)$$

$$z_{ik} \leq \frac{f_m(i,k)}{f_{\max}(i)} \quad \forall i,k \quad (9)$$

$$\sum_{\substack{k'=1 \\ k' \neq k}}^{K} \sum_{i=1}^{I} z_{ik} y_{jk'} r_{ij} \geq PF_{kj} \quad \forall k \in K, j \in J \quad (10)$$

$$\sum_{\substack{k'=1 \\ k'\neq k}}^{K}\sum_{i=1}^{I} z_{ik} y_{jk'} r_{ij} \leq A \times PF_{kj} \quad \forall k \in K, j \in J \tag{11}$$

$$Q_k \leq \sum_{j=1}^{J} PF_{kj} \quad \forall k \in K \tag{12}$$

$$A \times Q_k \geq \sum_{j=1}^{J} PF_{kj} \quad \forall k \in K \tag{13}$$

$$\sum_{c=1}^{C} x_{kc} \leq Q_k \quad \forall k \in K \tag{14}$$

$$M_c \leq \sum_{j=1}^{J}\sum_{k=1}^{K} x_{kc} \cdot v_{jc} \quad \forall c \in C \tag{15}$$

$$A \times M_c \geq \sum_{j=1}^{J}\sum_{k=1}^{K} x_{kc} \cdot v_{jc} \quad \forall c \in C \tag{16}$$

$$\sum_{j=1}^{J} v_{jc} \leq Lc_c \quad \forall c \in C \tag{17}$$

$$v_{jc} \leq \sum_{k=1}^{K} PF_{kj} \cdot x_{kc} \quad \forall j \in J, c \in C \tag{18}$$

$$A \times v_{jc} \geq \sum_{k=1}^{K} PF_{kj} \cdot x_{kc} \quad \forall j \in J, c \in C \tag{19}$$

$$\sum_{c=1}^{C} v_{jc} \leq 1 \quad \forall j \in J \tag{20}$$

$$\sum_{k=1}^{K} x_{kc} \geq 2 \times M_c \quad \forall c \in C \tag{21}$$

$$z_{ik}, y_{jk}, v_{jc}, x_{kc}, M_c, PF_{kj}, Q_k \in \quad \forall i \in I, j \in J, k \in K, c \in C \tag{22}$$

The first term of the objective function is to minimize the number of voids in the dedicated cells and the three next terms are to minimize the number of exceptional elements and voids in shared cells. Equation (2) guarantees that each part is assigned to only one dedicated cell. Equation (3) ensures that each machine is assigned to only one dedicated cell. Constraints (4) and (5) impose upper and lower limits for the number of machines in dedicated cells. Equation (6) shows the lower limit of dedicated cells in term of number of parts. Constraints (7), (8) and (9) determine the dedicated cell of each part; i.e., the cell in which the largest number of part operations are done. Constraints (10) and (11) form part family-machine matrix for which the part families that continue their process outside their dedicated cell and the corresponding required machines should be determined. In this step, all exceptional elements that cause the movements are identified. Constraints (12) and (13) determine the part family to enter the part family-machine matrix. For example, if part family 1 needs to be processed in a cell other than its dedicated cell, we have Q_1=1. Equation (14) expresses the fact that each part family is assigned to only a single shared cell in the part family-machine matrix. Constraints (15) and (16) are related to opening shared cells. Equation (17) specifies the upper limit for the number of machines in shared cells. Constraints (18) and (19) determine the machine to enter the part family-machine matrix. Equation (20) expresses the fact that each machine is assigned to a single shared cell in the part

family-machine matrix. Equation (21) states that to shared cells, only two part families can be assigned and finally equation (22) express that decision variables of the problem are binary.

3.1. Linearization

The proposed model is a non-linear integer programming model due to non-linear terms that include multiplication of binary variables in all terms of objective function and constraints (8), (9), (10), (11), (15), (16), (18) and (19). For linearization, five auxiliary binary variables are defined as $F_{ijk} = Z_{ik} \times y_{jk}$, $E_{kjc} = x_{kc} \times v_{jc} \times PF_{kj}$, $B_{kjc} = x_{kc} \times v_{jc}$, $G_{kjc} = x_{kc} \times PF_{kj}$, and $D_{ijk\acute{k}} = Z_{ik} \times y_{j\acute{k}}$. Regarding these variables, below constraints should be added to mathematical model:

$$F_{ijk} - z_{ik} - y_{jk} + 1.5 \geq 0 \quad \forall i, j, k \tag{23}$$

$$1.5 \times F_{ijk} - z_{ik} - y_{jk} \leq 0 \quad \forall i, j, k \tag{24}$$

$$E_{kjc} - x_{kc} - v_{jc} - PF_{kj} + 2.5 \geq 0 \quad \forall k, j, c \tag{25}$$

$$2.5 \times E_{kjc} - x_{kc} - v_{jc} - PF_{kj} \leq 0 \quad \forall k, j, c \tag{26}$$

$$B_{kjc} - x_{kc} - v_{jc} + 1.5 \geq 0 \quad \forall k, j, c \tag{27}$$

$$1.5 \times B_{kjc} - x_{kc} - v_{jc} \leq 0 \quad \forall k, j, c \tag{28}$$

$$G_{kjc} - PF_{kj} - x_{kc} + 1.5 \geq 0 \quad \forall k, j, c \tag{29}$$

$$1.5 \times G_{kjc} - PF_{kj} - x_{kc} \leq 0 \quad \forall k, j, c \tag{30}$$

$$D_{ijkk'} - z_{ik} - y_{jk'} + 1.5 \geq 0 \quad \forall i, j, k, k' \tag{31}$$

$$1.5 \times D_{ijkk'} - z_{ik} - y_{jk'} \leq 0 \quad \forall i, j, k, k' \tag{32}$$

$$f_m(i,k) - f_m(i,k') < M \times W_{ikk'} \quad \forall k, k', i, k' \neq k \tag{33}$$

$$f_m(i,k) - f_m(i,k') \geq (W_{ikk'} - 1) \times M \quad \forall k, k', i, k' \neq k \tag{34}$$

$$\sum_{k' \neq k} W_{ikk'} \geq (K-1) \times Z(i,k) \quad \forall \text{i,k} \tag{35}$$

3.2. Linearized Model

Linearized model is provided as in what follows.

$$\min\ z = \sum_{k=1}^{K}\left(\sum_{i=1}^{I}\sum_{j=1}^{J} F_{ijk} - \sum_{i=1}^{I}\sum_{j=1}^{J} F_{ijk} \times r_{ij}\right) + \sum_{k=1}^{K}\sum_{j=1}^{J} PF_{kj} - \left(\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{j=1}^{J} E_{kjc}\right) + \sum_{c=1}^{C}\left(\sum_{k=1}^{k}\sum_{j=1}^{J} B_{kjc} - \sum_{k=1}^{K}\sum_{j=1}^{J} E_{kjc}\right) \tag{36}$$

Subject to:

(2)– (7) و (12)– (14) و (17) و (20)– (22) و (23)– (35)

$$\sum_{\substack{k'=1 \\ k' \neq k}}^{K}\sum_{i=1}^{I} D_{ijkk'} \cdot r_{ij} \geq PF_{kj} \quad \forall k \in K, j \in J \tag{37}$$

$$\sum_{\substack{k'=1 \\ k' \neq k}}^{K}\sum_{i=1}^{I} D_{ijkk'} \cdot r_{ij} \leq A \times PF_{kj} \quad \forall k \in K, j \in J \tag{38}$$

$$M_c \leq \sum_{j=1}^{J}\sum_{k=1}^{K} B_{kjc} \quad \forall c \in C \tag{39}$$

$$A \times M_c \geq \sum_{j=1}^{J} \sum_{k=1}^{K} B_{kjc} \quad \forall c \in C \tag{40}$$

$$v_{jc} \leq \sum_{k=1}^{K} G_{kjc} \quad \forall j \in J, c \in C \tag{41}$$

$$A \times v_{jc} \geq \sum_{k=1}^{K} G_{kjc} \quad \forall j \in J, c \in C \tag{42}$$

$$\begin{aligned} &z_{ik}, y_{jk}, v_{jc}, x_{kc}, M_c, PF_{kj}, Q_k, F_{ijk}, E_{kjc}, B_{kjc}, G_{kjc}, \\ &D_{ijkk'}, W_{ikk'} \in \quad \forall i \in I, j \in J, k, k' \in K, c \in C \end{aligned} \tag{43}$$

4. Solution approaches

Here, two solution approaches include SA and AIS was described.

