International Journal of Industrial Engineering & Production Research (2019) 30: 405- 427
DOI: 10.22068/ijiepr.30.4.405

RESEARCH PAPER

A New Mathematical Model for Integration of Cell Formation with Machine Layout and Cell Layout by Considering Alternative Process Routing Reliability: A Novel Hybrid Metaheuristic

Amir-Mohammad Golmohammadi[1], Mahboobeh Honarvar[*2], Guangdong Tian[3], Hasan Hosseini-Nasab[4]

Received 11June 2019; Revised 8 September 2019; Accepted 15 September 2019; Published online 1 December 2019

ABSTRACT
Cellular manufacturing systems are still quite popular among researchers, and they have proposed metaheuristics capable to solve such complex optimization models. In this study, machines are considered unreliable whose lifespan follows a Weibull distribution. The intra- and inter-cell movements of both parts and machines determined using batch sizes for transferring parts are related to the distances traveled through a rectilinear distance. The objectives of this study are to minimize the total cost of parts' relocations and maximize the reliability of the processing routes due to alternative process routing. To solve the proposed problem, Genetic Algorithm (GA) and two recent nature-inspired algorithms, including Keshtel Algorithm (KA) and Red Deer Algorithm (RDA), are employed. In addition, the main innovation of this paper is to propose a novel hybrid metaheuristic algorithm based on the benefits of the aforementioned algorithms. Some numerical instances are defined and solved by the proposed algorithms and, also, validated by the outputs of the exact solver. A real case study is also utilized to validate the proposed solution and modeling algorithms. The results indicate that the proposed hybrid algorithm is more appropriate than the exact solver and outperforms the individual ones.

KEYWORDS: *Cell formation, Cellular manufacturing system, Machine reliability, Cell layout, Weibull distribution, Meta-heuristic algorithms.*

1. Introduction

The facility design is a significant requirement in the field of manufacturing systems engineering. Approximately, $250 billion is spent annually in the U.S on facility designing, planning, and re-planning [1]. Minimizing material transferring movements may be among the initial reasons for developing the Cellular Manufacturing System (CMS) [2-4]. The main role of the CMS is to assign a number of parts and machines to each other to produce some cells on the basis of their similarities in the production process, design, and geometrical characteristics comprehensively [5].

Recent decades have seen a great deal of interest in employing different CMS applications. There are numerous advantages of the CMS including the reduction of material transferring costs, the setup time, the delivery time, the lot-size, and the amount of Work-In-Process (WIP) [6]. One of the crucial steps in designing a CMS is the Cell Formation (CF) problem that has been extensively studied in the literature [7]. According to aforementioned factors, the Exceptional Elements (EEs) are considered to be a common issue in the CMS of manufacturing environments recognized as the major obstacle in cell formation and its scheduling processes [8]. An EE can be defined generally as a product that needs to be produced in more than one cell and causes inter-cell transfer of materials.

The facility layout is a key element in designing a CMS considering the layout of machines within

* Corresponding author: *Mahboobeh Honarvar*
mhonarvar@yazd.ac.ir

1. *Department of Industrial Engineering, Yazd University, Yazd, Iran.*
2. *Department of Industrial Engineering, Yazd University, Yazd, Iran.*
3. *School of Mechanical Engineering, Shandong University, Jinan 250061, China.*
   Key Laboratory of High Efficiency and Clean Mechanical Manufacture (Ministry of Education), Jinan 250061, China.
4. *Department of Industrial Engineering, Yazd University, Yazd, Iran.*

cells (Intra-cell layout) and the layout of cells (Inter-cell Layout) on the shop floor. An efficient facility layout can reduce material handling cost, work-in-process, and throughput rate [9]. A competent layout not only enhances the performance of the system, but also minimizes the production costs by 40 to 50% on average [10]. Hence, incorporating the facility layout problem into the CMS design process is of high significance. However, layout design in CMS has not received much attention, since most of the relevant studies have only investigated the CFP [11-14]. As stated in [12], the decision-making for the facility layout and CF models is difficult due to the complexity of solving [15-18]. This issue is a motivation for developing several optimization algorithms, mainly including heuristics and metaheuristics, to solve these complicated optimization problems.

Some new trends such as the reliability of CMS have been seen in recent works. Most of them either investigate some of these decisions or handle all of them, yet in a sequential fashion [19-21]. On the other hand, to simplify the mathematical formulation, a majority of approaches in the field of facility layout and CF problems usually minimize the number of inter-cell movements or intra-cell movements or both [21-22]. Accordingly, to minimize the material handling cost, the exact information about facility layout design, in addition to the notion of distance, must be considered. Based on this drawback, the layout may be inefficient. Instead, this is a motivation to focus on U-form imposing structures in this research area.

After employing the benefits of cellular layouts, a target function accurately calculates the costs of inter -and intra-cell movements for parts. Note that the cells and the machines should not be overlapped in an accurate cellular layout. In a number of studies, machines located in each position can be assigned to any of the cells without any restrictions [23-24]. The proposed model under study prevents both cells and machines from being overlapped by imposing some restrictions on assigning the machines to the cells, as well as the cells to the shop floor. Furthermore, machine reliability and cost considerations are incorporated into the proposed model. Since machine reliability has a probabilistic nature, it is assumed that the time between failures follows Weibull distribution. Since the failure rate is constant in exponential distribution, in an industrial shop with mechanical machines, Weibull distribution can be appropriate for considering machine reliability because failure rates increase over time.

Considering different real-life constraints and practical assumptions makes the model difficult to solve. Based on the decisions of the proposed system at tactical and operational levels, CMS decision-makers consider the computational cost to be a very important element and wish to reduce it. Therefore, efficient solution algorithms are required to address this dilemma. Metaheuristics is a popular feasible alternative to solve this complicated problem, as evident in the literature [25-26]. As one of the NP-hard problems, this chance always exists even with a low possibility rate for a new metaheuristic to better solve such complicated models to obtain a near-optimal solution in a shorter amount of time [27-30]. This reason motivates several authors to employ different types of metaheuristics in this research area [1-10]. To cover the limitations of several metaheuristics employed in the literature, this study not only utilizes Genetic Algorithm (GA) and two recent metaheuristics, namely Red Deer Algorithm (RDA) [20] and Keshtel Algorithm (KA) [21-22], but also develops a new hybrid metaheuristic to consider the benefits of these individual algorithms. The main reason to use RDA and KA is that they are the main parties of the proposed hybrid metaheuristic algorithm. Therefore, we consider the above to make a better comparison so as to be convinced of the abilities of the proposed hybrid metaheuristic.

Given a general view of other sections of this paper, its remaining is structured as follows: Problem definition and formulation is discussed in Section 2. The proposed solution approaches including GA, RDA, KA and the developed hybrid metaheuristic are given in Section 3. Computational results and analyses of algorithms are explored in Section 4. Finally, concluding remarks and future research directions are provided in Section 6.

2. Proposed Model

The proposed model aims to choose concurrently the formation of cells, the layout of machines inside cells, and the layout of cells on the shop floor such that the total transportation cost of parts and reconfiguration cost of cells are minimized [23]. In the proposed model, the job shop configuration is considered for the intra-cellular layout. The proposed mixed-integer nonlinear programming model with a number of assumptions, parameters, and decision variables is discussed in the following.

2-1. Model assumption

To simulate the model, the following assumptions are taken into consideration:

- The flow between machines is determined. This number is obtained from the parts' demand and parts' operational paths.
- The material handling cost is calculated according to the center-to-center distance between machines through a rectilinear distance.
- The unit cost of inter- and intra-cell movements for each part type is predetermined.
- There is only one number of each machine type.
- The maximum capacity of cells is known and remains the same in the planning horizon.
- Machines are considered as squares of equal areas and, hence, are supposed to have a unit dimension. It has been examined that the proposed considerations provide a suitable approximation to the real-world conditions where machines are not exact squares or rectangles (Heragu, 1997). The cells are considered to be rectangular in shape.
- The time between failures follows Weibull distribution with known characteristics and failure rate (breakdown).

