

Research Article

Soft Computing Methodology to Optimize the Integrated Dynamic Models of Cellular Manufacturing Systems in a Robust Environment

Amir-Mohammad Golmohammadi ¹, Hasan Rasay ², Zaynab Akhoundpour Amiri,³
Maryam Solgi,⁴ and Negar Balajeh⁵

¹Department of Industrial Engineering, Arak University, Arak, Iran

²Department of Industrial Engineering, Kermanshah University of Technology, Kermanshah, Iran

³Department of Industrial Engineering, System Management and Productivity, University of Alghadir, Tabriz, Iran

⁴Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran

⁵Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Correspondence should be addressed to Amir-Mohammad Golmohammadi; a-golmohammadi@araku.ac.ir

Received 17 August 2021; Revised 12 September 2021; Accepted 20 October 2021; Published 11 November 2021

Academic Editor: Alireza Goli

Copyright © 2021 Amir-Mohammad Golmohammadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine learning, neural networks, and metaheuristic algorithms are relatively new subjects, closely related to each other: learning is somehow an intrinsic part of all of them. On the other hand, cell formation (CF) and facility layout design are the two fundamental steps in the CMS implementation. To get a successful CMS design, addressing the interrelated decisions simultaneously is important. In this article, a new nonlinear mixed-integer programming model is presented which comprehensively considers solving the integrated dynamic cell formation and inter/intracell layouts in continuous space. In the proposed model, cells are configured in flexible shapes during the planning horizon considering cell capacity in each period. This study considers the exact information about facility layout design and material handling cost. The proposed model is an NP-hard mixed-integer nonlinear programming model. To optimize the proposed problem, first, three metaheuristic algorithms, that is, Genetic Algorithm (GA), Keshtel Algorithm (KA), and Red Deer Algorithm (RDA), are employed. Then, to further improve the quality of the solutions, using machine learning approaches and combining the results of the aforementioned algorithms, a new metaheuristic algorithm is proposed. Numerical examples, sensitivity analyses, and comparisons of the performances of the algorithms are conducted.

1. Introduction

Facility layout (FL) considers the layout of machines within cells (Intracell layout) and the layout of cells (Intercell Layout) on the shop floor which can be counted as an essential element to plan a CMS. The cutdown in material handling cost, work-in-process, and throughput rate is the result of an efficient FL [1]. The system's performance would have the capacity to be boosted by designing a capable layout and the expense of the production also would fall around 40% to 50% on average [2]. In the FL problem, the actual

parameters in the flow of some objectives might be ignored. The objectives which might diminish the intercellular circulation but none of them would certainly cause the charge of material handling to be minimized could be the minimization of the number of exceptional elements (EEs) or another common one can be referred to as the cost reduction of intercell movement. Although layout design has been somehow neglected in the CMS as many of the related researches have only investigated the CF, the vital point in this design could be using the FL problem [3, 4]. As stated, the decisions that are made in the FL and CF problem are

interrelated, and tellingly to have a favored CMS design inscribing these two concurrently is of high importance [5]. Nevertheless, it has been asserted a complicated issue to make any of these kinds of decisions [6, 7]. Therefore the simultaneous addressing of these decisions is a difficult issue.

In most studies, some of the mentioned decisions have been handled and sometimes all of them consecutively [8]. On the other hand, most approaches in the area of facility layout and CF problem, for simplicity, usually consider minimizing the number of intercell movements or intracell movements or both [9]. To make a minimum material handling cost in FL design, it is an important matter to inspect the exact details of this design like the concept of distance. On the other hand, these types of approaches might make some illusive presumptions being machine location or fixed cells in the FL problem which might lead to an incompetent outcome. Also, in earlier investigations, the machines existing in a cell space were placed only in one type of location which was a line-formed one. There is another way to locate the machines other than line-form as well which is a U-form but the problem would be the expenses add up to the system.

Roughly speaking, many models in the field of CMS can be classified as NP-hard problems. Thus, employing and proposing exact mathematical approaches to tackle the models of CMS is usually ineffective. To this end, different heuristic, metaheuristic, and machine learning approaches have been applied, which can effectively handle the NP-hard models of CMS.

Machine learning, neural networks, and metaheuristic algorithms are relatively new subjects employed broadly in different fields of industrial engineering and management studies. They are also closely related to each other: learning is somehow an intrinsic part of all of them [10, 11]. Computer science, probability and statistic rules, and information and decision theory play a major role in the development of machine learning and other algorithms of metaheuristics. Machine learning methods find applications in different fields of industrial engineering, vehicle routing problem, and lot sizing, and maintenance optimization models are among others [12].

As the proposed mixed-integer nonlinear model in this paper is NP-hard, four metaheuristic algorithms are employed to tackle the problem. In the first step, the Genetic Algorithm (GA), Keshtel Algorithm (KA), and Red Deer Algorithm (RDA) are designed to optimize the model. To further improve the solutions of the algorithms, using machine learning approaches, a novel metaheuristic algorithm, which benefits from the merits of the aforementioned technique, is proposed. Accordingly, using soft computing methodology to optimize the integrated dynamic cell formation problem in a robust environment is the main contribution of the current study.

The rest of the paper is organized as follows: In Section 2, the mathematical model for the problem is presented. Section 3 presents the metaheuristic algorithms for

optimizing the model. Section 4 discusses how the algorithms are tuned and reports the results of computations and calculations. Finally, Section 5 concludes the paper.