4.1. The proposed simulated annealing Algorithm

The idea that formed original SA was introduced by Metropolis et al. (1953). Thirty years later, Kirkpatrick et al. (1983) developed the idea proposed by Metropolis as a technique to search feasible solutions in optimization problems. Unlike the gradient methods and other accurate searching methods, simulated annealing algorithm is designed to avoid being trapped in local minima. In fact, simulated annealing algorithm is a searching method that uses Markov chain to converge transfer possibility function suitably. In other word, simulated annealing is a random searching technique to solve optimization methods inspired by solid crystal production process and the cooling of the molten material. Slower decrease of the temperature of the molten material results in formation of larger and arranged solid crystals. By the increasing of the cooling rate for liquid, crystals form slower and more organized. As a result, the produced solid has less porosity and more shock resistance.

So, annealing process includes the control of temperature and strain rate. The application of this algorithm can be seen in some of studies to solve cell formation problems (Paydar et al., 2010; Xambre and Vilarinho, 2003). The procedure of simulated annealing algorithm implemented for the problem is shown in Figure 3.

1- Defining objectives and adjusting the parameters of the algorithm.
Defining primary temperature T_0 and primary value x_0
Defining final temperature T_f
Defining value of temperature drop α
Initially, $n=1$, $T=T_0$, $x=x_0$, $x=x^*$
2- When $n=1$, $T=T_f$
Random displacing to new location (some changing in current solution for random searching)
$X_{n+1}=x_n+rand$
Calculating $\Delta f=f_{n+1}(x_{n+1})-f_n(x_n)$
3- If new solution is better, so it is accepted
Otherwise
A random number r is produced
If $p=exp[-\Delta f/T]>r$, new solution is accepted
End
Updating x^* and f^*
$n=n+1$
End

Figure 3. The pseudo-code of the proposed simulated annealing algorithm

Parameters of SA include:

x_0= The initial solution
x^*= Best known solution
$f(x)$= Value of objective function at point x
x_n= Solution in iteration n
n= Counter of algorithm iteration
T_0= Initial temperature
T_f= Final temperature
p= Probability of acceptance of xn when it is not better than xn-1 (transfer possible)
α=Rate of temperature decrease

Basic concepts of the proposed simulated annealing algorithm are explained as below:

a) Coding the Solution

Coding the solution in the problem is in the way that each solution should be a combination of machine layouts and parts in the dedicated cells and is shown in Table 2; so that, number of rows in each matrix is the same as number of dedicated cells and each matrix has two columns. The first column represents machines and the second column represents the parts available in each dedicated cell. For example, if six parts, six machines and three cells are taken, a structural matrix is as follows. Table 2 represents a solution to the problem. The first row represents that machines number 4 and 5 and parts numbers 1 and 3 are located in the first dedicated cell. Also, machines numbers 1 and 6 and parts numbers 2 and 4 are located in the second cell and machines numbers 2 and 3 and parts numbers 5 and 6 are located in the third cell.

Table 2. The numerical example of solution representation

	Machine	Part
Cell	[4,5]	[1,3]
	[1,6]	[2,4]
	[2,3]	[5,6]

b) Neighborhood-Move

Neighborhood-move mechanism is of high importance in SA method. In this problem, mutation function selects two dedicated cells per each solution taken to mutation function. For each pair of selected cells, a machine is selected randomly from one cell and replaced by other selected machine from the other cell. Therefore, layout of machines in cells is completely changed and as a result and the layout of dedicated cells should be changed. According to new layout of machines, dedicated cells are relocated. The output of this function, there is one new solution and one new layout of parts and machines. In the conducted mutation in Table 3, the first and second cells are selected randomly and machine number 4 from the first cell is replaced by machine number 6 of the second cell (swap operator). Similarly, for parts, this swap operator is done and all swapped numbers are bold.

Table 3. The numerical example of the neighborhood- move of the proposed SA algorithm

	Machine	Part
Cell	[**4**,5]	[1,**3**]
	[1,**6**]	[**2**,4]
	[2,3]	[5,6]

→

	Machine	Part
Cell	[**6**,5]	[1,**2**]
	[1,**4**]	[**3**,4]
	[2,3]	[5,6]

c) Solution Evaluation Function in SA

Functions of performance evaluation of a solution are used to compare the solutions to select the best one. Therefore, the objective function of the problem is used as a function to evaluate the performance of solutions; i.e., the smaller the objective function for a solution, the better the solution. In this problem, after determining the initial solution, the number of voids in dedicated cell and the number of voids and exceptional elements in shared cells are achieved by the objective function.

d) Acceptance of a Solution

In local optimization algorithms, the new solution is accepted only when the solution is improved according to the objective function while in SA, not only better solutions are accepted and the solutions are improved regarding the objective function value, but also worse solutions are also likely to be accepted. There is a criterion for acceptance probability of these solutions (P), which is dependent on temperature, in simulation annealing algorithm is given in equation (44).

$$P(\text{Accept}) = \exp^{-\frac{\Delta f}{T}} \tag{44}$$

Δf Represents changes of objective function. Also, $T \in R^{+}$ represents degree of temperature for each iteration.

e) Stopping Criterion of SA Algorithm

The proposed algorithm stops when the temperature reaches its final value, i.e., the final temperature.

4.2. The proposed artificial immune system algorithm

In this section, an artificial immune system (AIS) algorithm is proposed to solve the problem. This algorithm is inspired by the function of immune system of human body. AIS examines a set of solutions simultaneously; therefore, it can search different points of solution space simultaneously and this increases the ability of the algorithm to achieve a global optimal solution and also enables the algorithm to avoid local optimal.

The procedure of artificial immune system is presented as what follows:

1) Form the initial population of antibodies (solutions) randomly.
2) Calculate the affinity between each pair of antibodies and their objective function values.
3) Sort antibodies based on affinity function and select a certain number of best ones in terms of affinity to run the cloning phase.
4) Conduct the cloning phase for the antibodies selected from the previous stage based on their affinity.
5) Apply the mutation operator on the cloned antibodies based on their affinity.
6) Evaluate the mutant results by affinity function.
7) Replace a certain number of best solutions produced the worst results of the current society.
8) Check the stopping criterion. If the stopping condition is not met, go to step 2.