2-2. Notations

$i=\{1,2,\dots,,n\}$	Index set of parts;
$r=\{1,2,\dots,,R_i\}$	Index of process rout of parts;
$m,m'\{1,2,\dots,,M\}$	Index of machines;
$r=\{1,2,\dots,J_{ir}\}$	Index of action j for selected process rout of part i ;
$C,C'=\{1,2,\dots,,C\}$	Index of cells;

2-3. Model parameters

P_i	Demand of part i;
BS_i	Batch size for part i;
C	Number of cells;
UC	Maximum number of machines in one cell;
R_i	Number of process routes of part i;
J_{ir}	Number of operations of part i under process rout r;
C_i^{inter}	The inter-cell material handling cost for transporting part j per unit distance; ($/unit)
C_i^{intra}	The intra-cell material handling cost for transporting part j per unit distance; ($/unit)
E	The horizontal length of the shop floor; (The length of the shop floor)
F	The vertical length of the job shop; (The width of the shop floor)
$A_{cc'},B_{cc}$	The zero and one random variable;
N	An appropriate large positive number;

2-4. Decision variables

$Z_{ir}=\begin{cases}1 & \text{If rout } r \text{ is selected for part } i\\ 0 & \text{Otherwise}\end{cases}$

$V_{mc}=\begin{cases}1 & \text{If machine } m \text{ is allocated to cell } c\\ 0 & \text{Otherwise}\end{cases}$

x_{m}	The horizontal coordinate of the center of machine i
y_m	The vertical coordinate of the center of machine i
p_c^1	The horizontal coordinate of the left side of cell k
p_c^2	The horizontal coordinate of the right side of cell k
q_c^1	The vertical coordinate of the bottom side of cell c
q_c^2	The vertical coordinate of the top side of cell c

2-5. Mathematical formulation

Here, the first initial model is proposed. With respect to the aforementioned notations and illustrated assumptions, the nonlinear mixed-integer programming model presented to formulate the proposed problem is as follows:

$$Min\,TC = Inter\_Cell + Intra\_Cell \tag{1}$$

$$Subject\ To:$$

$$Inter\_Cell = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{\substack{c=1\\c\neq c'}}^{C}\sum_{c'=1}^{C} C_i^{inter} Z_{ir}\left[\frac{P_i}{BS_i}\right] V_{M_{ir}^j c} V_{M_{ir}^{j+1} c'}\left(\left|x_{M_{ir}^j}-x_{M_{ir}^{j+1}}\right| + \left|y_{M_{ir}^j}-y_{M_{ir}^{j+1}}\right|\right) \tag{2}$$

$$Intra\_Cell = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{c=1}^{C} C_i^{intra} Z_{ir}\left[\frac{P_i}{BS_i}\right] V_{M_{ir}^{j}c} V_{M_{ir}^{j+1}c}\left(\left|x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}}\right| + \left|y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}}\right|\right) \quad (3)$$

$$\sum_{r=1}^{R_i} Z_{ir} = 1 \qquad \forall\, i = 1,2,\dots,n \quad (4)$$

$$\sum_{c=1}^{C} V_{mc} = 1 \qquad \forall\, m = 1,2,\dots,M \quad (5)$$

$$\sum_{m=1}^{M} V_{mc} \le UC \qquad \forall\, c = 1,2,\dots,C \quad (6)$$

$$|x_m - x_{m'}| + |y_m - y_{m'}| \ge 1 \quad (7)$$

$$\begin{cases} x_m \ge p_c^1 + N(1-V_{mc}) \\ x_m \le p_c^2 + N(1-V_{mc}) \\ y_m \ge q_c^1 + N(1-V_{mc}) \\ y_m \le q_c^2 + N(1-V_{mc}) \end{cases} \qquad \forall m, c \quad (8)$$

$$\begin{cases} p_c^1 \ge 0 \\ q_c^1 \ge 0 \\ p_c^2 \le E \\ q_c^2 \le F \end{cases} \qquad \forall c \quad (9)$$

$$\begin{cases} p_c^1 - p_c^2 + NA_{cc'} + NB_{cc'} \ge 0 \\ p_c^2 - p_c^1 - NA_{cc'} - N(1-B_{cc'}) \le 0 \\ q_c^1 - q_c^2 + N(1-A_{cc'}) + NB_{cc'} \ge 0 \\ q_c^2 - q_c^1 - N(1-A_{cc'}) - N(1-B_{cc'}) \end{cases} \qquad 0 \le c < c' \le C \quad (10)$$

$$Z_{ir}, V_{mc} \in \{0,1\}$$
$$x_m, y_m, p_c^1, p_c^2, q_c^1, q_c^2 \ge 0 \quad and\ integer \quad (11)$$

The first term of the objective function represents the inter-cellular material transferring costs, and the second term represents the intra-cellular material transferring cost. The first set of constraints (Eq.4) guarantees that each machine is assigned to only one cell. The second constraint (Eq.5) ensures that each part is assigned to a single part family. The number of machines in a single cell is limited by constraint (Eq.6). The fifth constraint (Eq.7) prevents machines from being overlapped. As mentioned earlier, the machines are considered as squares with a unit dimension. The set of relationships (Eq.8) indicates that each machine must be relocated into the space of its corresponding cell. The next constraint (Eq.9) is developed to control the cells, which are in space of the job shop. The set of relationships (Eq.10) prevents cells from being overlapped. The set of relationships (Eq.11) ensures that Z_{ir} and V_{mc} are binary variables.

2-6. Linearization of the proposed model

Generally, the proposed CMS problem is NP-hard. Due to non-linear constraints, the complexity of the model increases hugely. Hence, finding an optimal solution for the model with non-linear constraints is quite time-consuming and may result in local optima. The proposed model is a non-linear mixed-integer programming model due to the absolute terms in Eqs. (2), (3), and (7). Here, these terms are linearized to enhance the efficiency of the model. To linearize these terms, a very common procedure is employed, which has been used in many scientific papers such as [24-30]. Thus, for the linearization of Eq. (9), non-negative variables $x_{ii'h}^{+}$, $x_{ii'h}^{-}$, $y_{ii'h}^{+}$, and $y_{ii'h}^{-}$ are presented as follows:

$$R_{irjcc'} = Z_{ir} \times V_{M_{ir}^{j}c} \times V_{M_{ir}^{j+1}c'} \quad (12)$$

$$S_{irjc} = Z_{ir} \times V_{M_{ir}^{j}c} \times V_{M_{ir}^{j+1}c} \quad (13)$$

$R_{irjcc'}$ equals 1 when the r^{th} process rout is selected for part i, the j^{th} operation is processed in cell c, and the $(j+1)^{th}$ operation is processed in cell c'. S_{irjc} equals 1 when process rout r is selected for part i and operations j and $(j+1)$ are processed in cell c. Considering these new variables in the objective function, the following constraints are added to the model.

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c'} - R_{irjcc'} \le 2 \quad \forall i,r,j,c,c' \quad (14)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c'} - 3R_{irjcc'} \ge 0 \quad \forall i,r,j,c,c' \quad (15)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c} - S_{irjc} \le 2 \quad \forall i,r,j,c \quad (16)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c} - 3S_{irjc} \ge 0 \quad \forall i,r,j,c \quad (17)$$

As such, the following steps are considered to linearize Constraint (7):

$$x^{+}_{M_{ir}^{j,j+1}} = \begin{cases} \left(x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}}\right) & if \quad x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}} > 0 \\ 0 & otherwise \end{cases} \quad (18)$$

$$x^{-}_{M_{ir}^{j,j+1}} = \begin{cases} \left(x_{M_{ir}^{j+1}} - x_{M_{ir}^{j}}\right) & if \quad x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}} \le 0 \\ 0 & otherwise \end{cases} \quad (19)$$

$$y^{+}_{M_{ir}^{j,j+1}} = \begin{cases} \left(y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}}\right) & if \quad y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}} > 0 \\ 0 & otherwise \end{cases} \quad (20)$$

$$y^{-}_{M_{ir}^{j,j+1}} = \begin{cases} \left(y_{M_{ir}^{j+1}} - y_{M_{ir}^{j}}\right) & if \quad y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}} \le 0 \\ 0 & otherwise \end{cases} \quad (21)$$

Then, the following terms can be defined to linearize the model:

$$\left|x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}}\right| = x^{+}_{M_{ir}^{j,j+1}} + x^{-}_{M_{ir}^{j,j+1}} \quad (22)$$

$$x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}} = x^{+}_{M_{ir}^{j,j+1}} - x^{-}_{M_{ir}^{j,j+1}} \quad (23)$$

$$\left|y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}}\right| = y^{+}_{M_{ir}^{j,j+1}} + y^{-}_{M_{ir}^{j,j+1}} \quad (24)$$

$$y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}} = y^{+}_{M_{ir}^{j,j+1}} - y^{-}_{M_{ir}^{j,j+1}} \quad (25)$$