2. Proposed Mathematical Model

2.1. Notations

2.1.1. Sets

$i, i' = \{1, 2, \dots, m\}$: machines group symbol

$j = \{1, 2, \dots, n\}$: parts symbol

$l, k, k' = \{1, 2, \dots, c\}$: cells group symbol

$h = \{1, 2, \dots, H\}$: periods group symbol

2.1.2. Parameters

D_{jh} : parts demand for type j in period h

B_j : batch size for parts transportation type j

C_{intra}^j : the expense of intracell material handling for the transportation of part j per unit distance (\$/unit)

C_{inter}^j : the expense of intercell material handling for the transportation of part j per unit distance (\$/unit)

C_i : the expense of relocating machine i (\$/unit)

R_{ij} : the number of operations done on part j using machine i

E : shop floor horizontal length (the length of the shop floor)

F : job shop vertical length (the width of the shop floor)

SP : the pairs group (i, j) such that $a_{ij} \geq 1$ (the group of elements that are not zero of a part-machine matrix)

NM : every cell's maximum figure of relocated machines per period

α_j : the expense coefficient (or penalty) caused by existence of any special part type j per period

N : an appropriate big positive figure

A_{kl}, B_{kl} : a random variable and a zero

$A_{i'h}, B_{i'h}$: a random variable and a zero

$f_{i'h}^j$: trips number for changing the location of part type j between machines i and i' in period h :

$$f_{i'h}^j = \begin{cases} \left\lceil \frac{D_{jh}}{B_j} \right\rceil, & \text{if } R_{i'j} - R_{ij} = 1, \\ 0, & \text{if } R_{i'j} - R_{ij} \neq 1. \end{cases} \quad (1)$$

2.1.3. Decision Variables

$$X_{ikh} = \begin{cases} 1, & \text{If machine } i \text{ is allocated to cell } k \text{ in period } h, \\ 0, & \text{Otherwise} \end{cases}$$

$$Y_{jkh} = \begin{cases} 1, & \text{If part } j \text{ is allocated to cell } k \text{ in period } h, \\ 0, & \text{Otherwise} \end{cases}$$

$$Z_{ih} = \begin{cases} 1, & \text{If machine } i \text{ displaces through periods } h \\ \text{and } (h + 1), 0, & \text{Otherwise} \end{cases}$$

$$U_{ijkh} = \begin{cases} 1, & \text{if } Y_{jkh} = 0 \text{ and } X_{ikh} = 1, \\ 0, & \text{Otherwise} \end{cases}$$

$$V_{ijkh} = \begin{cases} 1, & \text{if } Y_{jkh} = 1 \text{ and } X_{ikh} = 0, \\ 0, & \text{Otherwise} \end{cases}$$

x_{ih} : the center of machine i in period h in horizontal coordinate

y_{ih} : the center of machine i in period h in vertical coordinate

p_{kh}^1 : the left side of cell k in period h in horizontal coordinate

p_{kh}^2 : the right side of cell k in period h in horizontal coordinate

q_{kh}^1 : the bottom side of cell k in period h in vertical coordinate

q_{kh}^2 : the top side of cell k in period h in vertical coordinate

Thus, the expense of displacing part j between machines i and i' in period h concerning the movement of the intercell or intracell could be determined as follows.

If $X_{ikh}, X_{i'kh} > 0$ this cost is equal to the following equation:

$$C_{i'i'h}^j = (|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|)C_{\text{intra}}^j. \quad (2)$$

If $X_{ikh}X_{i'kh} = 0$ and $X_{ikh}X_{i'k'h} > 0$ this cost is equal to the following equation:

$$C_{i'i'h}^j = (|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|)C_{\text{inter}}^j. \quad (3)$$

The way cells are configured, the machines plot inside them, and also their layout on the shop floor are the goals of this model to be defined simultaneously in kinetic situations somehow that some expenses such as cells redesigning, parts total transportation cost, and EES number are reduced. In the given model, the job shop figure is considered for the intracellular layout. Some mixed-integer nonlinear programming models are discussed below along with multiple presumptions, parameters, and decision variables.

2.2. Model Assumption. The following presumptions are taken into account to simulate the model:

- (i) In every period, the flow between machines is defined. The demand for parts, operational paths of parts, and also the parts transportation batch size are how this figure is gained from.
- (ii) If each product's batch size is in all periods constant and determined, the parts can move within the batches. For both inter- and intracell displacements, the parts batch largeness is considered the same.
- (iii) By applying rectilinear distance, the expense of material handling is assessed regarding the center-to-center distance between machines.
- (iv) The expense of material handling for inter- and intracell relocations for machines and parts pertains to the distance traveled.
- (v) The unit expense of intercell and intracell relocations for every part type is predestined and stays constant in the planning horizon.
- (vi) During the periods, the unit expense of machine displacement remains the same and is predestined for every kind of machine. The expense contains the opening, transferring, and resetting of the machine.
- (vii) In every period, by using the expected workload in every cell, the number of cells to be shaped can be ascertained beforehand. Nevertheless, during the planning horizon, since the cells are malleable, they can be shaped easily; hence, their configuration would not be predetermined.
- (viii) Only one number of each machine kind exists.
- (ix) During the planning horizon, the maximum capacity of cells is determined and stays constant.
- (x) The machines are supposed to have a unit dimension, since they are squares of equal area. No extra inventory should exist between the periods; each demand must be supplied in the related period and no delayed orders are permitted.
- (xi) 100% is assumed for machines and production efficiency.