Basic concepts of artificial immune system algorithm are as follows:

a) Coding the Solution

How to display the solution in artificial immune algorithm is similar to that of simulated annealing algorithm. Therefore, it is not explained in details in this section and the reader is referred to sub-section (4.1).

b) Initial Population

Initial population is the same for the two algorithms with searching method based on population. The initial populations with a certain number of solutions are produced randomly.

c) Affinity Function

Affinity function in artificial immune algorithm is a criterion to evaluate the quality of solutions in the algorithm. The larger the value of the affinity function, the better the solutions. Affinity function in this study is defined as the inverse of the objective function expressed as *(1/objective function)*.

d) Cloning phase

In the AIS algorithm, n_c antibodies with the highest affinity values are selected for cloning step. The cloned number of each n_c selected antibodies is calculated by (n_c-k+1), where k denotes the antibody with the kth highest affinity function value in the antibody population (Lin and Ying, 2013). Therefore, the antibodies with higher affinity values have a higher number of clones.

e) Mutation phase

Mutation is performed to produce neighborhood-move in artificial immune system algorithm. Mutation is a relatively small change in a part of the solution. Mutation operator in artificial immune system algorithm is applied to all the cloned solutions in cloning step. The mutation used for the problem is similar to that performed in neighborhood-move in simulated annealing algorithm. Therefore, in this section, it is not explained in details (please read sub-section (4.1)).

f) Stopping Criterion

Stopping criterion in the proposed AIS is specified as a certain iteration number determined by trial and error and examining convergence diagrams. The pseudo-code of AIS algorithm is shown in Figure 4.

Artificial immune system algorithm:

1. Input: define r_{ij} matrix, number of Dedicated cell and number of shared cell, upper bound and lower bound of part in each dedicated cell
2. Select the first combination of parts and machines as initial solution (s_0).
3. **For** all part and machine combinations
4. Generate the $popsize$ of the s_0 permutation randomly as initial population, determine the $iteration$ value according to size of s_0;
5. Calculate the affinity function of these antibodies;
6. **For** *max gen*
7. Select the best n_c of the antibodies;
8. Make (n_c-k+1) copies of each antibodies (clone);
9. Mutate all of the cloned antibodies;
10. Calculate the affinity function value for all of the mutated antibodies;
11. The n_c of the worse antibodies in current population are replaced by the n_c of the best mutated antibodies;
12. **End for**
13. Store the obtained best affinity function value;
14. **End for**

Figure 4. The pseudo-code of AIS algorithm

5. Model validation

In this section, three valid numerical examples are provided to demonstrate the performance of the proposed model coded and solved by GAMS software on a desktop computer equipped with 6 GB of RAM and an Intel® Core™ i7 @2.20 GHz CPU.

Example 1: In this example, 6 parts and 6 machines, for which the part-machine incidence matrix is shown in Table 4 (Chen and Cheng, 1995), are considered. This example has three dedicated cells. After solving the proposed model, final cell formation in Table 5 and also a shared cell is formed. All the exceptional elements belong to part families 2 and 3. The operations of the parts in these part families are done in the formed shared cell as shown in Table 6. Value of objective function is 3. If machines 1, 3 and 6 are purchased initially and dedicated cells are formed without machine duplication, the value of the objective function is 6 and no shared cell is formed.

Table 4. Part-machine incidence matrix related to example 1

		Machine					
		1	2	3	4	5	6
Part	1	0	0	0	1	1	0
	2	1	0	1	0	0	1
	3	0	0	0	1	1	0
	4	1	0	0	0	0	1
	5	0	1	1	0	0	1
	6	1	1	1	0	0	0
L_k		2					
U_k		10					
Lf_k		2					
Lc_c		4					

Table 5. Final configuration of dedicated cell related to example 1

		Machine					
		4	5	1	6	2	3
Part	1	1	1	0	0	0	0
	3	1	1	0	0	0	0
	2	0	0	1	1	0	1
	4	0	0	1	1	0	0
	5	0	0	0	1	1	1
	6	0	0	1	0	1	1

Table 6. Part family- machine incidence matrix related to example 1

		Part Family	
		PF_2	PF_3
Machine	1	0	1
	3	1	0
	6	0	1

Example 2: A problem with 10 parts and 10 machines is presented in this example. The part-machine incidence matrix is shown in Table 7 (Chen and Cheng, 1995). By solving the proposed model with three dedicated cells, final dedicated cell formation related is obtained as shown in Table 8. Also, for this example, a shared cell is formed and all exceptional elements are related to part families 2 and 3 that are processed in shared cells as depicted in Table 9. In addition, machines 9 and 10 are purchased. The objective function value for this example is 6. If machines 9 and 10 with similar situations are purchased initially and dedicated cells are formed, the value

of the objective function is 20 and this shows the benefits of the proposed model over the ones previously proposed.

Table 7. Part-machine incidence matrix related to example 2

		Machine									
		1	2	3	4	5	6	7	8	9	10
Part	1	1	1	1	0	0	0	0	0	1	1
	2	1	0	0	0	0	0	0	0	1	1
	3	0	0	0	1	1	0	0	0	1	1
	4	0	0	0	0	0	1	1	1	1	0
	5	0	0	0	0	0	0	1	1	0	0
	6	1	0	1	0	0	0	0	0	1	1
	7	0	0	0	1	0	0	0	0	1	0
	8	0	0	0	1	1	0	0	0	0	1
	9	0	0	0	0	0	1	1	1	1	0
	10	0	0	0	0	0	0	1	1	0	0
L_k		2									
U_k		6									
Lf_k		2									
Lc_c		4									

Table 8. Final configuration of dedicated cell related to example 2

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

Table 9. Part family- machine incidence matrix related to example 2

		Part Family	
		PF_2	PF_3
Machine	9	1	1
	10	0	1

Example 3: This example deals with 20 parts and 8 machines. The part-machine incidence matrix is shown in Table 10 (Chandrasekharan and Rajagopalan, 1986a). Also, for this example, a shared cell is formed for which the exceptional elements are related to part families 1, 2 and 3 whose operations are performed in shared cells as shown in Table 11. By solving the proposed

model with three dedicated cells, final cell formation of dedicated cells is obtained as shown in Table12. Also, machines 1, 2, 4, 5, 6 and 8 are purchased. The objective function value for this example is 11. If the same kinds of machines with similar situation are purchased at first and dedicated cells are formed, the value of the objective function is increased to 80. This demonstrates the advantage of the proposed model.

Table 10. Part-machine incidence matrix related to example 3

		Machine							
		1	2	3	4	5	6	7	8
Part	1	1	0	0	0	1	1	0	1
	2	0	1	0	0	1	1	0	1
	3	1	1	0	1	1	0	0	1
	4	1	1	0	1	0	0	0	1
	5	0	0	1	0	0	1	1	1
	6	0	1	1	0	1	0	1	0
	7	0	1	1	1	0	1	1	0
	8	0	0	1	1	0	1	1	1
	9	1	1	0	1	0	0	0	1
	10	1	0	0	1	1	1	0	1
	11	0	1	1	0	0	1	1	0
	12	0	0	1	0	1	1	1	0
	13	0	0	1	1	1	0	1	1
	14	1	0	0	1	0	1	0	1
	15	1	0	0	1	1	1	0	0
	16	1	0	1	0	1	1	1	1
	17	0	0	1	1	1	0	1	1
	18	1	1	0	1	0	1	0	0
	19	1	0	1	1	1	1	1	0
	20	0	1	1	1	1	0	1	0
L_k		1							
U_k		10							
Lf_k		1							
Lc_c		10							

Table 11. Part family- machine incidence matrix related to example 3

		Part Family		
		PF_2	PF_3	PF_1
Machine	1	0	1	1
	2	1	1	0
	4	0	1	0
	5	1	1	0
	6	1	1	0
	8	0	0	1

The results of model validation for four problems brought from the literature (Chandrasekharan and Rajagopalan, 1986a; Chen and Cheng, 1995; Stanfel, 1985) are shown in Table 13. For the first three problems, by using GAMS software, the optimal solutions are obtained. A solution could not be found for the fourth problem after running the solver for more than twenty-four hours. This is due to the complexity of the problem and limitations of exact methods utilized by GMAS software; therefore, a meta-heuristic algorithm should be developed for solving the

large-sized problems. For this purpose, two algorithms of simulated annealing and artificial immune system are presented and described in the following sections.