Now, by considering the aforementioned formulations, the transformed mixed-integer linear programming formulation is presented as follows:

$$Min\ TC = Inter\_Cell + Intra\_Cell$$

$$Subject\ To:$$

$$Inter\_Cell = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{\substack{c=1\\c\neq c'}}^{C}\sum_{c'=1}^{C} C_i^{inter}\left[\frac{P_i}{BS_i}\right] R_{irjcc'}\left(x^{+}_{M_{ir}^{j,j+1}} + x^{-}_{M_{ir}^{j,j+1}} + y^{+}_{M_{ir}^{j,j+1}} + y^{-}_{M_{ir}^{j,j+1}}\right) \quad (26)$$

$$Intra\_Cell = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{c=1}^{C} C_i^{intra}\left[\frac{P_i}{BS_i}\right] S_{irjc}\left(x^{+}_{M_{ir}^{j,j+1}} + x^{-}_{M_{ir}^{j,j+1}} + y^{+}_{M_{ir}^{j,j+1}} + y^{-}_{M_{ir}^{j,j+1}}\right) \quad (27)$$

$$\sum_{r=1}^{R_i} Z_{ir} = 1 \quad \forall\, i = 1,2,\dots,n \quad (28)$$

$$\sum_{c=1}^{C} V_{mc} = 1 \quad \forall\, m = 1,2,\dots,M \quad (29)$$

$$\sum_{m=1}^{M} V_{mc} \le UC \quad \forall\, c = 1,2,\dots,C \quad (30)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c'} - R_{irjcc'} \le 2 \quad \forall i,r,j,c,c' \quad (31)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c'} - 3R_{irjcc'} \ge 0 \quad \forall i,r,j,c,c' \quad (32)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c} - S_{irjc} \le 2 \quad \forall i,r,j,c \quad (33)$$

$$Z_{ir} + V_{M_{ir}^{j}c} + V_{M_{ir}^{j+1}c} - 3S_{irjc} \geq 0 \qquad \forall i,r,j,c \qquad (34)$$

$$\begin{cases} x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}} = x^{+}_{M_{ir}^{j,j+1}} - x^{-}_{M_{ir}^{j,j+1}} \\ y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}} = y^{+}_{M_{ir}^{j,j+1}} - y^{-}_{M_{ir}^{j,j+1}} \end{cases} \qquad \forall m,m' \qquad (35)$$

$$\begin{cases} x_{M_{ir}^{j}} - x_{M_{ir}^{j+1}} + NA_{j,j+1} + NB_{j,j+1} \geq 1 \\ x_{M_{ir}^{j+1}} - x_{M_{ir}^{j}} - NA_{j,j+1} - N(1 - B_{j,j+1}) \geq 1 \\ y_{M_{ir}^{j}} - y_{M_{ir}^{j+1}} + N\left(1 - A_{j,j+1}\right) + NB_{j,j+1} \geq 1 \\ y_{M_{ir}^{j+1}} - y_{M_{ir}^{j}} - N\left(1 - A_{j,j+1}\right) - N(1 - B_{j,j+1}) \geq 1 \end{cases} \qquad \forall 1 \leq m < m \leq M \qquad (36)$$

$$\begin{cases} x_m \geq p_c^1 + N(1 - V_{mc}) \\ x_m \leq p_c^2 + N(1 - V_{mc}) \\ y_m \geq q_c^1 + N(1 - V_{mc}) \\ y_m \leq q_c^2 + N(1 - V_{mc}) \end{cases} \qquad \forall m,c \qquad (37)$$

$$\begin{cases} p_c^1 \geq 0 \\ q_c^1 \geq 0 \\ p_c^2 \leq E \\ q_c^2 \leq F \end{cases} \qquad \forall c \qquad (38)$$

$$\begin{cases} p_c^1 - p_c^2 + NA_{cc'} + NB_{cc'} \geq 0 \\ p_c^2 - p_c^1 - NA_{cc'} - N(1 - B_{cc'}) \leq 0 \\ q_c^1 - q_c^2 + N(1 - A_{cc'}) + NB_{cc'} \geq 0 \\ q_c^2 - q_c^1 - N(1 - A_{cc'}) - N(1 - B_{cc'}) \leq 0 \end{cases} \qquad 0 \leq c < c' \leq C \qquad (39)$$

$$\begin{gathered} Z_{ir}, V_{mc}, R_{irjcc'}, S_{irjc} \in \{0,1\} \\ x_m, y_m, p_c^1, p_c^2, q_c^1, q_c^2 \geq 0 \quad \textit{and integer} \end{gathered} \qquad (40)$$

2-7. Reliability in the CMS design

It is known here that the failure rate is constant in exponential distribution and is appropriate for estimating the reliability of electrical systems; however, in a manufacturing system with mechanical machines, given the passage of time and consequent depreciation of machines, the assumption of a fixed failure rate is unrealistic during the production period. Hence, the Weibull distribution can be more appropriate for considering machine reliability [30-35].

Since machine reliability has a probabilistic nature, it is assumed that machine reliability follows a Weibull distribution with a known failure (breakdown) rate. As such, machine breakdown cost is assumed to be known in advance and is based on its repair, install/uninstall costs.

Assuming that the failure of the machine follows the Weibull distribution, the reliability for the machine type *j* is as follows:

$$R_j(t) = exp\left[-\left(\frac{t}{\theta_j}\right)^{\beta_j}\right] \qquad (41)$$

In this regard:

t	Planned time period
θ_j	Characteristic of the lifespan for machine *j*
β_j	Parameter shape for machine *j*

Therefore, there are three modes for the parameter β:

$\beta = 1$	A fixed failure rate and a reliability function are considered to be exponential, where an average lifespan equals $\theta = 1 / \lambda$
$\beta < 1$	The failure rate is decreasing.
$\beta > 1$	The failure rate is increasing.

It is also known that MTBF, MTTR, and MTTF data function in the following equations:

$$MTBF = MTTF + MTTR \qquad (42)$$

If we assume that MTBF is known for all the machines, the MTTF Weibull failure model is equal to:

$$MTTF = \theta\Gamma(1 + \frac{1}{\beta}) \qquad (43)$$

For a system to be repaired, when the failures are analyzed for MTTF estimation, MTTR can be eliminated and MTTF is considered to be equal to MTBF [36].
Assume that, in an operational route, there are a number of machines (1-3-2) as follows:

M_1 M_3 M_2

Fig. 1. An operational route

By taking into account the reliability of each machine in the above operational route, based on a series system, the reliability of the route is calculated as follows:

$$R_{s_{(2-3-1)}} = exp\left[-\left(\frac{t}{\theta_2}\right)^{\beta_2}\right] \times exp\left[-\left(\frac{t}{\theta_3}\right)^{\beta_3}\right] \times exp\left[-\left(\frac{t}{\theta_1}\right)^{\beta_1}\right] \quad (44)$$

This equation can be simplified in the following ways:

$$R_{s_{(2-3-1)}} = exp\left\{-\left[\left(\frac{t}{\theta_2}\right)^{\beta_2} + \left(\frac{t}{\theta_3}\right)^{\beta_3} + \left(\frac{t}{\theta_1}\right)^{\beta_1}\right]\right\} \quad (45)$$

$$R_{s_{(2-3-1)}} = exp\left\{-\sum_{j=1,2,3}\left(\frac{t}{\theta_j}\right)^{\beta_j}\right\} \quad (46)$$

$$Ln\, R_{s_{(2-3-1)}} = -\sum_{j=1,2,3}\left(\frac{t}{\theta_j}\right)^{\beta_j} \quad (47)$$

$$Ln\, \frac{1}{R_{s_{(2-3-1)}}} = \sum_{j=1,2,3}\left(\frac{t}{\theta_j}\right)^{\beta_j} \quad (48)$$

We show $Ln\, \frac{1}{R_{s_{(2-3-1)}}}$ with $LIR_{(2-3-1)}$. It is clear that the LIR is reliability reversal; then, as long as the objective is to maximize reliability, the LIR must be minimized. Moreover, by obtaining θ according to the MTBF and β, we can write the following:

$$MTBF = \theta\Gamma\left(1+\frac{1}{\beta}\right) \;\rightarrow\; \theta = \frac{MTBF}{\Gamma(1+\frac{1}{\beta})} \quad (49)$$

Moreover, by inserting them in Equation (49), we have:

$$Ln\, \frac{1}{R_s} = \sum_j\left(\frac{t}{\theta_j}\right)^{\beta_j} = \sum_j\left(\frac{t}{\frac{MTBF}{\Gamma(1+\frac{1}{\beta})}}\right)^{\beta_j} = \sum_j\left(\frac{t\,.\Gamma(1+\frac{1}{\beta})}{MTBF}\right)^{\beta_j} = LIR \quad (50)$$

Therefore, this equation can be used to obtain the maximum processing reliability route such that the selection of this route affects the layout of the machines in the cells.
Then, by considering reliability and input parameters and variables, the objective function is rewritten as follows:

$$Min\, TC = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{c=1}^{C}\sum_{\substack{c'=1 \\ c\neq c'}}^{C} C_i^{inter}\left[\frac{P_i}{BS_i}\right]R_{irjcc'}\left(x^{+}_{M_{ir}^{j,j+1}} + x^{-}_{M_{ir}^{j,j+1}} + y^{+}_{M_{ir}^{j,j+1}} + y^{-}_{M_{ir}^{j,j+1}}\right) + \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{j=1}^{J_{ir}-1}\sum_{c=1}^{C} C_i^{intra}\left[\frac{P_i}{BS_i}\right]S_{irjc}(x^{+}_{M_{ir}^{j,j+1}} + x^{-}_{M_{ir}^{j,j+1}} + y^{+}_{M_{ir}^{j,j+1}} + y^{-}_{M_{ir}^{j,j+1}}) \quad (51)$$

$$Min\, Z = \sum_{i=1}^{n}\sum_{r=1}^{R_i} LIR_{ir} \quad (52)$$

$Subject\ to$:

$$LIR_{ir} = \sum_{i=1}^{n}\sum_{r=1}^{R_i}\sum_{c=1}^{C}\sum_{j=1}^{J_{ir}} \left(\frac{t\,.\,\Gamma(1+\frac{1}{\beta})}{MTBF}\right)^{\beta_j} Z_{ir}\, V_{M_{ir}^{j}c} \quad (53)$$

Eqs. (28) to (40)

3. Proposed Solution Algorithm

As stated in the literature [10-20], the CMS scheduling models are non-polynomial time hard problems that are difficult to solve using different types of exact methods. In addition to the natural complexity, considering the reliability of the alternative processing routes increases its difficulty and combinatorial nature. Hence, meta-heuristic approaches should be employed to obtain satisfying solutions in a reasonable time. Several algorithms have been applied in the context of a CMS design. One of the most popular algorithms is the Genetic Algorithm (GA). This motivated us to use GA in this study based on previous works in the literature [15-18]. Due to a No Free Lunch theory, this chance for a new metaheuristic always exists to reveal a better output in comparison with other existing algorithms [34-38]. With regard to this theory, this study employs two recent nature-inspired metaheuristics including Keshtel Algorithm (KA) and Red Deer Algorithm (RDA). Last but not the least, the main innovation of this study is to propose a novel hybrid metaheuristic based on the advantages of the aforementioned algorithms. Here, first of all, the encoding plan of metaheuristics is illustrated and, after that, the procedures of metaheuristics utilized are addressed in the following sub-sections. Note that, due to the proposed bi-objective model, both functions have merged by two weights into a single objective optimization problem.