2.3. Mathematical Formulation. Concerning input parameters and variables, the presented nonlinear model for this problem is as follows:

$$\text{minimize } \sum_{h=1}^H \sum_{j=1}^n \sum_{i=1}^m \sum_{i'=1}^m f_{i'i'h}^j C_{i'i'h}^j + \sum_{h=2}^H \sum_{i=1}^m C_i Z_{ih} + \sum_{h=1}^H \sum_{k=1}^C \sum_{(i,j) \in sp} \alpha_j \cdot \frac{(U_{ijkh} + V_{ijkh})}{2}. \quad (4)$$

It is subject to

$$\sum_{k=1}^C X_{ikh} = 1, \quad i = 1, 2, \dots, m, \forall h, \quad (5)$$

$$\sum_{k=1}^C Y_{jkh} = 1, \quad j = 1, 2, \dots, n, \forall h, \quad (6)$$

$$1 \leq \sum_{i=1}^m X_{ikh} \leq NM, \quad k = 1, 2, \dots, C, \forall h, \quad (7)$$

$$NZ_{ih} \geq |x_{ih} - x_{i(h+1)}| + |y_{ih} - y_{i(h+1)}|, \quad \forall i, h < H, \quad (8)$$

$$|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}| \geq 1, \quad (9)$$

$$\begin{cases} x_{ih} \geq p_{kh}^1 - N(1 - X_{ikh}), \\ x_{ih} \leq p_{kh}^2 + N(1 - X_{ikh}), \\ y_{ih} \geq q_{kh}^1 - N(1 - X_{ikh}), \\ y_{ih} \leq q_{kh}^2 + N(1 - X_{ikh}) \end{cases}, \quad \forall i, k, h, \quad (10)$$

$$\begin{cases} p_{kh}^1 \geq 0, \\ q_{kh}^1 \geq 0, \\ p_{kh}^2 \leq E, \\ q_{kh}^2 \leq F \end{cases}, \quad \forall k, h, \quad (11)$$

$$\begin{cases} p_{kh}^1 - p_{lh}^2 + NA_{kl} + NB_{kl} \geq 0, \\ p_{kh}^2 - p_{lh}^1 - NA_{kl} - N(1 - B_{kl}) \leq 0, \\ q_{kh}^1 - q_{lh}^2 + N(1 - A_{kl}) + NB_{kl} \geq 0, \\ q_{kh}^2 - q_{lh}^1 - N(1 - A_{kl}) - N(1 - B_{kl}) \leq 0, \\ 0 \leq k < l \leq C, \end{cases} \quad (12)$$

$$\begin{cases} x_{ih} - x_{i'h} + NA_{ii'h} + NB_{ii'h} \geq 1, \\ x_{i'h} - x_{ih} - NA_{ii'h} - N(1 - B_{ii'h}) \geq 1, \\ y_{ih} - y_{i'h} + N(1 - A_{ii'h}) + NB_{ii'h} \geq 1, \\ y_{i'h} - y_{ih} - N(1 - A_{ii'h}) - N(1 - B_{ii'h}) \geq 1 \end{cases}, \quad \forall 1 \leq i < i' \leq M. \quad (13)$$

The intracellular and intercellular material transferring costs are represented by the first term of the objective

function. The expense of cell reshaping which might alter from period to period is defined in the following term. The decreasing number of exceptional parts is in connection with the third term. The double calculation of decision variables when they are 1 in this relationship is the reason behind the coefficient of 1/2. The first set of constraints (Equation (5)) guarantees that every machine is allocated to just one cell. Every part that has been allocated to one part family only is ensured by the second constraint (Equation (6)). Constraint (7) is the limitation for the number of machines in one cell. Variable Z_{ih} equals 1 when machine type i is displaced during periods h and $(1+h)$, which is ensured by the fourth constraint (Equation (8)). The fifth constraint (Equation (9)) which is replaced with Equation (13) prevents machines from being overlapped. As mentioned, the machines are considered squares with a unit dimension. Each machine that has to displace in space of its corresponding cell is indicated in the set of relationships (10). The next constraint (Equation (11)) is developed to control the cells which are in space of the job shop. Preventing cells from being overlapped is shown in the set of relationships in (12).

2.4. Proposed Robust Model. Robust optimization is used when there is uncertainty in the parameters. In this case, with a slight change in the value of one of the parameters, the optimality and justification of the answer may be compromised. Therefore, to control the situation, it is necessary to use an optimization model that, considering the existing uncertainties, obtains an optimal answer that remains an optimal and justified answer to the problem under investigation in the face of changes in uncertain parameters.

In this paper, to face the uncertainty of the parameters, set-induced robust optimization is used. In this optimization, it is assumed that uncertain data belongs to an uncertainty set, and the aim is to choose the best solution among those "immunized" against data uncertainty, that is, candidate solutions that remain feasible for all realizations of the data from the uncertainty set.

According to the above, a symmetric interval for the range of parameter changes is considered as follows.