Table 12. Final configuration of dedicated cell related to example 3

		Machine							
		2	5	6	1	4	8	3	7
Part	1	0	1	1	1	0	1	0	0
	2	1	1	1	0	0	1	0	0
	3	1	1	0	1	1	1	0	0
	4	1	0	0	1	1	1	0	0
	9	1	0	0	1	1	1	0	0
	10	0	1	1	1	1	1	0	0
	14	0	0	1	1	1	1	0	0
	15	0	1	1	1	1	0	0	0
	18	1	0	1	1	1	0	0	0
	5	0	0	1	0	0	1	1	1
	6	1	1	0	0	0	0	1	1
	7	1	0	1	0	1	0	1	1
	8	0	0	1	0	1	1	1	1
	11	1	0	1	0	0	0	1	1
	12	0	1	1	0	0	0	1	1
	13	0	1	0	0	1	1	1	1
	16	0	1	1	1	0	1	1	1
	17	0	1	0	0	1	1	1	1
	19	0	1	1	1	1	0	1	1
	20	1	1	0	0	1	0	1	1

Table 13. The results of the linearized model solved by GAMS

Article/ attributes	No. of component types							CPU time (s)	Solution status
	Part	Machine	Dedicated cells	Shared cells	Variable	Constraint	Objective		
(Chen and Cheng, 1995)	6	6	3	1	738	1425	3	24.656	Global opt.
(Chen and Cheng, 1995)	10	10	3	1	3906	7643	7	62.246	Global opt.
(Chandrasekharan and Rajagopalan, 1986a)	20	8	3	1	2466	4803	11	1120.791	Global opt.
(Stanfel, 1985)	24	14	7	N/A	21314	42147	N/A	N/A	N/A

6. Adjusting parameter of algorithms

Parameters may influence on the algorithm performance. Considering the effect of parameters, in recent years the algorithm configuration is attracted wide attention. There are many ways to design experiments but the most frequently used approach is Taguchi method.

6.1. Adjusting parameter of SA algorithm

SA method is known to be more parameter-sensitive and the parameter values may affect the efficiency and efficacy of solutions. These parameters include:

Initial temperature: to find a proper initial temperature (T_0), the available data of problem is used. Specifically, if *max(Δf)* shows the highest value that the objective function may have (the objective of problem is minimization), equation (45) represents a proper primary temperature

$$T_0 \approx -\frac{\max(\Delta f)}{\ln p_0} \tag{45}$$

In the above equation, P_0 is a pre-determined probability and is usually set to 0.95 according to previous researches (Kia et al., 2012). Here, the primary temperature is obtained for each problem using parameter adjusting by Taguchi method and equation (45) is used as an approximate value of (T_0).

Final temperature: in theory, the final temperature should be zero so that no worse solution is accepted. Otherwise, if $T_f \rightarrow 0$ some of evaluations are useless. In practice, a very small number is selected as the final temperature which varies between 10^{-5} and 10^{-10} ($T_f=10^{-10}\sim10^{-5}$) and is selected depending on quality of needed solution and time limit. In this research, the final temperature is also optimized by using Taguchi method.

Cooling Coefficient: Other important matter that should be taken into account is how to control the gradual reduction of temperature of cooling process. There are several methods to control cooling or temperature drop. In this study, geometric cooling schedule decreases the temperature using a cooling coefficient of $0<\alpha<1$; for this purpose, equation (46) is used to calculate temperature in each iteration.

$$T(t) = T_0 \alpha^t \quad , \quad t = 1, 2, ..., t_f \tag{46}$$

The advantage of this equation is that, when T moves toward zero, the temperature approaches zero. In practice, usually α=0.7~0.9 is used to as an approximation to determine the accurate value by Taguchi method.

There are various methods to adjust the parameters in an algorithm. In this study, one of the common methods, known as Taguchi multi-factorial experiment, is utilized (Taguchi, 1986). MINITAB 14 software is used in order to analyze. The manner of Taguchi method include:

1. Determining the effective parameters of the algorithm.
2. Allocating the amount of tasting levels for every parameter.
3. Allocating one of the Taguchi's orthogonal arrays tables to the parameters.
4. Analyzing of data by following equation: *S/N ratio*=-10×Log_{10}(*objective function*)^2

The process to adjust the parameters for simulated annealing algorithm is presented by three examples brought from the literature review.

1. The small-sized problem, brought from reference (Cheng et al., 1998), with 6 parts, 6 machines, 3 dedicated cells and 3 candidate shared cells.
2. The medium-sized problem, brought from reference (Chen et al., 1995), with 20 parts, 8 machines, 3 dedicated cells and 3 candidate shared cells.
3. The large-sized problem, brought from reference (Chen et al., 1995), with 40 parts, 24 machines, 7 dedicated cells and 3 candidate shared cells.

In average *S/N ratio* table, for each parameter in different levels, rank is the smallest *S/N ratio*. Given various experiments, the problem with small dimensions shows no significant sensitivity to the parameters of SA in this study. To adjust the parameter in problem with medium size using Taguchi method, three levels, according to Table 14, are considered for cooling coefficient α, final temperature T_f and initial temperature T_0. Table of orthogonal arrays of Taguchi method for all factors and three levels ($L9$) are shown in Table 15.

Table 14. SA algorithm factors and their levels for medium size

Parameters	A	B	C
Name	α	T_f	T_0
Levels	0.85	10^{-8}	500
	0.985	10^{-9}	526
	0.99	10^{-10}	550

Table 15. The orthogonal array L9

Experiment #	Parameters		
	A	B	C
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2

The problem is solved twenty-five times for each experiment in Table 15. Then, the ratios of *S/N* are obtained as shown in Table 16 by calculating the average of solutions and average of computational times and result analysis. According to *S/N* ratio diagram related to each parameter in Figure 5, the best value for parameters α, T_f and T_0 are 0.99, 10^{-9} and 550, respectively.

Table 16. Average *S/N* ratio table of SA algorithm for medium size

Level	A	B	C
1	-19.00	-18.63	-12.93
2	-21.29	-26.06	-24.56
3	-25.49	-20.40	-25.19
Delta	6.49	7.43	12.26
Rank	3	2	3

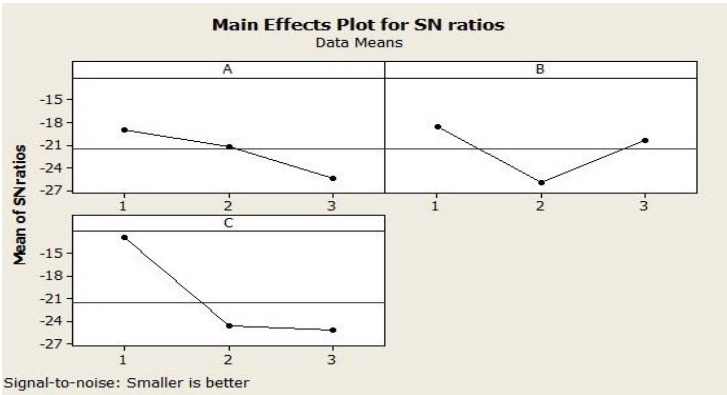


Figure 5. The mean *S/N* ratio plot for each level of SA factors for medium size

To adjust the parameter in the large-sized problem using Taguchi method, three levels according to Table 17 are considered for cooling coefficient α, final temperature T_f and initial temperature T_0. Table of orthogonal arrays of Taguchi method for all factors and three levels ($L9$) are shown in Table 15.