3-1. Solution representation

To show how the model constraints would be handled by the presented solution algorithms, an encoding plan is given to consider the solution representation in the format of a string-based solution representation [35-40]. Each solution represents a feasible answer for the proposed mathematical model in the search space and is formed from a sequence of a fixed number of genes from a certain alphabet. The alphabet can be a set of binary and integer numbers, matrices, or a combination of characters. These ways are considered to determine how a problem is formulated in the form of an algorithm and what metaheuristic operators are applied.

Figs. 2 and 3 illustrate general and detailed views of the solution presentation of algorithms related to the machine's alignment for the manufacturing cells, respectively.

[[Z] [X] [Y]]

Fig. 2. General view of the solution representation.

Z_{11} Z_{12} … Z_{1M}	x_{11} x_{12} … x_{1M}	y_{11} y_{12} … y_{1M}
This part assigns machines to cells	This part represents the X components of machines	This part represents the Y components of machines

Fig. 3. Detailed view of solution representation in the first step.

3-2. Genetic algorithm

The GA is a well-known evolutionary algorithm by considering each solution as a chromosome and performing the mutation and crossover operators to search the feasible area. The processing principle in this algorithm is random and guided in an optimum place. Appropriate genetic operators (i.e., mutation and crossover) are employed to obtain new solutions [40-42].

The mutation operator randomly alters the genetic composition of a chromosome according to a small probability of mutation.

The crossover operator is applied to every pair of randomly selected chromosomes as parents and combines their information to generate two offspring [41]. In the proposed algorithm, the crossover operator substitutes a part of a row from one parent with the same part of the same row of another parent in order to generate two offspring similar to both parents [42]. In this study, the parameterized uniform crossover is employed. The first parent is chosen amongst the best individuals in the population, while the other one is chosen randomly from the whole population. Then, a real random number row is produced for each row at an interval [0,1]. If the random number is larger than a predetermined threshold value, called crossover probability (CProb), then the allele of the first parent is applied. Otherwise, the allele of the second parent is applied to generate the offspring. An example process of the crossover is given in Fig. 4. In this example, Offspring 1 inherits the gene of Parent 1 with at a probability level of 0.5 and inherits the gene of Parent 2 at a probability level of 0.5.

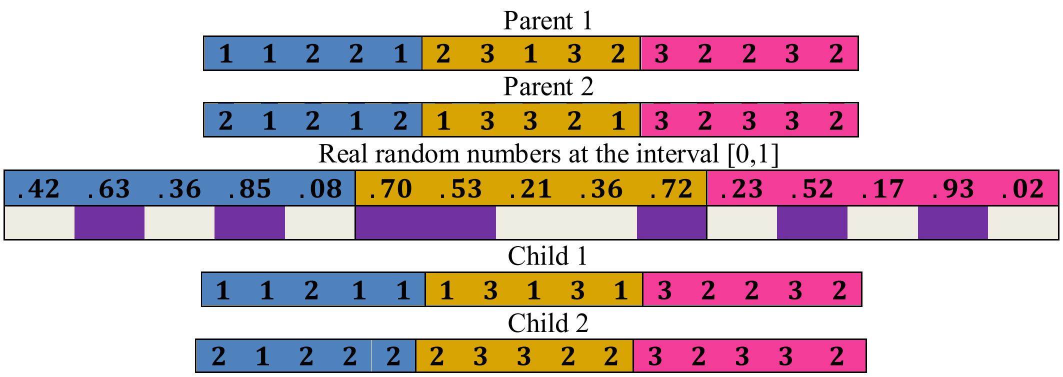

Fig. 4. Example process of the crossover

After performing mutation and crossover and choosing the next generation of solutions, to stop GA and present a final solution, stopping criterion should be considered. In the proposed algorithm, the specified maximum number of the generations is used as the stopping criterion.

3-3. Keshtel algorithm (KA)

The Keshtel algorithm is an optimizer developed by Hjiaghaei-Keshteli and Aminnayeri in 2013 [21]. The main inspiration of KA is the feeding behavior of a dabbling dock called Keshtel. Their swirling process is the main property of this animal to search the food in a lake [19].

As a metaheuristic, the initial population called Keshtel is divided into three types (i.e., N_1, N_2, and N_3). N_1 includes some Keshtels that found good foods in the first time and called them lucky Keshtels. As such, N_3 includes the worst solutions. The lucky Keshtels search for more food around them [23].

The steps in KA are developed such that the user can employ just one or two operators among the three ones. According to the best of our knowledge and previous works [19-23], there is no attempt to employ this metaheuristic in the literature of CMS scheduling problems. Therefore, this study proposes KA by forming the solution representation of the proposed model. Given more details about this metaheuristic, the steps of KA are addressed in Fig. 5.

Initialize Keshtels population.
Calculate the fitness and sort them in three types: N_1, N_2, and N_3
X*=the best solution.
while (t< maximum number of iterations)
for each N_1
Calculate the distance between this lucky Keshtel and all Keshtels.
Select the closest neighbor.
S=0;
while (S< maximum number of swirling)
Do the swirling.
if the fitness (at least, one of objective functions has improved) of this new position is better than the previous one
Update this lucky Keshtel.

break
endif
S=S+1
endwhile
endfor
for each N_2
Move the Keshtel between the two Keshtels, randomly.
endfor
for each N_3
Create a random solution.
endfor
Merge N_1, N_2, and N_3 .
Sort the Keshtels and form N_1, N_2, and N_3 for the next iteration.
Update X* if there is a better solution.
t=t+1;
end while
return X*

Fig. 5. The pseudo-code of KA

3-4. Red deer algorithm (RDA)

There is a great deal of interest in developing new nature-inspired metaheuristics by focusing on the two important phases including exploration and exploitation and their trade-offs. The Red Deer algorithm introduced by Fathollahi Fard and Hajiaghaei-Keshteli [20] is one of the first successful methods among recent metaheuristics to give an opportunity to a user to make a balance between the exploitation and exploration phases, easily [21-22]. The Scottish Red Deer (*Cervus Elaphus Scoticus*) is a subspecies of Red Deer and lives in British Isles [23]. The males aim to extend their territory and the number of hinds in their harems. Generally, as a metaheuristic, the RDA starts with an initial population, called Red Deers (RD). They are divided into two types: hinds and male RDs. The roaring, fighting, and mating operators are the main procedures of this algorithm. According to the best of our knowledge [19-23], there has been no attempt to use this algorithm in the area of CMS scheduling problems. In this regard, this is the first attempt to employ this recent nature-inspired optimizer to solve the proposed problem. In what follows, the steps of the algorithm are detailed by considering a multi-objective version of RDA, as seen in Fig. 6.

Initialize the Red Deers population.
Calculate the fitness, sort them, and form the hinds (N_{hind}) and male RDs (N_{male}).
X*=the best solution.
while (*t*< maximum number of iterations)
for each male RD
A local search near his position.
Update the position if better than the prior ones (at least, one of objectives among all has been improved).
end for
Sort the males and, also, form the stags and the commanders.
for each male commander
Fight between male commander and stag.
Update the position of the male commander and stag.
end for

Form harems.
for each male commander
Mate male commander with the selected hinds of his harem randomly.
Select a harem randomly and name it *k*.
Mate male commander with some of the selected hinds of the harem.
end for
for each stag
Calculate the distance between the stag and all hinds and select the nearest hind.
Mate stag with the selected hind.
end for
Select the next generation with roulette wheel selection.
Update the X* if there is a better solution.
t=t+1;
end while
return X*

Fig. 6. The pseudo-code of RDA

3-5. Novel hybrid metaheuristic (H-RDKGA)

Today has seen a rapid development of hybrid capable metaheuristics to solve the complex engineering design problems [46-51]. The main innovation of this study is to propose a new hybrid metaheuristic called H-RDKGA. As indicated from the previous discussion, the KA is very good at doing the exploitation action. It appears that the swirling process can be done instead of two processes including roaring and fighting in RDA. Accordingly, for each male, the closest neighbor is specified and the swirling action is done. Due to the mating process, the GA mechanism is considered in this regard. Considering a brief illustration, the KA chooses the intensification properties, and the GA is measured in the diversification phase. This opinion is employed to examine the proposed method with their individual methods and also other feasible alternatives for combinations. Given more details of the proposed H-RDKGA, a pseudo-code is provided, as seen in Fig. 7.