The objective coefficients and the constraint coefficients possibly change with an unknown distribution but they are symmetrical and independent in the following intervals:

$$\tilde{C}_{\text{intra}}^j \in \left(C_{\text{intra}}^j - \tilde{C}_{\text{intra}}^j, C_{\text{intra}}^j + \tilde{C}_{\text{intra}}^j \right),$$

$$\tilde{C}_{\text{inter}}^j \in \left(C_{\text{inter}}^j - \tilde{C}_{\text{inter}}^j, C_{\text{inter}}^j + \tilde{C}_{\text{inter}}^j \right),$$

$$\tilde{C}_i \in \left(C_i - \tilde{C}_i, C_i + \tilde{C}_i \right),$$

$$\tilde{D}_{jh} \in \left(D_{jh} - \tilde{D}_{jh}, D_{jh} + \tilde{D}_{jh} \right),$$

$$\begin{aligned}
 \tilde{C}_{ii'h}^j &\in \left((|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|) \left(C_{\text{intra}}^j - \widehat{C}_{\text{intra}}^j (|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|) \left(C_{\text{intra}}^j + \widehat{C}_{\text{intra}}^j \right) \right) \right), \\
 \tilde{C}_{ii'h}^j &\in \left((|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|) \left(C_{\text{inter}}^j - \widehat{C}_{\text{inter}}^j (|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}|) \left(C_{\text{inter}}^j + \widehat{C}_{\text{inter}}^j \right) \right) \right), \\
 \tilde{f}_{ii'h}^j &\in \begin{cases} \left[\frac{D_{jh} - \widehat{D}_{jh}}{B_j}, \frac{D_{jh} + \widehat{D}_{jh}}{B_j} \right], & \text{if } R_{i'j} - R_{ij} = 1, \\ 0, & \text{if } R_{i'j} - R_{ij} \neq 1. \end{cases}
 \end{aligned} \tag{14}$$

For each parameter \tilde{a}_{ij} which is subject to uncertainty, a_{ij} represents the nominal value of the parameters and \widehat{a}_{ij} represents constant perturbation (which are positive).

In this paper, a robust approach developed by Bertsimas and Sim is used to face the uncertainty of the parameters.

Therefore, the robust counterpart of the original problem is obtained by replacing the original constraint with its robust counterpart constraint:

$$\begin{aligned}
 F \geq & \sum_{h=1}^H \sum_{j=1}^n \sum_{i=1}^m \sum_{i'=1}^m f_{ii'h}^j C_{ii'h}^j + \sum_{h=2}^H \sum_{i=1}^m C_i Z_{ih} + \Gamma_0 Z_0 + \sum_{h=1}^H \sum_{j=1}^n \sum_{i=1}^m \sum_{i'=1}^m q_{ii'h}^j + \sum_{h=2}^H \sum_{i=1}^m s_{ih} \\
 & + \sum_{h=1}^H \sum_{k=1}^C \sum_{(i,j) \in sp} \alpha_j \frac{(U_{ijkh} + V_{ijkh})}{2},
 \end{aligned}$$

$$Z_0 + q_{ii'h}^j \geq \frac{\widehat{D}_{jh}}{B_j} (|xx_{ih} - xx_{i'h}| + |yy_{ih} - yy_{i'h}|) \widehat{C}_{\text{intra}}^j, \quad \forall i, h, j, i',$$

$$Z_0 + q_{ii'h}^j \geq \frac{\widehat{D}_{jh}}{B_j} (|xx_{ih} - xx_{i'h}| + |yy_{ih} - yy_{i'h}|) \widehat{C}_{\text{inter}}^j, \quad \forall i, h, j, i',$$

$$Z_0 + s_{ih} \geq \widehat{C}_i Z Z_{ih}, \quad \forall i, h,$$

$$-xx_{ih} \leq x_{ih} \leq xx_{ih}, \quad \forall i, h,$$

$$-xx_{i'h} \leq x_{i'h} \leq xx_{i'h}, \quad \forall i', h,$$

$$-yy_{ih} \leq y_{ih} \leq yy_{ih}, \quad \forall i, h,$$

$$-yy_{i'h} \leq y_{i'h} \leq yy_{i'h}, \quad \forall i', h,$$

$$-ZZ_{ih} \leq Z_{ih} \leq ZZ_{ih}, \quad \forall i, h,$$

$$\sum_{k=1}^C X_{ikh} = 1, \quad i = 1, 2, \dots, m, \forall h,$$

$$\sum_{k=1}^C Y_{jkh} = 1, \quad j = 1, 2, \dots, n, \forall h,$$

$$1 \leq \sum_{i=1}^m X_{ikh} \leq NM, \quad k = 1, 2, \dots, C, \forall h,$$

$$NZ_{ih} \geq |x_{ih} - x_{i(h+1)}| + |y_{ih} - y_{i(h+1)}|, \quad \forall i, h < H,$$

$$\begin{cases}
|x_{ih} - x_{i'h}| + |y_{ih} - y_{i'h}| \geq 1, \\
\begin{cases}
x_{ih} \geq p_{kh}^1 - N(1 - X_{ikh}) \\
x_{ih} \leq p_{kh}^2 + N(1 - X_{ikh}) \\
y_{ih} \geq q_{kh}^1 - N(1 - X_{ikh}) \\
y_{ih} \leq q_{kh}^2 + N(1 - X_{ikh})
\end{cases}, \forall i, k, h, \\
\begin{cases}
p_{kh}^1 \geq 0 \\
q_{kh}^1 \geq 0 \\
p_{kh}^2 \leq E \\
q_{kh}^2 \leq F
\end{cases}, \forall k, h, \\
\begin{cases}
p_{kh}^1 - p_{lh}^2 + NA_{kl} + NB_{kl} \geq 0, \\
p_{kh}^2 - p_{lh}^1 - NA_{kl} - N(1 - B_{kl}) \leq 0, \\
q_{kh}^1 - q_{lh}^2 + N(1 - A_{kl}) + NB_{kl} \geq 0, \\
q_{kh}^2 - q_{lh}^1 - N(1 - A_{kl}) - N(1 - B_{kl}) \leq 0, \\
0 \leq k < l \leq C,
\end{cases} \\
\begin{cases}
x_{ih} - x_{i'h} + NA_{ii'h} + NB_{ii'h} \geq 1 \\
x_{i'h} - x_{ih} - NA_{ii'h} - N(1 - B_{ii'h}) \geq 1 \\
y_{ih} - y_{i'h} + N(1 - A_{ii'h}) + NB_{ii'h} \geq 1 \\
y_{i'h} - y_{ih} - N(1 - A_{ii'h}) - N(1 - B_{ii'h}) \geq 1
\end{cases}, \forall 1 \leq i < i' \leq M, \\
xx_{ih}, xx_{i'h}, yy_{ih}, yy_{i'h}, q_{ii'h}^j, s_{ih} \geq 0,
\end{cases} \quad (15)$$