Table 17. SA algorithm factors and their levels for large size

Parameters	A	B	C
Name	A	T_f	T_0
	0.90	10^{-8}	2690

Levels	0.985	10^{-9}	3000
	0.99	10^{-10}	3099

The problem is solved twenty-five times for each experiment in Table 15. Then, the ratios of *S/N* are obtained as shown in Table 18 by calculating the average of responses and average of computational times and result analysis. According to *S/N* ratio diagram related to each parameter in Figure 6, the best value for parameters α, and T_0 are 0.90, 10^{-8} and 3000, respectively.

Table 18. Average S/N ratio table of SA algorithm for large size

Level	A	B	C
1	-34.09	-28.08	-23.88
2	-21.78	-21.84	-25.88
3	-19.05	-26.61	-25.86
Delta	15.03	6.25	2.00
Rank	1	1	2

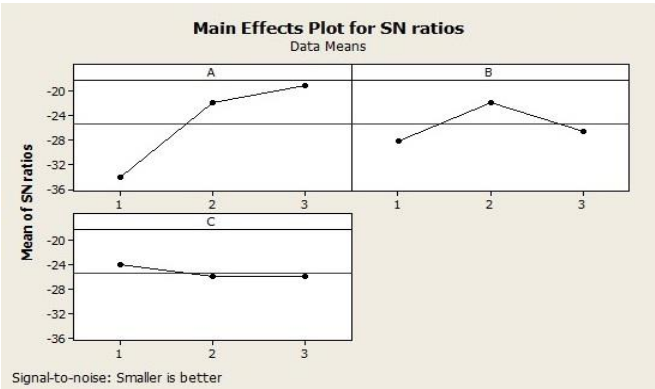


Figure 6. The mean *S/N* ratio plot for each level of SA factors for large size

6.2. Adjusting Parameter of AIS Algorithms

Artificial immune system algorithm has three controlling parameters including number of iterations (*maxgen*), initial population (*Popsize*) and number of mutant population (n_c). Three levels according to Table 19 are considered for each one of controlling parameters. The parameter levels are defined through trial and error. Orthogonal arrays table of Taguchi method for three factors and three levels ($L9$) are shown in Table 15.

Table 19. AIS algorithm factors and their levels

Parameters	A	B	C
Name	*Maxgen*	*Popsize*	n_c
Levels	100	50	*Popsize*/2
	120	60	*Popsize*/3
	150	70	*Popsize*/4

The problem is solved twenty-five times for each experiment in Table 15. Then, the ratios of *S/N* are obtained as shown in Table 20 by calculating the average of responses and average of computational times and result analysis. According to *S/N* ratio diagram related to each parameter in Figure 7, the best value for parameters *maxgen*, *Popsize* and n_c are 100, 50 and *Popsize*/3, respectively.

Table 20. Average *S/N* ratio table of AIS algorithm

Level	A	B	C

1	-21.41	-23.42	-19.40
2	-18.40	-15.39	-21.41
3	-18.40	-15.39	-18.40
Delta	3.01	8.03	3.01
Rank	1	1	2

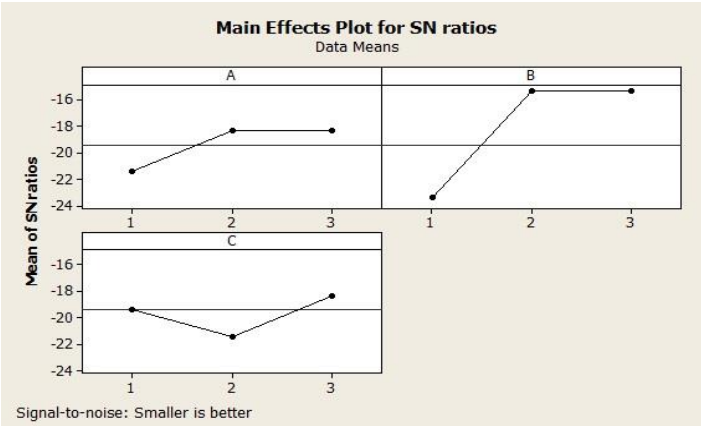


Figure 7. The mean *S/N* ratio plot for each level of AIS factors

Furthermore, since similar behavior was observed by AIS in medium and large dimensions, to avoid increasing the size of the paper, their reporting has been ignored. So, the best value for parameters *maxgen*, *Popsize* and n_c are 100, 50 and *Popsize*/3, respectively per all three dimensions.

7. Performance Analysis of the Algorithms

The performance of the algorithms with the optimized levels of parameters can be compared to that of exact solutions methods. In this study, 21 benchmark problems brought from the literature (Boctor, 1991; Carrie, 1973; Chan and Milner, 1982) are used to compare the performance of artificial immune system algorithm and simulated annealing algorithm. To cover different sizes, problems with small size (e.g. 5×7), medium size (e.g. 10×15, 14×24) and large size (e.g. 24×40) are selected. The number of candidate shared cells is 3 for all examples. The data related to examples are in Table 21.

There are several evaluation measures to examine the quality of solutions to the problems mentioned in the literature (Boulif and Atif, 2006; Chandrasekharan and Rajagopalan, 1986b). Two known measure include grouping efficiency and grouping efficacy. Kumar and Chandrasekharan (1990) proposed a grouping efficacy measure which can be calculated as:

$$\mu = \frac{e - e_0}{e + e_v} \tag{47}$$

In equation (47), e is the total number of 1s in the given part-machine incidence matrix, e_v is the number of voids and e_0 is the number of exceptional elements. Grouping efficacy values related to Figure 1-A, Figure 1-B and Figure 1-C are 0.52, 0.51 and 0.29, respectively.

The examples in Table 21 are solved using GAMS software, SA and AIS algorithms and the computational results are summarized in Table 22 that shows the grouping efficacy of the solutions obtained by the proposed algorithms and the exact method. Given the obtained results, SA algorithm solves the small and medium-size problems efficiently and optimal solutions are resulted for small size problems. According to grouping efficacy measure in SA and AIS

algorithms in Table 22, SA, as a single-solution-based search algorithm has a superior performance in comparison with AIS as a population-based algorithm. This table shows the total number of opened shared cells per 21 problems are equal to 32 and 35 for SA and AIS, respectively. This concept implies that AIS has a superior performance in comparison with SA.

Table 21. The values of different parameters in the tested problems

NO.	Problem Source	No. of		
		Machines	Parts	Cells
1	Fig. 4a in (Waghodekar and Sahu, 1984)	5	7	2
2	(Chen and Cheng, 1995)	6	6	3
3	(Seifoddini, 1989)	5	18	2
4	(Kusiak and Cho, 1992)	6	8	2
5	(Kusiak and Heragu, 1987)	7	11	5
6	(Boctor, 1991)	7	11	4
7	(Chandrasekharan and Rajagopalan, 1986a)	8	20	3
8	(Chandrasekharan and Rajagopalan, 1986b)	8	20	2
9	(Mosier and Taube, 1985)	10	10	5
10	(Chen and Cheng, 1995)	10	10	3
11	(Chan and Milner, 1982)	10	15	3
12	(Stanfel, 1985)	14	24	7
13	(King, 1980)	16	43	7
14	(Mosier and Taube, 1985)	20	20	5
15	(Carrie, 1973)	20	35	5
16	Data set 1 in (Chandrasekharan and Rajagopalan, 1989)	24	40	7
17	Data set 2 in (Chandrasekharan and Rajagopalan, 1989)	24	40	7
18	Data set 3 in (Chandrasekharan and Rajagopalan, 1989)	24	40	7
19	Data set 5 in (Chandrasekharan and Rajagopalan, 1989)	24	40	10
20	Data set 6 in (Chandrasekharan and Rajagopalan, 1989)	24	40	12
21	Data set 7 in (Chandrasekharan and Rajagopalan, 1989)	24	40	12

To better demonstrate this superiority, three figures are drawn. As is shown in Table 22, although the GAMS has the highest CPU time, it has reached global optimum solutions and it was only able to answer the first eleven problems. This matter shows that this problem is subset of NP-hard, especially in large dimensions examples. Moreover, the comparison between two algorithms includes SA and AIS in terms of CPU time is illustrated in Figure 8. This figure shows that SA has less CPU time than AIS. Besides, Figure 9 illustrates the comparison between SA and AIS in terms of max grouping efficacy of dedicated cell. The results show that SA has a superior performance in comparison with AIS.