Initialize the Red Deers population.
Calculate the fitness, sort them, and form the hinds (N_{hind}) and male RDs (N_{male}).
X*=the best solution.
while (*t*< maximum number of iterations)
for each male RD
Calculate the distance between this male and all males.
Select the closest neighbor.
S=0;
while (S< maximum number of swirling)
Do the swirling.
if the fitness of this new position is better than the previous one
Update this lucky male.
break
endif
S=S+1
endwhile
endfor
Sort the males and, also, form the stags and the commanders.

for each male commander
Select a hind by roulette wheel selection.
Mate (Crossover) male commander with the selected hind.
end for
for each stag
Select a hind randomly.
Mate (Crossover) stag with the selected hind.
end for
Select the next generation via roulette wheel selection.
Update X* if there is a better solution.
t=t+1;
end while
return X*

Fig. 7. The pseudo-code of H-RDKGA

4. Computational Results

A comparative study is presented in this section. First of all, to enhance the performance of employed metaheuristics and make a fair comparison, a full factorial design method is applied to tune the algorithms' parameters properly. After that, a numerical instance is provided to show the validation of the model. In the end, an extensive comparison among metaheuristics based on different criteria is presented in the following subsections.

4-1. Tuning of metaheuristics

As a metaheuristic, it is essential to calibrate the parameters of the algorithms [42-45]. A full factorial method is employed to achieve this goal in this study.

As given in the solution algorithm, the main parameters under consideration for GA include the population size, the maximum number of iterations, and mutation and crossover rates. In the proposed KA, the population size, maximum number of iterations, the percentage of *N1* and *N2*, and maximum number of swirling are the key parameters. As such, the main parameters of RDA include the population size, the maximum number of iterations, number of males, and the rate of alpha, beta, and gamma. At last, the proposed parameters of H-RDKGA include only the population size, maximum number of iterations, the number of males, and the maximum number of swirling. The levels of algorithms' parameters and their tuned values are given in Table 1. The results of Relative Percentage Deviation (RPD) [44-45] are depicted to show the best level of algorithms' parameters, as seen in Fig. 8.

Tab. 1. Tuning of metaheuristics

Metaheuristic	Parameters	Levels			Tuned value
GA	Population size	100	150	200	200
	Maximum number of iterations	300	500	700	500
	Rate of mutation	0.05	0.15	0.25	0.15
	Rate of crossover	0.6	0.7	0.8	0.8
KA	Population size	100	150	200	100
	Maximum number of iterations	300	500	700	300
	Percentage of N1	0.1	0.2	0.3	0.1
	Percentage of N2	0.4	0.5	0.6	0.6
	Maximum number of swirling	5	10	15	10
RDA	Population size	100	150	200	150
	Maximum number of iterations	300	500	700	700
	Number of males	15	25	30	25
	Alpha	0.5	0.6	0.7	0.6
	Beta	0.7	0.8	0.9	0.7
	Gamma	0.8	0.9	1	0.8
H-RDKGA	Population size	100	150	200	150
	Maximum number of iterations	300	500	700	500
	Number of males	15	25	30	30
	Maximum number of swirling	5	10	15	15

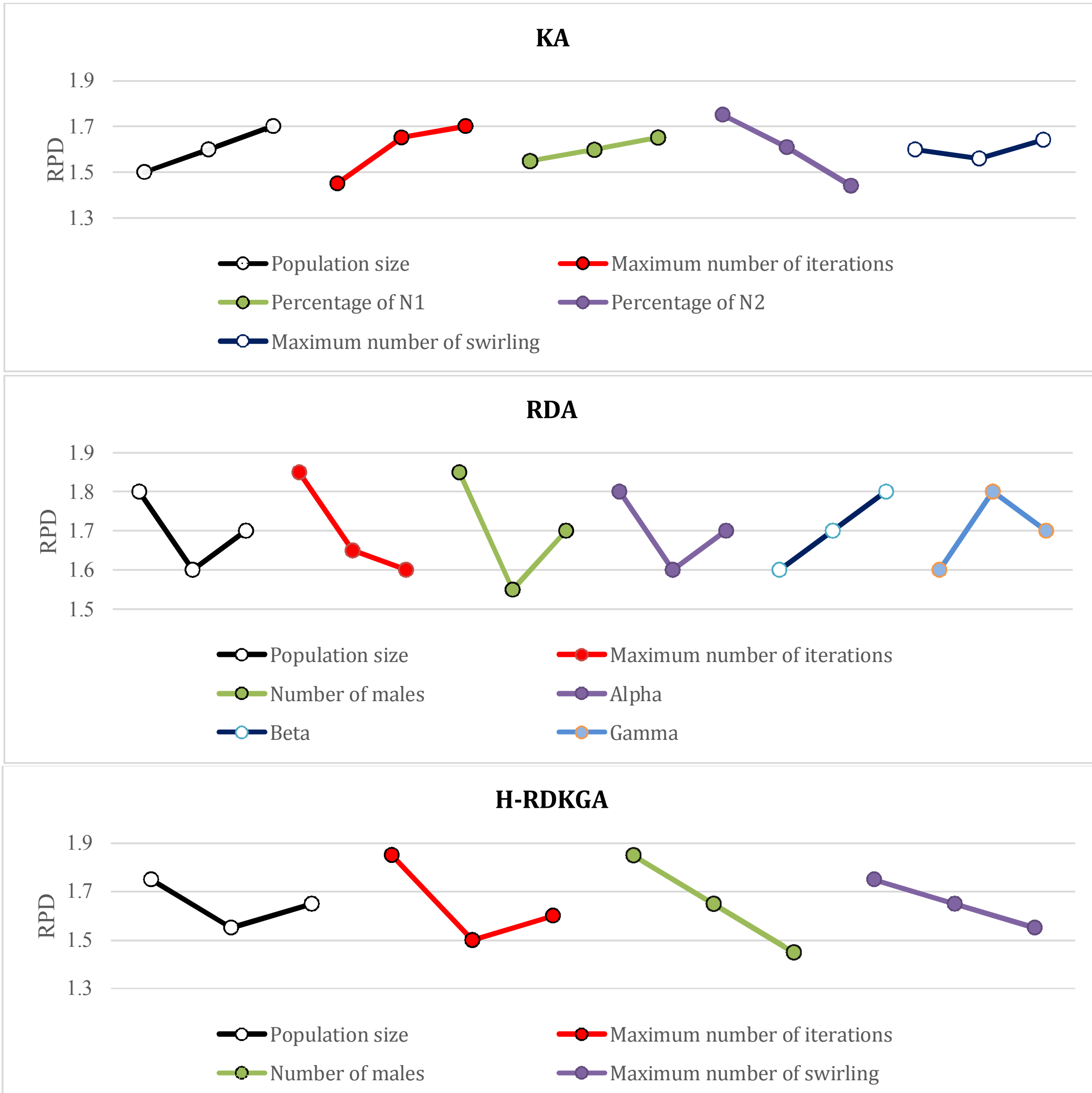

Fig. 8. The results of RPD for algorithms

4-2. Numerical example of the proposed model

Here, a numerical example based on an industrial example in the Netherlands to validate the proposed formulation is provided. All data are collected from Bata Industrials Company. This example includes 8 parts and 5 machines in which all of the presented hypotheses in the "model assumption section" are valid. Our goal is to determine machine locations and cell allocation by considering the route that has the highest reliability. All of the input parameters for a numerical example (12-machine & 12-part) are depicted in Tables 2 and 3, and the part-machine relations are shown by the following matrix (Fig. 9):

Tab. 2. Production information for the problem 12*12

PN	PD	RN	PS	PT (min)
1	100	1	6—5—3—12—8—11	2—6—1—8—1—5
		2	10—11—6—5—7	6—3—4—5—8
2	125	1	10—2—4—1—5	2—3—3—9—6

		2	4—1—1—3—6	2—8—3—5—4
		3	12—2—6	6—1—7
3	110	1	8—5—2—12	6—7—1—9
		2	12—8	9—7
4	120	1	9—2—4	10—3—10
		2	2—7—3—11—12	1—3—2—9—7
5	200	1	1—7—4—2—9	10—4—2—1—8
6	125	1	12—3—2—11—8—5	4—1—2—6—9—7
		2	11—10—5—8	2—6—5—4
7	90	1	10—7—11—5	3—5—4—10
		2	3—4—10—7	2—3—9—6
8	50	1	5—2—4	1—3—4
9	80	1	6—7—11—3—2	8—6—5—1—2
		2	2—3—11—6	2—2—4—5
10	50	1	4—8—5	1—4—3
11	60	1	3—2—10—9—12	5—4—1—6—4
12	45	1	6—7—2	3—4—5

PN: Part Number
RT: Routing Number
PT: Production Time
PD: Production Demand
PS: Production Sequence

Tab. 3. Parameters for the Problem 12*12

M	n	BS_i	C^j_{intra}	C^j_{inter}	C	E	F	UC
12	12	10	1	10	3	5	5	5

M/P	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
M_1		1			1							
M_2		1	1	1	1	1		1	1		1	1
M_3	1					1			1		1	
M_4		1		1	1			1		1		
M_5	1	1	1			1	1	1		1		
M_6	1								1			1
M_7					1		1		1			1
M_8	1		1			1				1		
M_9				1	1						1	
M_{10}		1					1				1	
M_{11}	1					1	1		1			
M_{12}	1		1			1					1	

M.P: Machine/Part

Fig. 9. Machine-Part Matrix for the problem 12×12 under processing rote 1

The model of the problem is coded and solved using GAMS software, run on a Pentium 4 personal computer (PC). The results were obtained after two hours of launching the program. The parts, machines, and locations assignments to cells and also layouts for machines and cells are illustrated in Table 4. For example, in Cell 1, Machines 1, 2, 4, 7, and 9 are assigned and also processing rote 1 are selected for them, although just one part is dedicated to Cell 3 and the second processing rote is chosen for it. Moreover, the optimal machine-part matrix and achieved layout without considering machine reliability are shown in Figs. 10 and 11.