where Γ is the adjustable parameter controlling the size of the uncertainty set.

3. Proposed Solution Algorithm

The literature approved that the CMS models are classified as NP-hard problems [13–19]. The high complexity of CMS in large-scale instances motivates several researchers to propose novel metaheuristics [20]. This study in addition to the Genetic Algorithm (GA) applies two recent nature-inspired metaheuristics: Keshtel Algorithm (KA) and Red Deer Algorithm (RDA). To improve the benefits of these recent and old metaheuristics, a novel hybrid algorithm is also developed to better address the proposed problem and to provide a comparison among these algorithms based on the solution time and quality. Next, the encoding plan to run the initial population of the metaheuristics is explained. Then, the proposed optimizers are introduced.

3.1. Solution Representation. Since all stochastic optimizers such as GA, KA, and RDA use a continuous search space, an encoding plan to transform it into a discrete area to confirm that the algorithm can address the constraints of our model is highly needed [21–23]. A general view of the encoding scheme is depicted in Figure 1. Then, the assignment of machines and details of the cell manufacturing to generate a feasible solution are given in Figure 2.

Finally, to compute the objective functions (the fitness functions), a matrix with H rows and N columns is generated

regarding the matrixes in Figure 2. The structure of the final matrix for the alignment is given in Figure 3.

3.2. Keshtel Algorithm (KA). Swarm intelligence is one of the main inspirations for metaheuristics. Considering that the swarm behavior of birds, bees, and ants is a hot topic for metaheuristics studies, the Keshtel Algorithm (KA) is another swarm intelligence-based metaheuristic. This algorithm was firstly developed by Hajiaghaei-Keshteli and Aminnayeri [24]. With regard to inspiration from a dabbling duck, this metaheuristic studies the feeding behavior of *Anas* ducks.

According to behaviors of this type of birds, they live in Asia and normally in northern countries like Russia, Azerbaijan, and Iran. They always migrate from northern lands in Russia to the northern parts of Iran and Azerbaijan near the Caspian sea. The Keshtels have amazing behavior in their feeding. When they find a source of food in the lake, other Keshtels approach this lucky Keshtel who firstly found a portion of good food and they swirl together in a circle. Other Keshtels who cannot find a good source of food move to other parts of the lake or fly to another lake [24].

To model these behaviors of Keshtels, Hajiaghaei-Keshteli and Aminnayeri [24] proposed a nature-inspired metaheuristic for solving optimization problems. They generated the initial Keshtels as a set of random solutions in the lake. They divided this population into three groups (i.e., N_1 , N_2 , and N_3) with regard to the fitness or the cost of the objective function. N_1 is the group of the lucky Keshtels who

$$[[Z] [X] [Y]]$$

FIGURE 1: General view of the solution representation.

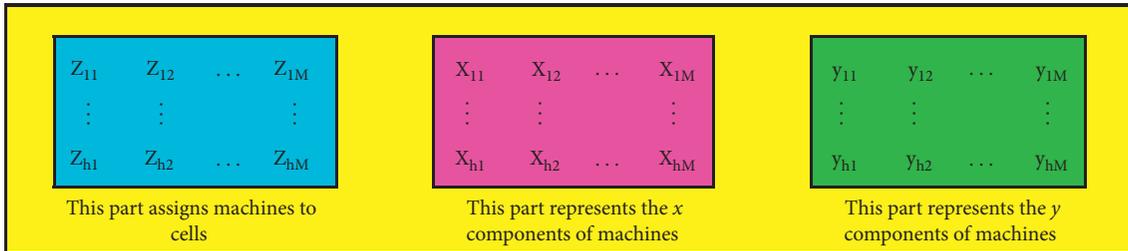


FIGURE 2: Generation of a feasible solution.

$$\begin{bmatrix} z'_{11} & z'_{12} & \dots & z'_{1N} \\ \vdots & \vdots & \dots & \vdots \\ z'_{h1} & z'_{h1} & \dots & z'_{hN} \end{bmatrix}$$

FIGURE 3: Final matrix to compute the fitness function.

have found a good source of food in the lake. N_2 moves fast between the lucky Keshtels to search for source of food . The best source of food is the global solution and each Keshtel finding it is the best solution in all iterations. Finally, the last group, that is, N_3 population, is generated randomly in each iteration. They are new Keshtels that may land in the lake.

