Research Article

Simultaneous Hierarchical Clustering for Cell Formation Problems with Production Information

Yingyu Zhu and Simon Li

Department of Mechanical and Manufacturing Engineering, University of Calgary, Alberta, Canada

Correspondence should be addressed to Simon Li; simoli@ucalgary.ca

Received 13 January 2018; Revised 6 April 2018; Accepted 18 April 2018; Published 20 May 2018

Academic Editor: Muhammad N. Akram

Copyright © 2018 Yingyu Zhu and Simon Li. 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.

The purpose of this paper is to advance the similarity coefficient method to solve cell formation (CF) problems in two aspects. Firstly, while numerous similarity coefficients have been proposed to incorporate different production factors in literature, a weighted sum formulation is applied to aggregate them into a nonbinary matrix to indicate the dependency strength among machines and parts. This practice allows flexible incorporation of multiple production factors in the resolution of CF problems. Secondly, a two-mode similarity coefficient is applied to simultaneously form machine groups and part families based on the classical framework of hierarchical clustering. This practice not only eliminates the sequential process of grouping machines (or parts) first and then assigning parts (or machines), but also improves the quality of solutions. The proposed clustering method has been tested through twelve literature examples. The results demonstrate that the proposed method can at least yield solutions comparable to the solutions obtained by metaheuristics. It can yield better results in some instances, as well.

1. Introduction

The pressure of mass customization remains high for manufacturers to provide a family of product variants with low cost. To address this challenge, the strategy of group technology (GT) has been applied to focus on the production of part families using manufacturing cells. The general advantage is that each manufacturing cell can be flexibly adjusted to produce customized parts. Toward the successful application of GT, one important key is the design of cellular manufacturing systems (CMS), and cell formation (CF) is one early design decision that defines the machine groups and the part families.

Among a wide range of mathematical techniques to solve CF problems, cluster analysis is one of early attempts [1]. One key feature of cluster analysis is the concept of similarity coefficients that are used to quantify the “similarity” between any pairs of machines (or parts) for grouping purpose. The relevant techniques are labeled as similarity coefficient method (SCM) [2]. In this paper, SCM refers to the methodical approaches that use similarity coefficient under the traditional framework of hierarchical (or agglomerative)

clustering for solving CF problems. Along the advancement of optimization techniques (especially metaheuristics), SCM has received certain criticisms [3, p. 159], [4], which can be summarized into two aspects below:

- (i) Criticism #1: solution optimality: different from the exact solution strategies and metaheuristic algorithms, hierarchical clustering (HC) engages a smaller search space by comparing similarity values. Also, the grouping process in HC is irreversible so that the quality of final solutions may suffer due to some less-optimal subgroups formed earlier.
- (ii) Criticism #2: consideration of production factors: beyond the binary machine-part incidence matrix, CF problems need to consider more production factors such as production volumes, operation sequences, and machine reliability. Yet, SCM has been criticized for less capability to effectively handle complex production factors.

Concerning Criticism #1, it should be noted that CF problems are essentially NP-hard [5]. Thus, optimal solutions cannot

be guaranteed without extensive computational effort. The heuristics and irreversibility of HC can be viewed as the strategy that makes HC simple and quick to apply. Currently, the SCM algorithmic strategy mainly forms groups for the same type of objects at one time (i.e., deciding on machine groups or part families at one time). Yet, the nature of CF problems is about the formation of groups for two sets of objects [6]. Thus, this paper adapts the two-mode similarity coefficient [8] that puts machines and parts into one single hierarchy (i.e., dendrogram or tree) for simultaneous clustering. As a result, we can obtain high-quality CF solutions that are comparable to those from metaheuristic approaches.

Concerning Criticism #2, it should be noted that a certain number of similarity coefficients have been developed using production factors in literature, and more details can be found in Section 2.1. By taking the implied concern in this criticism, it is observed that the similarity coefficients with production factors are usually formulated in some complex equations. This makes the SCM less extendable to incorporate multiple production factors. To address this concern, this paper proposes the use of nonbinary incidence matrix to indicate the overall dependency strength between a machine and a part. In the analysis, the overall dependency strength is based on the aggregation of individual production factors so that the matrix entries can be quantified and adjusted according to the problem's context.

In this paper, we will address the CF problems with the consideration of production volumes and operation sequences. While this kind of problems has been addressed in literature, the new contributions are twofold. Firstly, the nonbinary incidence matrix is used to show its flexibility to handle different production factors. Secondly, the two-mode similarity coefficient is applied to support simultaneous clustering using HC. The quality of the results is at least comparable to the existing solutions in literature (some cases yield better results) with low computing time.