Tab. 4. The results for the problem 12*12 without considering machine reliability

Cells	1	2	3
Machines	1 2 4 7 9	3 5 6 10 11	8 12
Parts (Best Routing)	4(1) 5(1) 8(1) 11(1) 12(1)	1(2) 2(2) 6(2) 7(1) 9(2) 10(1)	3(2)

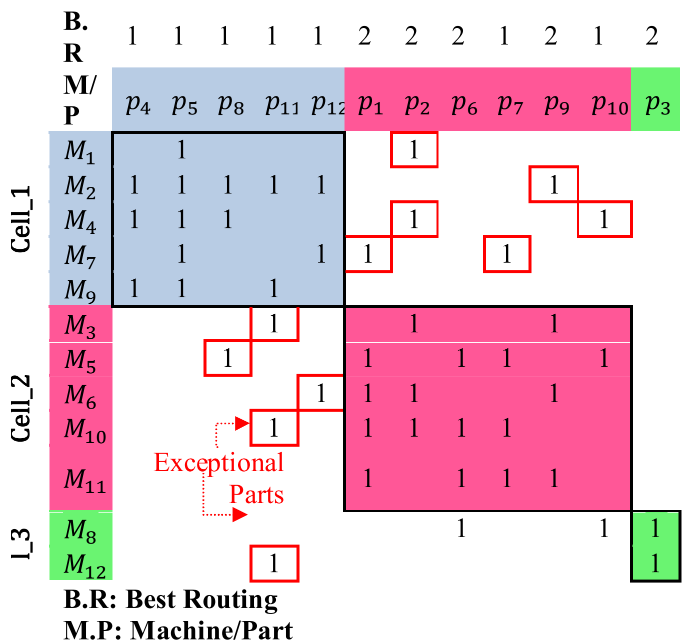

Fig. 10. Optimal machine-part matrix without considering machine reliability

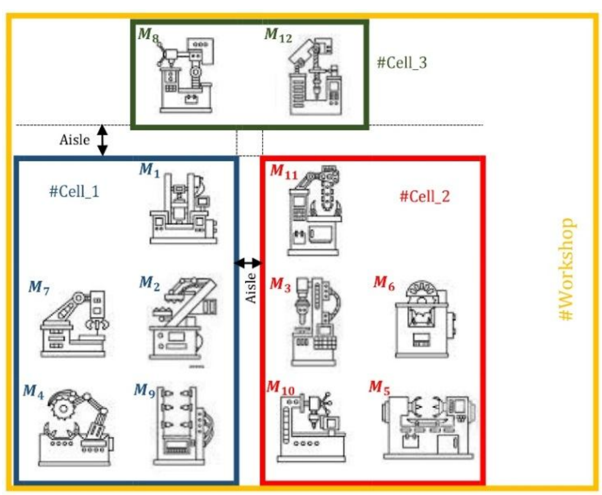

Fig. 11. Designed by solving model without considering machine reliability

In the second approach, by considering machine reliability, the locations' assignments and layouts will change. The parts, machines, and locations' assignments to cells and also optimal machine-part matrix and achieved layout in the new condition are shown in Table 5, Fig. 12, and Fig.13, respectively.

Tab. 5. The results for the Problem 12*12 by Considering machine reliability

Cells	1	2	3
Machines	5 6 8 10 12	1 7	2 3 4 9 12
Parts (Best Routing)	1(1) 3(2) 6(2) 9(2) 10(1)	12(1)	2(3) 4(1) 5(1) 7(2) 8(1) 11(1)

	B.R	1	2	2	2	1	1	3	1	1	2	1	1
	M/P	p_1	p_3	p_6	p_9	p_{10}	p_{12}	p_2	p_4	p_5	p_7	p_8	p_{11}
Cell_1	M_5	1		1		1						1	
Cell_1	M_6	1			1		1	1					
Cell_1	M_8	1	1	1		1							
Cell_1	M_{10}			1							1		1
Cell_1	M_{11}	1		1	1								
Cel 1 2	M_1									1			
Cel 1 2	M_7						1			1	1		
Cell_3	M_2				1		1	1	1	1		1	1
Cell_3	M_3	1			1						1		1
Cell_3	M_4					1			1	1	1	1	
Cell_3	M_9								1	1			1
Cell_3	M_{12}	1	1					1					1

B.R: Best Routing
M.P: Machine/Part

Fig. 12. Optimal Machine-Part Matrix by considering machine Reliability

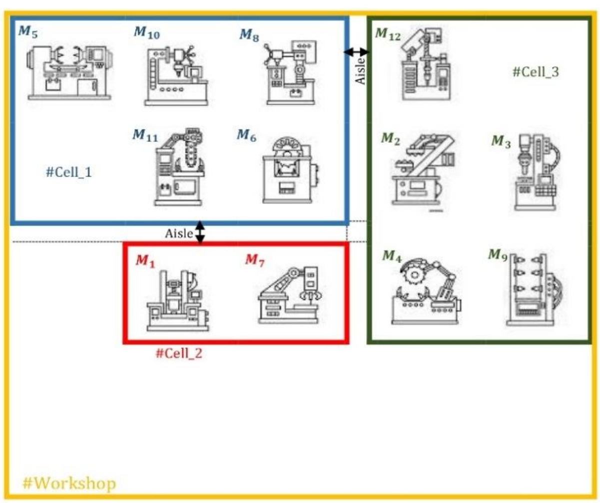

Fig. 13. Designed by the solving model by considering machine reliability

4-3. Comparison among employed metaheuristics

Here, an extensive comparison among employed metaheuristics is done. First of all, nine test problems with different complexity from small to large are conducted to evaluate the algorithms. Due to natural stochastic metaheuristics, all of them in each size test problem are run 10 times. In this regard, the best, the worst, the average, and the standard deviation of solutions among all outputted ones are analyzed. The computational time of algorithms is also noted. To check the optimal results of algorithms, an exact solver implemented by GAMS software is also selected to solve the test problems. All results are given in Table 6. Regarding the table, it should be noted that the exact solver is not capable of finding feasible solutions for the large-sized problems, even after 1 hour. However, metaheuristics have solved these examples in less than 2 minutes. The behavior of algorithms based on the computational time is given in Fig. 14. The gap of the best metaheuristic solutions from the best solution ever found by the exact solver is depicted in Fig. 15. In the end, some statistical analyses are done to show the robustness of metaheuristics in comparison with each other, as seen in Fig. 16.

As indicated from Fig. 14, the behavior of algorithms remains the same, overall. The proposed hybrid algorithm and KA show competitive results concerning this item. Generally, the best algorithm in this criterion is the KA. However, the worst behavior can be observed by RDA in most of the testes.

At first glance, Fig. 15 reveals that there is a set of similarities between the gap behaviors of metaheuristics. However, the proposed hybrid H-RDKGA shows robust behavior and reaches a near-global solution in all of the small and medium test problems.

As can be seen from Fig. 16, there is a clear difference between the performances of metaheuristics. The best algorithm is clearly the proposed hybrid metaheuristic. Conversely, the worst metaheuristic is the GA in this comparison. There is a little difference between the KA and the RDA. However, the RDA is slightly better than the KA.

In conclusion, by considering different measurements to analyze the performance of algorithms, it is evident that the proposed hybrid algorithm, called H-RDKGA, is the most powerful metaheuristic in this comparative study.