As a metaheuristic, it is very important to find an interaction between two main search phases, that is, intensification and diversification. In this metaheuristic, this classification of three groups is very useful to explore the new search areas. The first group (N_1) does the exploitation or intensification phase. Other groups help the algorithm to perform the diversification phase. Most importantly, the N_3 group finds a way for the algorithm to escape from the local solutions. To the best of our knowledge, no paper contributes the KA in this research area. To have a conclusion about the steps of KA, a pseudocode is presented in Figure 4.

3.3. Red Deer Algorithm (RDA). Evolutionary algorithms are another well-known classification of metaheuristics. These algorithms are also nature-inspired. However, from the current to the next generation, only a group of animals who are probably stronger than other ones will remain and other agents will be removed. As another evolutionary metaheuristic, Fathollahi-Fard et al. [25] recently proposed the Red Deer Algorithm (RDA) inspired by an amazing behavior of males and females in a breeding season.

This algorithm studies the behavior of red deer with regard to roaring, fighting, and mating behaviors. These animals are naturally living in the British Isles, mainly in Scotland. In this regard, the scientists called them Scottish Red Deer (*Cervus elaphus scoticus*). In a breeding season, the males that are also known as stags roar loudly and repeatedly to attract the females, so-called hinds. Based on this feature of the males, the hinds select their preferable stag, and he will

create his territory and harem. A harem is a group of hinds and a commander as the head of this group manages and controls them. The fighting action always exists among males. Stags and commanders fight, and the winner will achieve the territory and harem. This competition among males is the main activity. The last part of this season is the mating behaviors among males and hinds and, as a result, the new red deer will have born for the next breeding season. Among all roaring, fighting, and mating processes, the evolutionary concept confirms that only the strongest will always remain in nature and this rule has existed among red deer.

Fathollahi-Fard et al. [25] modeled these facts as another evolutionary algorithm. They generated the first population of red deer as the random solutions. This population is divided into males and hinds. Then, males roar, and, based on their power, a group of them will be selected as the commanders and the others are stags. Next, a fight between commanders and stags occurs. After that, for each commander, a harem will be generated by some random hinds. The number of hinds in a harem is directly related to the power of the commander. After that, the commander has this ability to mate with a number of his hinds in the harem and a few hinds in another harem. The stag that does not have a chance to be a commander can mate with one hind that is closest to him geographically. After the mating, offspring is created for each mating. Finally, for the next generation, the males will be selected as the best solutions among all available solutions and the hinds will be selected by an evolutionary mechanism like the roulette wheel selection method.

With these features, the authors developed an interesting and successful metaheuristic and called it RDA. To have a brief illustration of RDA, its pseudocode is presented in Figure 5.

3.4. Proposed Novel Hybrid Metaheuristic (H-RDKGA). Based on the aforementioned description, it is approved that the KA uses a high exploitive behavior. The RDA is good at the exploration phase. The GA has also a classical crossover operator to do the explorative behavior. The proposed novel hybrid metaheuristic called H-RDKAGA uses the aforementioned benefits.

```

Initialize Keshtels population.
Calculate the fitness and sort them into 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 the objective functions has been improved) of this new position is better
than prior
                Update this lucky Keshtel.
                break
            endif
             $S=S+1$ 
        end while
    endfor
    for each  $N_2$ 
        Move the Keshtel between the two Keshtels, randomly.
    endfor
    for each  $N_3$ 
        Create a random solution.
    endfor
    Merge the  $N_1$ ,  $N_2$ , and  $N_3$ .
    Sort the Keshtels and form  $N_1$ ,  $N_2$ , and  $N_3$  for the next iteration.
    Update the  $X^*$  if there is a better solution.
     $t=t+1$ ;
end while
return  $X^*$ 

```

FIGURE 4: The pseudocode of KA.

In the proposed hybrid algorithm, the RDA acts as the main loop and two other algorithms improve the subloop of this algorithm. This hybrid metaheuristic uses the swirling process instead of roaring and fighting operators in the RDA. In this regard, each male performs the swirling process with his closest neighbor. The proposed hybrid algorithm also considers the crossover of the GA instead of the mating operator. Other steps are similar to the main RDA. Finally, the pseudocode of H-RDKGA is provided in Figure 6. The evaluation metrics for the performance of the algorithms are shown in Table 1.

Then, the acquired results for every problem are converted to the Relative Percentage Deviation (RPD) computed by

$$RPD = \frac{|Alg_{sol} - Best_{sol}|}{Best_{sol}}, \quad (16)$$

where Alg_{sol} is the output of the algorithm and $Best_{sol}$ is the best value ever found in the problem size. It should be noted that the lower value for the RPD is preferred.

4. Tuning of Algorithms and Comparison Studies

In this section, first, using the design of the experimental approach, the parameters of the aforementioned algorithms

are tuned. Then, a comprehensive study is done to evaluate the performances of the algorithms.

4.1. Tuning of Metaheuristics. As all metaheuristics have some controlling parameters, tuning is needed satisfactorily. Here, based on the concept of the Design of Experiment (DOE), all algorithms have been calibrated. This method can analyze the impact of different candidate values on the parameters of the algorithms and evaluate the behavior of the algorithms. Without a good calibration of the parameters, the behavior of the metaheuristics is not reliable.

To do the tuning, the parameters of the given metaheuristics are considered. With regard to the DOE method, a full factorial method to analyze all possible experiments with regard to the levels is done. The levels and tuned values for each parameter are given in Table 2. It should be noted that all candidate level values are taken from similar studies in the literature [18, 21, 26].