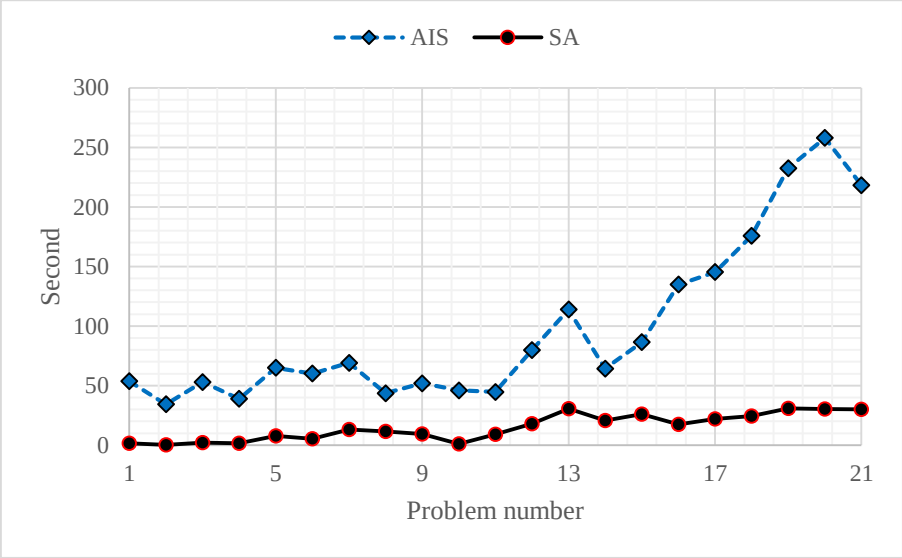


Figure 8. Comparison between two algorithms in terms of CPU time

Table 22. Performance of the proposed algorithms compared to GAMS

No.	Size	Number of opened shared cell			CPU (sec)			Max. Grouping efficacy of dedicated cell			Max. Grouping efficacy of shared cell		
		GAMS	SA	AIS	GAMS	SA	AIS	GAMS	SA	AIS	GAMS	SA	AIS
1	5×7	1	1	1	2.809	1.63	53.65	73.68	73.68	73.68	100	100	100
2	6×6	1	1	1	4.656	0.31	34.23	80	80	80	50	50	50
3	5×18	1	1	1	2.376	2.21	52.98	79.59	79.59	79.59	100	100	100
4	6×8	2	1	1	1.369	1.63	38.91	68	76.92	76.92	100	100	100
5	7×11	3	1	1	1000.4	7.79	64.94	56	56.52	56	100	58.33	53.33
6	7×11	2	1	1	178.74	5.46	60.13	70.83	66.66	65.38	100	50	50
7	8×20	1	1	1	1120.7	13.16	69.16	71.31	71.31	66.9	55.55	55.55	55.55
8	8×20	1	1	1	70.214	11.62	43.60	77.04	77.04	69.82	45.83	45.83	50
9	10×10	3	1	2	1000.4	9.45	51.89	52.77	52.23	42.50	72.72	75	71.42
10	10×10	1	1	1	62.246	0.902	45.943	69.23	69.23	69.23	75	75	75
11	10×15	0	0	1	75.08	9	44.67	92	92	74.54	-	-	100
12	14×24	N/A	1	3	N/A	17.86	79.82	N/A	69.8	65.26	N/A	70.58	55.55
13	16×43	N/A	2	3	N/A	30.49	113.84	N/A	59.62	48.88	N/A	54.28	46.42
14	20×20	N/A	3	1	N/A	20.66	64.3	N/A	47.51	35.40	N/A	100	67.74
15	20×35	N/A	2	1	N/A	26.14	86.40	N/A	73.91	58.49	N/A	100	40
16	24×40	N/A	0	0	N/A	17.40	134.83	N/A	100	100	N/A	-	-
17	24×40	N/A	2	3	N/A	21.87	145.32	N/A	89.69	60.19	N/A	47.72	30.43
18	24×40	N/A	3	3	N/A	24.29	175.81	N/A	78.51	51.77	N/A	63.63	60.71
19	24×40	N/A	3	3	N/A	30.90	232.51	N/A	57.88	43.61	N/A	53.52	52.94
20	24×40	N/A	3	3	N/A	30.21	258.01	N/A	48.95	40.86	N/A	90	47.91
21	24×40	N/A	3	3	N/A	30.19	218.35	N/A	45.87	38.79	N/A	58.49	52.50

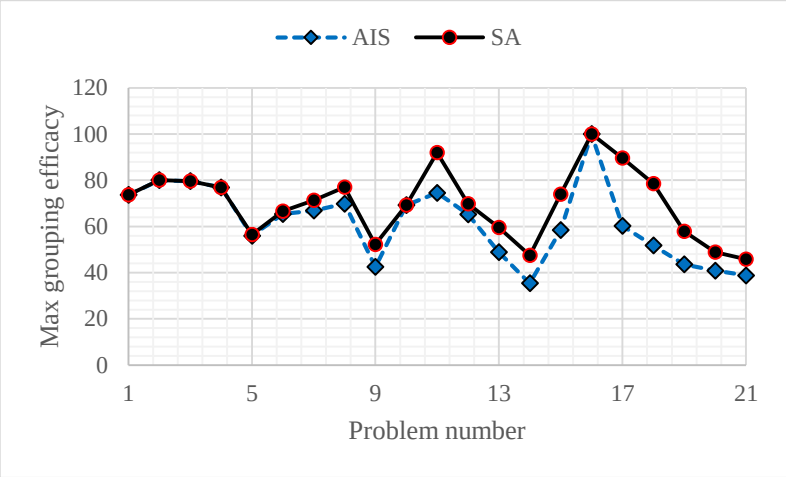


Figure 9. Comparison between two algorithms in terms of max grouping efficacy of dedicated cell

Moreover, Figure 10 demonstrates the comparison between SA and AIS in terms of max grouping efficacy of shared cell. The results show that SA has a superior performance in comparison with AIS.

Finally, based on obtained results it is found that the GAMS cannot be able to solve this proposed model in large dimensions examples, and it also takes a lot of time for small and medium-size examples. It shows that this problem is subset of NP-hard, especially in large dimensions examples. On the other hand, the results prove that the SA algorithm has a superior performance in comparison with AIS in term of CPU time, max grouping efficacy of shared cell, and max grouping efficacy of dedicated cell. These results can be useful for some industries that use cellular manufacturing to produce parts.