Tab. 6. Comparison of algorithms (EX=exact solver; B=best, W=worst, A=average, SD=standard deviation, and CPU=computational time based on the second)

Algorithm		Test problems								
		4*6	5*8	6*9	7*11	8*13	12*12	12*18	13*20	15*21
EX	A	24283	28641	84180	119046	218907	476036	-	-	-
	CPU	18	64	201	836	1872	3315	3600	3600	3600
GA	B	24283	28641	84180	119046	221470	476124	694902	952906	120094
	W	27925	32937	98482	139038	257709	572564	812835	1118365	138304
	A	24283	28641	85637	120903	224095	497882	706813	972491	120265
	SD	4205	4960	15740	22090	40414	101164	129847	180812	20929
	CPU	22	17	22	32	42	65	79	96	102
KA	B	24283	28641	84180	119046	219968	480789	701712	962243	121270
	W	24525	28927	85021	123807	228766	504828	736797	1010355	127333
	A	24428	28812	84684	121902	225246	495212	722763	991110	124907
	SD	194	230	677	3834	7085	19358	28254	38745	4882
	CPU	18	15	20	28	38	58	72	88	92
RDA	B	24283	28641	84180	119046	219850	476362	687953	943376	118893
	W	24524	28926	85021	122617	226445	490652	708591	971677	122459
	A	24451	28840	84768	121545	224466	486365	702399	963186	121389
	SD	160	190	560	2382	4399	9532	13767	18879	2378
	CPU	24	18	26	33	43	66	78	95	106
H-RDKGA	B	24283	28641	84180	119046	218907	476101	681004	933847	117692
	W	24283	29213	85863	121426	223285	485623	694624	952523	120045
	A	24283	28927	85021	120236	221096	480862	687814	943185	118868
	SD	0	286	841	1190	2189	4761	6810	9338	1176
	CPU	22	16	20	26	40	62	72	90	94

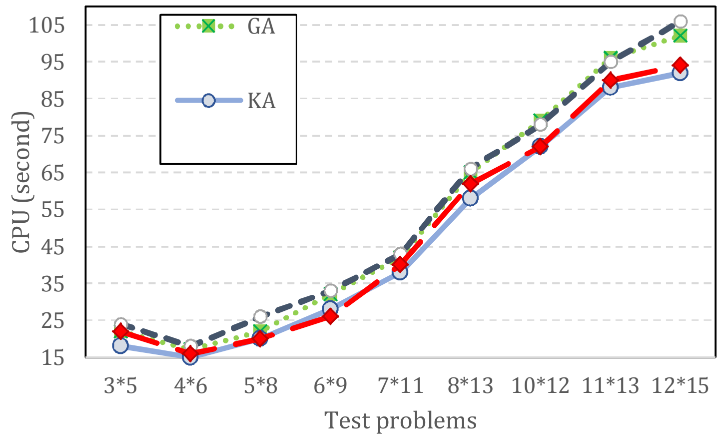

Fig. 14. Behavior of algorithms in terms of the computational time

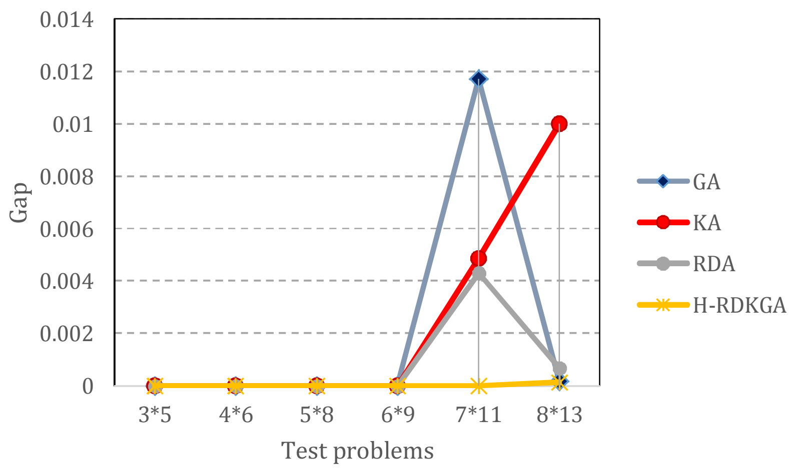

Fig. 15. Gap behavior of algorithms

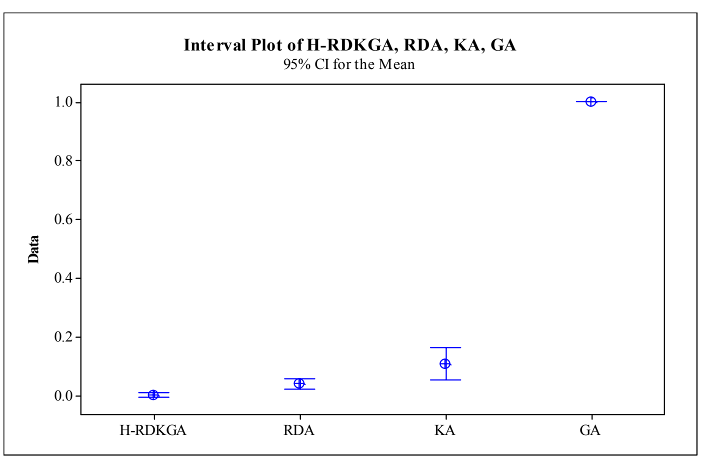

Fig. 16. Interval plots of algorithms based on the standard deviation of algorithms

5. Conclusion

A non-linear mixed-integer programming model was applied by this study to consider the simultaneous cell formation and inter/intra cell layouts in the continuous space. The purpose of the model was to determine concurrently the formation of cells and the intra- and inter-cellular layouts in a way that the total transportation cost of parts, machine breakdown cost, and the number of EEs were minimized. The excellent advantage of the presented model was to calculate the material handling costs realistically. In the proposed model, cells were configured in flexible shapes during planning horizon considering cell capacity. The problem was then simplified by linearizing the nonlinear constraints to enhance the efficiency of the problem. As a complicated optimization problem with several real-life constraints and operational decisions, which should be made in a shorter amount of time, four different metaheuristics were employed to tackle the problem. Another innovation of this study was to propose a novel hybrid metaheuristic algorithm based on the advantages of GA, KA, and RDA, simultaneously. The results showed that the proposed hybrid algorithm called H-RDKGA could find near-optimal solutions in a shorter computational time in comparison with the main original algorithms. To validate the results of metaheuristics, an exact solver by using GAMS software was utilized, which was unable to solve large instances with more than four machines for the exact solver to be impossible.

There are several recommendations for the future directions of this study. For example, incorporating production data such as dynamic conditions, setup times, and holding inventory between periods makes the model become more realistic. Another interesting idea is that the proposed problem can be investigated in uncertain situations. Hence, introducing uncertainty in a part demand and developing probabilistic or fuzzy models is recommended to cope with uncertainty. In terms of the proposed novel hybrid algorithm, more in-depth analyses by other large-scale optimization problems may be considered. New metaheuristics can be suggested to compare the results of own algorithms. Considering more evolutionary concepts such as levy fly or mutation procedures to improve the proposed hybrid algorithm is another interesting continuation of this study.

References

[1] Wang, S. and B.R. Sarker, *Locating cells with bottleneck machines in cellular manufacturing systems.* International Journal of Production Research, Vol. 40, No. 2, (2002), pp. 403-424.

[2] Ghosh, T., B. Doloi, and P.K. Dan, An Immune Genetic algorithm for inter-cell layout problem in cellular manufacturing system. Production Engineering, Vol. 10, No. 2, (2016), pp. 157-174.

[3] Alfa, A.S., M. Chen, and S.S. Heragu, *Integrating the grouping and layout problems in cellular manufacturing systems.* Computers & Industrial Engineering, Vol. 23, Nos. 1-4, (1992), pp. 55-58.

[4] Mak, K., Y. Wong, and X. Wang, *An adaptive genetic algorithm for manufacturing cell formation.* The International Journal of Advanced Manufacturing Technology, Vol. 16, No. 7, (2000), pp. 491-497.

[5] Mahdavi, I. and B. Mahadevan, *CLASS: An algorithm for cellular manufacturing system and layout design using sequence data.* Robotics and Computer-Integrated Manufacturing, Vol. 24, No. 3, (2008) pp. 488-497.

[6] Chan, H.M. and D. Milner, *Direct clustering algorithm for group formation in cellular manufacture.* Journal of Manufacturing systems, Vol. 1, No. 1, (1982), pp. 65-75.

[7] Wemmerlöv, U. and N.L. Hyer, Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management*, Vol. 6, No. 2, (1986), pp. 125-147.

[8] Fathollahi Fard, A. M., & Hajiaghaei-Keshteli, M. Integrated capacitated transportation and production scheduling problem in a fuzzy environment.

International Journal of Industrial Engineering & Production Research, Vol. 29, No. 2, (2018), pp. 197-211.

[9] Goodarzian, F., & Hosseini-Nasab, H. Applying a fuzzy multi-objective model for a production–distribution network design problem by using a novel self-adoptive evolutionary algorithm. *International Journal of Systems Science: Operations & Logistics*, (2019), pp. 1-22.

[10] Das, K., R. Lashkari, and S. Sengupta, "Reliability consideration in the design and analysis of cellular manufacturing systems." *International Journal of Production Economics*, Vol. 105, No. 1, (2007), pp. 243-262.

[11] Wu, X., et al., *A genetic algorithm for cellular manufacturing design and layout.* European Journal of Operational Research, Vol. 181, No. 1, (2007), pp. 156-167.