4.2. Comparison among Employed Metaheuristics. To do the comparison among the employed metaheuristics, nine test studies are benchmarked from the literature [15]. These tests are selected from small-scale to large-scale instances. As the model of our work is novel and differs from previous works, no comparison between our results and previous studies is

```

Initialize the Red Deers population.
Calculate the fitness and 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 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 better solution.
   $t = t + 1$ ;
end while
return X*

```

FIGURE 5: The pseudocode of RDA.

```

Initialize the Red Deers population.
Calculate the fitness and 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 prior
      Update this lucky male.
      break
    end if
    S=S+1
  end while
  end for
  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 the X* if there is better solution.
   $t = t + 1$ ;
end while
return X*

```

FIGURE 6: The pseudocode of H-RDKGA.

TABLE 1: Evaluation metrics to the performance of the algorithms (i.e., DM, SNS, DEA, and POD).

Instances		DM			
$M \times P$	GA	KA	RDA	H-RDKGA	
3 * 5	15861	14389	16452	14015	
4 * 6	18753	17275	19743	16527	
5 * 8	20213	19833	21872	18817	
6 * 9	22916	21806	33112	20763	
7 * 11	25817	24319	39671	23917	
8 * 13	27918	26518	43749	24008	
10 * 12	32650	31997	55761	29879	
11 * 13	38650	36521	57144	34699	
12 * 15	47840	47003	60195	45810	
Instances		SNS			
$M \times P$	GA	KA	RDA	H-RDKGA	
3 * 5	2498	2267	1748	2699	
4 * 6	6122	7210	5426	7495	
5 * 8	7445	7296	6948	8155	
6 * 9	3485	3105	2915	4039	
7 * 11	2143	1834	7501	2867	
8 * 13	1077	1282	675	2049	
10 * 12	5482	4912	4466	4288	
11 * 13	6388	5187	5514	6382	
12 * 15	6237	5853	6432	7528	
Instances		DEA			
$M \times P$	GA	KA	RDA	H-RDKGA	
3 * 5	0.18	0.16	0.12	0.15	
4 * 6	0.20	0.12	0.18	0.12	
5 * 8	0.24	0.22	0.26	0.18	
6 * 9	0.28	0.14	0.22	0.14	
7 * 11	0.16	0.26	0.18	0.16	
8 * 13	0.24	0.12	0.12	0.19	
10 * 12	0.18	0.14	0.20	0.22	
11 * 13	0.26	0.18	0.14	0.18	
12 * 15	0.14	0.22	0.20	0.35	
Instances		POD			
$M \times P$	GA	KA	RDA	H-RDKGA	
3 * 5	0.16	0.22	0.14	0.22	
4 * 6	0.18	0.18	0.19	0.21	
5 * 8	0.22	0.20	0.10	0.18	
6 * 9	0.15	0.14	0.11	0.16	
7 * 11	0.17	0.18	0.16	0.12	
8 * 13	0.19	0.14	0.12	0.18	
10 * 12	0.22	0.16	0.14	0.12	
11 * 13	0.22	0.18	0.14	0.16	
12 * 15	0.20	0.16	0.08	0.22	

done. Accordingly, we compare our metaheuristics with each other, as well as the results of the exact solver.

As the metaheuristics are naturally random, we run each algorithm 10 times and the best, the worst, the average, and the standard deviation of solutions among runs are reported. An average of the computational time of the algorithms is noted. To check the validation of the results, an exact solver implemented by GAMS software is used. Table 3 provides all the results. It should be noted that the exact solver is not able to find a solution for the large-scale instances after one hour. But all the metaheuristics can solve the problem in a few minutes.

Generally, the behavior of the algorithms in the criterion of the solution time is very close. Both hybrid algorithms and KA have a neck-and-neck competition. However, the KA is slightly better than all the metaheuristics. Without a doubt, the proposed H-RDKGA outperforms other algorithms and its solution is very close to the global solution based on the results.

Finally, the results indicate that, based on the average of the standard deviation of the results and the gaps of the algorithm in the interval plot, the proposed hybrid algorithm, that is, H-RDKAGA, is highly better than other algorithms and outperforms the best. After this algorithm, a

TABLE 2: Tuning of metaheuristics.

Metaheuristic	Parameters	Levels			Tuned value
		-1	0	+1	
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 N_1	0.1	0.2	0.3	0.1
	Percentage of N_2	0.4	0.5	0.6	0.6
	Maximum number of swirlings	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 swirlings	5	10	15	15

TABLE 3: Comparison of algorithms.

Algorithm		Test problems								
		P1	P2	P3	P4	P5	P6	P7	P8	P9
EX	B	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

B = best, W = worst, A = average, SD = standard deviation, EX = exact solver, and CPU = run time (seconds).

slight difference exists between the KA and the RDA, but the RDA is better. The last algorithm is the GA, as it has the weakest performance in this comparison.

5. Conclusion

Inter/intracell layouts and dynamic cell formation in a steady space were investigated concurrently in this paper by

a novel mixed-integer nonlinear programming model. This model was performed somehow to minimize the cost of the number of exceptional elements (EEs), parts total transportation expense, and cell redesigning. As the model was an NP-Hard problem, first, three metaheuristic algorithms are proposed for optimization. In the next step, to further improve the solutions, a new hybrid metaheuristic algorithm combining the results of the three metaheuristics is

proposed. To combine and improve the solutions of the algorithms, machine learning approaches are employed. More precisely, combining the merits of the aforementioned algorithms, the new metaheuristic algorithm is proposed.