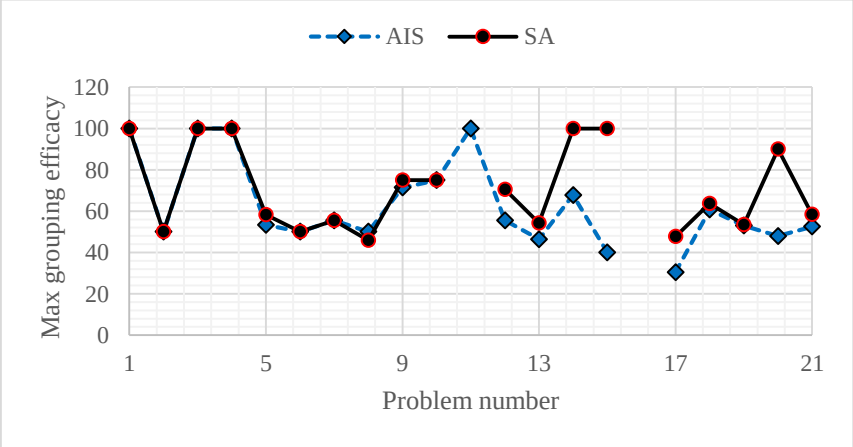


Figure 10. Comparison between two algorithms in terms of max grouping efficacy of shared cell

8. Conclusion

In this study, a new concept called part family-machine matrix and its clustering was introduced to form shared cells to remove exceptional elements and increase the similarity. A model was introduced by clustering part family-machine matrix to reach the cell formation problem which is the considering of similarity concept in the whole manufacturing process both dedicated cells and shared cells. Shared cells are established when the required budget is provided. Otherwise, they assumed virtually. In this model, dedicated cells were formed to minimize the number of voids and increase the similarity. After identifying the exceptional elements, shared cells were formed to minimize number of exceptional elements and voids. Two exact and meta-heuristic

methods were used to solve the model. GAMS Software was utilized for finding exact solutions. Due to the computational complexity of the problem, two meta-heuristic approaches of artificial immune system and simulated annealing were proposed to solve the problem. These approaches utilized according to their satisfying results in solving optimization problems. In this study, SA algorithm had superior performance in comparison with the AIS algorithm in terms of CPU time, max grouping efficacy of shared cell, and max grouping efficacy of dedicated cell. These results can be useful for some industries that use cellular manufacturing to produce parts. Furthermore, considering dedicated cell, shared cell, and part family clustering caused the model be close to real-world and it means that these assumptions make the model flexible and the managers of the organizations can be able to have a proper framework. Given the conducted studies and the literature review, some suggestions for future researches are as follows:

- Considering multiple optimization criteria that may be significant in production systems.
- The algorithms used in this study are relatively new meta-heuristic approaches that have shown a high potential for solving complex problems. Therefore, the use of a combination of approaches could be considered.
- Considering the uncertainty in the demand for manufacturing parts and proposing a model with fuzzy and/or stochastic parameters.
- Proposing a two-stage model instead of a model that considers the objective functions simultaneously.
- Incorporating the scheduling concepts into the cell formation and machine layout problems.
- Considering the dynamics of the problem.

References

Albadawi, Z., Bashir, H.A., and Chen, M., (2005). "A mathematical approach for the formation of manufacturing cells", *Computers & Industrial Engineering*, Vol. 48, No. 1, pp. 3–21.

Alimoradi, S., Hematian, M., and Moslehi, G., (2016). "Robust scheduling of parallel machines considering total flow time", *Computers & Industrial Engineering*, Vol. 93, pp. 152–161.

Anvari, M., Mehrabad, M.S., and Barzinpour, F., (2010). "Machine–part cell formation using a hybrid particle swarm optimization", *The International Journal of Advanced Manufacturing Technology*, Vol. 47, No. 5–8, pp. 745–754.

Ayough, A., Tabriz, A.A., and Javani, A., (2015). "Virtual manufacturing cells scheduling considering lotstreaming and sequence dependent setup times", *Journal of Industrial Engineering and Management Studies*, Vol. 2, No. 1, pp. 61–73.

Ballakur, A., and Steudel, H.J., (1987). "A within-cell utilization based heuristic for designing cellular manufacturing systems", *International Journal of Production Research*, Vol. 25, No. 5, pp. 639–665.

Banerjee, I., and Das, P., (2012). "Group technology based adaptive cell formation using predator–prey genetic algorithm", *Applied Soft Computing*, Vol. 12, No. 1, pp. 559–572.

Boctor, F.F., (1991). "A Jinear formulation of the machine-part cell formation problem", *International Journal of Production Research*, Vol. 29, No. 2, pp. 343–356.

Bootaki, B., Mahdavi, I., and Paydar, M.M., (2015). "New bi-objective robust design-based utilisation towards dynamic cell formation problem with fuzzy random demands", *International Journal of Computer Integrated Manufacturing*, Vol. 28, No. 6, pp. 577–592.

Bouaziz, H., Berghida, M., and Lemouari, A., (2020). "Solving the generalized cubic cell formation problem using discrete flower pollination algorithm", *Expert Systems with Applications*, Vol. 150, p. 113345.

Boulif, M., and Atif, K., (2006). "A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem", *Computers & Operations Research*, Vol. 33, No. 8, pp. 2219–2245.

Brown, J.R., (2015). "A capacity constrained mathematical programming model for cellular manufacturing with exceptional elements", *Journal of Manufacturing Systems*, Vol. 37, pp. 227–232.

Buruk Sahin, Y., and Alpay, S., (2016). "A metaheuristic approach for a cubic cell formation problem", *Expert Systems with Applications*, Vol. 65, pp. 40–51.

Carrie, A.S., (1973). "Numerical taxonomy applied to group technology and plant layout", *International Journal of Production Research*, Vol. 11, No. 4, pp. 399–416.

Chan, H.M., and Milner, D.A., (1982). "Direct clustering algorithm for group formation in cellular manufacture", *Journal of Manufacturing Systems*, Vol. 1, No. 1, pp. 65–75.

Chandrasekharan, M.P., and Rajagopalan, R., (1986a). "MODROC: an extension of rank order clustering for group technology", *International Journal of Production Research*, Vol. 24, No. 5, pp. 1221–1233.

Chandrasekharan, M.P., and Rajagopalan, R., (1986b). "An ideal seed non-hierarchical clustering algorithm for cellular manufacturing", *International Journal of Production Research*, Vol. 24, No. 2, pp. 451–463.

Chandrasekharan, M.P., and Rajagopalan, R., (1989). "GROUPABILITY: an analysis of the properties of binary data matrices for group technology", *International Journal of Production Research*, Vol. 27, No. 6, pp. 1035–1052.

Chen, C.-L., Cotruvo, N.A., and Baek, W., (1995). "A simulated annealing solution to the cell formation problem", *International Journal of Production Research*, Vol. 33, No. 9, pp. 2601–2614.

Chen, S.-J., and Cheng, C.-S., (1995). "A neural network-based cell formation algorithm in cellular manufacturing", *International Journal of Production Research*, Vol. 33, No. 2, pp. 293–318.

Cheng, C.H., Gupta, Y.P., Lee, W.H., and Wong, K.F., (1998). "A TSP-based heuristic for forming machine groups and part families", *International Journal of Production Research*, Vol. 36, No. 5, pp. 1325–1337.

Duran, O., Rodriguez, N., and Consalter, L.A., (2008). "A PSO-Based Clustering Algorithm for Manufacturing Cell Design", *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)*, IEEE, pp. 72–75.

Elbenani, B., Ferland, J.A., and Bellemare, J., (2012). "Genetic algorithm and large neighbourhood search to solve the cell formation problem", *Expert Systems with Applications*, Vol. 39, No. 3, pp. 2408–2414.

Erenay, B., Suer, G.A., Huang, J., and Maddisetty, S., (2015). "Comparison of layered cellular manufacturing system design approaches", *Computers & Industrial Engineering*, Vol. 85, pp. 346–358.

Farughi, H., Mostafayi, S., and Afrasiabi, A., (2019). "Bi-objective robust optimization model for configuring cellular manufacturing system with variable machine reliability and parts demand: A real case study", *Journal of Industrial Engineering and Management Studies*, Vol. 6, No. 2, pp. 120–146.

Goli, A., Babaee Tirkolaee, E., and Soltani, M., (2019). "A robust just-in-time flow shop scheduling problem with outsourcing option on subcontractors", *Production & Manufacturing Research*, Vol. 7, No. 1, pp. 294–315.