[12] Kannan, V.R. and S. Ghosh, Using dynamic cellular manufacturing to simplify scheduling in cell based production systems. Omega, Vol. 23, No. 4, (1995), pp. 443-452.

[13] Balakrishnan, J. and C.H. Cheng, Multi-period planning and uncertainty issues in cellular manufacturing: A review and future directions. European Journal of Operational Research, Vol. 177, No. 1, (2007), pp. 281-309.

[14] Rheault, M., J.R. Drolet, and G. Abdulnour, *Physically reconfigurable virtual cells: a dynamic model for a highly dynamic environment.* Computers & Industrial Engineering, Vol. 29, Nos. 1-4, (1995), pp. 221-225.

[15] Bulgak, A.A. and T. Bektas, Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration. European journal of operational research, Vol. 192, No. 2, (2009), pp. 414-428.

[16] Schaller, J., Designing and redesigning cellular manufacturing systems to handle demand changes. Computers & Industrial Engineering, Vol. 53, No. 3, (2007), pp. 478-490.

[17] Wicks, E.M. and R.J. Reasor, Designing cellular manufacturing systems with dynamic part populations. IIE transactions, Vol. 31, No. 1, (1999), pp. 11-20.

[18] Chen, M., A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. Annals of Operations Research, Vol. 77, (1998), pp. 109-128.

[19] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. The social engineering optimizer (SEO). *Engineering Applications of Artificial Intelligence*, Vol. 72, (2018), pp. 267-293.

[20] Fard, A. M. F., & Hajiaghaei-Keshteli, M. "Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating." In *International Conference on Industrial Engineering, IEEE.,*(2016 e) pp. 331-342.

[21] Rellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility. (2000), Virginia Tech.

[22] Safaei, N. and R. Tavakkoli-Moghaddam, Integrated multi-period cell formation and subcontracting production planning in dynamic cellular manufacturing systems. International Journal of Production Economics, Vol. 120, No. 2, (2009), pp. 301-314.

[23] Arkat, J., M.H. Farahani, and F. Ahmadizar, *Multi-objective genetic algorithm for cell formation problem*

considering cellular layout and operations scheduling. International Journal of Computer Integrated Manufacturing, Vol. 25, No. 7, (2012), pp. 625-635.

[24] Mohammadzadeh, H., Sahebjamnia, N., Fathollahi-Fard, A. M., & Hahiaghaei-Keshteli, M. New Approaches in Metaheuristics to Solve the Truck Scheduling Problem in a Cross-docking Center. *International Journal of Engineering-Transactions B: Applications*, Vol. 31, No. 8, (2018), pp. 1258-1266.

[25] Mahdavi, I., et al., An integrated model for solving cell formation and cell layout problem simultaneously considering new situations. Journal of Manufacturing Systems, Vol. 32, No. 4, (2013), pp. 655-663.

[26] Mohammadi, M. and K. Forghani, A novel approach for considering layout problem in cellular manufacturing systems with alternative processing routings and subcontracting approach. Applied Mathematical Modelling, Vol. 38, No. 14, (2014), pp. 3624-3640.

[27] Mohammadi, M. and K. Forghani, *Designing cellular manufacturing systems considering S-shaped layout.* Computers & Industrial Engineering, (2016).

[28] Safaei, N., M. Saidi-Mehrabad, and M. Jabal-Ameli, *A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system.* European Journal of Operational Research, Vol. 185, No. 2, (2008), pp. 563-592.

[29] Askin, R.G., H.M. Selim, and A.J. Vakharia, *A methodology for designing flexible cellular manufacturing systems.* IIE transactions, Vol. 29, No. 7, (1997), pp. 599-610.

[30] Safaei, N., M. Saidi-Mehrabad, and M. Babakhani, *Designing cellular manufacturing systems under dynamic and uncertain conditions.* Journal of Intelligent Manufacturing, Vol. 18, No. 3, (2007), p. 383.

[31] Tavakkoli-Moghaddam, R., N. Safaei, and F. Sassani, *A new solution for a dynamic cell formation problem with alternative routing and machine costs using simulated annealing.* Journal of the Operational Research Society, Vol. 59, No. 4, (2008), pp. 443-454.

[32] Tavakkoli-Moghaddam, R., et al., *Solving a dynamic cell formation problem using metaheuristics.* Applied Mathematics and Computation, Vol. 170, No. 2, (2005), pp. 761-780.

[33] Bajestani, M.A., et al., *A multi-objective scatter search for a dynamic cell formation problem.* Computers & operations research, Vol. 36, No. 3, (2009), pp. 777-794.

[34] Saidi-Mehrabad, M. and N. Safaei, *A new model of dynamic cell formation by a neural approach.* The International Journal of Advanced Manufacturing Technology, Vol. 33, No. 9, (2007), pp. 1001-1009.

[35] Chung, S.-H., T.-H. Wu, and C.-C. Chang, An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. Computers & Industrial Engineering, Vol. 60, No. 1, (2011), pp. 7-15.

[36] Rafiee, K., et al., A new approach towards integrated cell formation and inventory lot sizing in an unreliable cellular manufacturing system. Applied Mathematical Modelling, Vol. 35, No. 4, (2011), pp. 1810-1819.

[37] Taufik, M., & Jain, P. K. CNC-assisted selective melting for improved surface

finish of FDM parts. *Virtual and Physical Prototyping*, Vol. 11, No. 4, (2016), pp. 319-341.

[38] Jayanth, N., Senthil, P., & Prakash, C. Effect of chemical treatment on tensile strength and surface roughness of 3D-printed ABS using the FDM process. *Virtual and Physical Prototyping*, Vol. 13, No. 3, (2018), pp. 155-163.

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

[40] Alhourani, F., Cellular manufacturing system design considering machines reliability and parts alternative process routings. International Journal of Production Research, Vol. 54, No. 3, (2016), pp. 846-863.

[41] Fu, Y., Tian, G., Fathollahi-Fard, A. M., Ahmadi, A., & Zhang, C. Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. *Journal of Cleaner Production*, Vol. 226, (2019), pp. 515-525.

[42] Fathollahi Fard, A. M., & Hajiaghaei-Keshteli, M. Integrated capacitated transportation and production scheduling problem in a fuzzy environment. *International Journal of Industrial Engineering & Production Research*, Vol. 29, No. 2, (2018), pp. 197-211.

[43] Shirzad, S., Tavakkoli-Moghaddam, R., Kia, R., & Mohammadi, M., A multi-objective imperialist competitive algorithm for integrating intra cell layout and processing route reliability in a cellular manufacturing system. *International Journal of Computer Integrated Manufacturing,* Vol. 66, (2016), pp. 184-206.

[44] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Mirjalili, S. A set of efficient heuristics for a home healthcare problem. *Neural Computing and Applications*, (2019), pp. 1-21.

[45] Fattahi, P., Keneshloo, S., & Daneshamooz, F. A Hybrid Genetic Algorithm and Parallel Variable Neighborhood Search for Jobshop Scheduling With an Assembly Stage. *International Journal of Industrial Engineering & Production Research*, Vol. 30, No. 1, (2019), pp. 25-37.

[46] Fathollahi-Fard, A. M., Govindan, K., Hajiaghaei-Keshteli, M., & Ahmadi, A. A green home health care supply chain: New modified simulated annealing algorithms. *Journal of Cleaner Production*, 118200, (2019).

[47] Paydar, M. M., & Saidi-Mehrabad, M. A hybrid genetic algorithm for dynamic virtual cellular manufacturing with supplier selection. *The International Journal of Advanced Manufacturing Technology*, Vol. 92, Nos. 5-8, (2017), pp. 3001-3017.

[48] Behnia, B., Mahdavi, I., Shirazi, B., & Paydar, M. M. A bi-level mathematical programming for cell formation problem considering workers' interest. *International Journal of Industrial Engineering & Production Research*, Vol. 28, No. 3, (2017), pp. 267-277.

[49] Mahdavi, I., Bootaki, B., & Paydar, M. M. Manufacturing cell configuration considering worker interest concept applying a bi-objective programming approach. *International Journal of Industrial Engineering*, Vol. 25, No. 1, (2014), pp. 41-53.

[50] Gholipoor, A., Paydar, M. M., & Safaei, A. S. A faucet closed-loop supply chain network design considering used faucet exchange plan. *Journal of Cleaner Production*, Vol. 235, (2019), pp. 503-518.

[51] Shamsi, F., Mahdavi, I., & Paydar, M. M. A possibilistic programming approach to analyze a closed-loop polyethylene tanks supply chain based on decision tree and discounted cash flow. *International Journal of Management Science and Engineering Management*, (2019), pp. 1-16.

Follow This Article at The Following Site:

Golmohammadi A, Honarvar M, Guangdong G, Hosseini-Nasab H. A New Mathematical Model for Integration of Cell Formation with Machine Layout and Cell Layout by Considering Alternative Process Routing Reliability; A Novel Hybrid Metaheuristic. IJIEPR. 2019; 30 (4) :405-427
URL: http://ijiepr.iust.ac.ir/article-1-911-en.html