Several recommendations can be proposed for better orientations in this study. Merging the proposed model for instance with a scheduling problem would be interesting. Also, to overcome the uncertainty, a two-stage or multistage stochastic programming method could be used. Applying more in-depth analyses by other large-scale optimization problems could be another approach from the aspect of the new suggested hybrid algorithm. Lastly, to evaluate the outcomes of the offered algorithms, new metaheuristics can be proposed.

Data Availability

The data and MATLAB codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Benjaafar, "Modeling and analysis of congestion in the design of facility layouts," *Management Science*, vol. 48, no. 5, pp. 679–704, 2002.
- [2] J. Balakrishnan and C. Hungcheng, "The dynamic plant layout problem: incorporating rolling horizons and forecast uncertainty," *Omega*, vol. 37, no. 1, pp. 165–177, 2009.
- [3] S. Wang and B. R. Sarker, "Locating cells with bottleneck machines in cellular manufacturing systems," *International Journal of Production Research*, vol. 40, no. 2, pp. 403–424, 2002.
- [4] T. Ghosh, 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, pp. 157–174, 2016.
- [5] A. S. Alfa, M. Chen, and S. S. Heragu, "Integrating the grouping and layout problems in cellular manufacturing systems," *Computers & Industrial Engineering*, vol. 23, no. 1-4, pp. 55–58, 1992.
- [6] I. Mahdavi, E. Teymourian, N. T. Baher, and V. Kayvanfar, "An integrated model for solving cell formation and cell layout problem simultaneously considering new situations," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 655–663, 2013.
- [7] K. L. Mak, Y. S. Wong, and X. X. Wang, "An adaptive genetic algorithm for manufacturing cell formation," *International Journal of Advanced Manufacturing Technology*, vol. 16, no. 7, pp. 491–497, 2000.
- [8] H. Bayram and R. Şahin, "A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear Programming embedded hybrid solution techniques," *Computers & Industrial Engineering*, vol. 91, pp. 10–29, 2016.
- [9] A. M. Golmohammadi, M. Honarvar, H. Hosseini-Nasab, and R. Tavakkoli-Moghaddam, "Machine reliability in a dynamic cellular manufacturing system: a comprehensive approach to a cell layout problem," *International Journal of Industrial Engineering & Production Research*, vol. 29, no. 2, pp. 175–196, 2018.
- [10] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [11] T. P. Carvalho, F. A. Soares, R. Vita, R. D. P. Francisco, J. P. Basto, and S. G. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, Article ID 106024, 2019.
- [12] N. Enshaei and F. Naderkhani, "Application of deep learning for fault diagnostic in induction Machine's bearings," in *Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–7, San Francisco, CA, USA, June 2019.
- [13] M. T. Leung, R. Quintana, and A.-S. Chen, "A paradigm for Group Technology cellular layout planning in JIT facility," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1174–1178, Singapore, December 2008.
- [14] I. Mahdavi 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, pp. 488–497, 2008.
- [15] I. Mahdavi, M. M. Paydar, M. Solimanpur, and A. Heidarzade, "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6598–6604, 2009.
- [16] R. Kia, A. Baboli, N. Javadian, R. Tavakkoli-Moghaddam, M. Kazemi, and J. Khorrami, "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, 2012.
- [17] Z. Alizadeh Afrouzy, M. M. Paydar, S. H. Nasser, and I. Mahdavi, "A meta-heuristic approach supported by NSGA-II for the design and plan of supply chain networks considering new product development," *Journal of Industrial Engineering International*, vol. 14, no. 1, pp. 95–109, 2018.
- [18] A. M. Golmohammadi, M. Honarvar, G. Guangdong, and H. Hosseini-Nasab, "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," *International Journal of Industrial Engineering & Production Research*, vol. 30, no. 4, pp. 405–427, 2019.
- [19] M. Mohammadi and K. Forghani, "Designing cellular manufacturing systems considering S-shaped layout," *Computers & Industrial Engineering*, vol. 98, pp. 221–236, 2016.
- [20] A. F. Fard and M. Hajiaghahi-Keshteli, "Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating," in *Proceedings of the International Conference on Industrial Engineering*, pp. 33–34, Tehran, Iran, January 2016.
- [21] A.-M. Golmohammadi, M. Honarvar, H. Hosseini-Nasab, and R. Tavakkoli-Moghaddam, "A bi-objective optimization model for a dynamic cell formation integrated with machine and cell layouts in a fuzzy environment," *Fuzzy Information and Engineering*, vol. 12, pp. 1–19, 2020.
- [22] X. Wu, C.-H. Chu, Y. Wang, and W. Yan, "A genetic algorithm for cellular manufacturing design and layout," *European Journal of Operational Research*, vol. 181, no. 1, pp. 156–167, 2007.

- [23] M. M. Paydar and M. Saidi-Mehrabad, "Revised multi-choice goal programming for integrated supply chain design and dynamic virtual cell formation with fuzzy parameters," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 3, pp. 251–265, 2015.
- [24] M. Hajiaghaei-Keshteli and M. Aminnayeri, "Keshtel Algorithm (KA); a new optimization algorithm inspired by Keshtels' feeding," in *Proceedings of the in IEEE conference on industrial engineering and management systems*, pp. 2249–2253, Bangkok, Thailand, December 2013.
- [25] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, "Red deer algorithm (RDA): a new nature-inspired meta-heuristic," *Soft Computing*, vol. 24, pp. 1–29, 2020.
- [26] N. Safaei, M. Saidi-Mehrabad, and M. S. 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, pp. 563–592, 2008.