Golpîra, H., and Tirkolaee, E.B., (2019). "Stable maintenance tasks scheduling: A bi-objective robust optimization model", *Computers & Industrial Engineering*, Vol. 137, p. 106007.

Gonçalves, J.F., and Resende, M.G.C., (2004). "An evolutionary algorithm for manufacturing cell formation", *Computers & Industrial Engineering*, Vol. 47, No. 2–3, pp. 247–273.

Heragu, S.S., (1994). "Group technology and cellular manufacturing", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 2, pp. 203–215.

James, T.L., Brown, E.C., and Keeling, K.B., (2007). "A hybrid grouping genetic algorithm for the cell formation problem", *Computers & Operations Research*, Vol. 34, No. 7, pp. 2059–2079.

Jouzdani, J., Barzinpour, F., Shafia, M.A., and Fathian, M., (2014). "Applying simulated annealing to a generalized cell formation problem considering alternative routings and machine reliability", *Asia-Pacific Journal of Operational Research*, Vol. 31, No. 04, pp. 1–26.

Kia, R., Baboli, A., Javadian, N., Tavakkoli-Moghaddam, R., Kazemi, M., and Khorrami, J., (2012). "Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing", *Computers & Operations Research*, Vol. 39, No. 11, pp. 2642–2658.

Kia, R., Shirazi, H., Javadian, N., and Tavakkoli-Moghaddam, R., (2015). "Designing group layout of unequal-area facilities in a dynamic cellular manufacturing system with variability in number and shape of cells", *International Journal of Production Research*, Vol. 53, No. 11, pp. 3390–3418.

King, J.R., (1980). "Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm", *International Journal of Production Research*, Vol. 18, No. 2, pp. 213–232.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., (1983). "Optimization by Simulated Annealing", *Science*, Vol. 220, No. 4598, pp. 671–680.

Kusiak, A., and Cho, M., (1992). "Similarity coefficient algorithms for solving the group technology problem", *International Journal of Production Research*, Vol. 30, No. 11, pp. 2633–2646.

Kusiak, A., and Heragu, S.S., (1987). "Group technology", *Computers in Industry*, Vol. 9, No. 2, pp. 83–91.

Lin, S.-W., and Ying, K.-C., (2013). "Minimizing makespan in a blocking flowshop using a revised artificial immune system algorithm", *Omega*, Vol. 41, No. 2, pp. 383–389.

Mahdavi, I., Aalaei, A., Paydar, M.M., and Solimanpur, M., (2012). "A new mathematical model for integrating all incidence matrices in multi-dimensional cellular manufacturing system", *Journal of Manufacturing Systems*, Vol. 31, No. 2, pp. 214–223.

Mahdavi, I.., Paydar, M.M., Solimanpur, M., and Heidarzade, A., (2009). "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing", *Expert Systems with Applications*, Vol. 36, No. 3, pp. 6598–6604.

Martins, I.C.,, Pinheiro, R.G.S., Protti, F., and Ochi, L.S., (2015). "A hybrid iterated local search and variable neighborhood descent heuristic applied to the cell formation problem", *Expert Systems with Applications*, Vol. 42, No. 22, pp. 8947–8955.

Mehdizadeh, E., Daei Niaki, S.V., and Rahimi, V., (2016). "A vibration damping optimization algorithm for solving a new multi-objective dynamic cell formation problem with workers training", *Computers & Industrial Engineering*, Vol. 101, pp. 35–52.

Mehdizadeh, E., and Rahimi, V., (2016). "An integrated mathematical model for solving dynamic cell formation problem considering operator assignment and inter/intra cell layouts", *Applied Soft Computing*, Vol. 42, pp. 325–341.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., (1953). "Equation of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics*, Vol. 21, No. 6, pp. 1087–1092.

Mosier, C., and Taube, L., (1985). "Weighted similarity measure heuristics for the group technology machine clustering problem", *Omega*, Vol. 13, No. 6, pp. 577–579.

Niakan, F., Baboli, A., Moyaux, T., and Botta-Genoulaz, V., (2016). "A bi-objective model in sustainable dynamic cell formation problem with skill-based worker assignment", *Journal of Manufacturing Systems*, Vol. 38, pp. 46–62.

Noktehdan, A., Karimi, B., and Husseinzadeh Kashan, A., (2010). "A differential evolution algorithm for the manufacturing cell formation problem using group based operators", *Expert Systems with Applications*, Vol. 37, No. 7, pp. 4822–4829.

Onwubolu, G.C., and Mutingi, M., (2001). "A genetic algorithm approach to cellular manufacturing systems", *Computers & Industrial Engineering*, Vol. 39, No. 1–2, pp. 125–144.

Pailla, A., Trindade, A.R., Parada, V., and Ochi, L.S., (2010). "A numerical comparison between simulated annealing and evolutionary approaches to the cell formation problem", *Expert Systems with Applications*, Vol. 37, No. 7, pp. 5476–5483.

Papaioannou, G., and Wilson, J.M., (2010). "The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research", *European Journal of Operational Research*, Vol. 206, No. 3, pp. 509–521.

Paydar, M.M., Mahdavi, I., Khonakdari, S.V., and Solimanpur, M., (2011). "Developing a mathematical model for cell formation in cellular manufacturing systems", *International Journal of Operational Research*, Vol. 11, No. 4, p. 408.

Paydar, M.M., Mahdavi, I., Sharafuddin, I., and Solimanpur, M., (2010). "Applying simulated annealing for designing cellular manufacturing systems using MDmTSP", *Computers & Industrial Engineering*, Vol. 59, No. 4, pp. 929–936.

Paydar, M.M., and Saidi-Mehrabad, M., (2013). "A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy", *Computers & Operations Research*, Vol. 40, No. 4, pp. 980–990.

Seifoddini, H., (1989). "A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications", *International Journal of Production Research*, Vol. 27, No. 7, pp. 1161–1165.

Stanfel, L.E., (1985). "Machine clustering for economic production", *Engineering Costs and Production Economics*, Vol. 9, No. 1–3, pp. 73–81.

Suresh Kumar, C., and Chandrasekharan, M.P., (1990). “"Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology", *International Journal of Production Research*, Vol. 28, No. 2, pp. 233–243.

Taguchi, G., (1986). *Introduction to Quality Engineering : Designing Quality into Products and Processes*.

Waghodekar, P.H., and Sahu, S., (1984). "Machine-component cell formation in group technology: MACE", *International Journal of Production Research*, Vol. 22, No. 6, pp. 937–948.

Wemmerlöv, U., and Hyer, N.L., (1989). "Cellular manufacturing in the U.S. industry: a survey of users", *International Journal of Production Research*, Vol. 27, No. 9, pp. 1511–1530.

Wu, T.-H., Chang, C.-C., and Chung, S.-H., (2008). "A simulated annealing algorithm for manufacturing cell formation problems", *Expert Systems with Applications*, Vol. 34, No. 3, pp. 1609–1617.

Xambre, A.R., and Vilarinho, P.M., (2003). "A simulated annealing approach for manufacturing cell formation with multiple identical machines", *European Journal of Operational Research*, Vol. 151, No. 2, pp. 434–446.

Yin, Y., and Yasuda, K., (2006). "Similarity coefficient methods applied to the cell formation problem: A taxonomy and review", *International Journal of Production Economics*, Vol. 101, No. 2, pp. 329–352.

This article can be cited: Firouzian, S., Mahdavi, I., Paydar, M.M., Saadat, M., (2020). " Simulated annealing and artificial immune system algorithms for cell formation with part family clustering", *Journal of Industrial Engineering and Management Studies*, Vol. 7, No. 1, pp. 191-219.