The rest of this paper is organized as follows. Section 2 will review the literature works concerning similarity coefficients and solution approaches to address CF problems. Section 3 will specify the CF problem to be addressed in this paper and discuss the use of nonbinary matrix. Section 4 will discuss the similarity analysis to support simultaneous hierarchical clustering. Section 5 will propose the methodical procedure to guide the solution process. Section 6 will demonstrate the proposed method with various examples and discuss the results to support the method's utility. Section 7 will conclude this paper.

2. Literature Review

2.1. Similarity Coefficients in CF. While the notion "similarity" sounds intuitive to many people, its formal meaning is not entirely straightforward. By examining the Jaccard similarity coefficient [1] and the classical works from numerical taxonomy [11] and psychology [12], it is found that a similarity measure generally consists of (1) commonality factors (to explain why two objects are similar) and (2) distinction factors (to normalize the measure with respect to some common ground). Then, the variety of similarity coefficients in the

CF literature is mainly caused by varying the commonality and distinction factors based on the interpretation of the production information for better CF design.

Being aimed at minimizing intercellular movements, the formulations of similarity measures mainly focus on the factors of production volume (to reduce the weight of movements) and operation sequences (to reduce the number of moves). In view of production volume, the early idea is to extend the Jaccard coefficient by multiplying the volume values with the matching and unmatching factors [13]. Then, it remains common to incorporate the production volume as the weight of movements in determining some generalized similarity measures (e.g., [14, 15]).

Beyond production volume, another common factor considered to formulate similarity measures is operation sequence, and two major lines of research are found in literature. One line is to evaluate the similarity of two parts based on their sequences of operations. One early work is found in Choobineh [16], where similarity was defined on the common subsequences between two parts. Then, the notions of longest common sequence (LCS) and shortest composite supersequence (SCS) emerged to consider the relative orders (rather than just consecutive orders) [17, 18]. Goyal et al. [19] further refined the similarity measure by considering the bypassing moves and appended sequences. Notably, nonsymmetric similarity measures have also been proposed by considering the number of operation changes required to produce part x based on the sequence of part y [20, 21].

In addition to part-part similarity, another line is to evaluate the similarity between two machines. Gupta [14] considered the similarity between two machines based on their consecutive processes to produce some common parts. This similarity idea was also adopted by Alhourani [22] and Wu and Suzuki [23, 24] in their proposals of similarity measures. To extend the generalized similarity by Won and Kim [25], Yin and Yasuda [26] developed the formulation of sequence ratio (SR) that compares the actual and theoretically highest intermachine movements between two machines. Nair and Narendran [3] focused on the intermediate operations that will incur two intercell movements as compared to the first and last operations. This similarity notion was also adopted by Won and Lee [27].

The notion of similarity keeps expanding in CF and other manufacturing applications. To support the process of ship-building, Yuguang et al. [28] classified the characteristics of similarity in view of assembly processes, equipment, and resources. Alhourani [29] considered machine reliability, where two machines are "less similar" if they have high breakdown costs individually. To analyze the networked operations, Navaei and ElMaraghy [30] developed the similarity measure of two part variants based on common consecutive operations.

Besides, the similarity comparison is always made between two objects of the same type. Yet, CF problems are intended to form groups of two types of objects, where traditional clustering methods such as k-means clustering and hierarchical clustering mainly focus on grouping objects of the same type. Boutsinas [31] has highlighted such difference

by adapting biclustering techniques to solve CF problems. Li and Mehrabadi [8] have proposed a two-mode coefficient that assesses the appropriateness of putting a machine and a part together in one group. This paper is intended to extend this work by applying the two-mode coefficient to resolve CF problems with more production factors.

2.2. Search Strategies for CF Solutions. Due to the combinatorial nature of grouping decisions, the search space for CF solutions is often large. In view of this fundamental challenge, some reviews have already classified CF methods based on the solution search strategies [2, 4, 32]. While this paper does not intend to propose a new taxonomy, we want to highlight that SCM should be classified as heuristics in view of its algorithmic procedure. To explain this point, we employ the classification of search strategies for CF solutions in two main categories [32]: (1) exact solution approach and (2) approximate solution approach, which includes metaheuristics and heuristics.

The main feature of exact solution approach is the assurance of optimal solutions. To employ this approach, the first step is to formulate CF problems using integer variables for assignment and grouping decisions such as the p -median methods [33, 34] and the generalized CF models [35, 36]. Notably, using integer programming models only does not imply the obtainment of optimal solutions. Early literature has reported that exact solution methods (e.g., branch-and-bound) implemented in software (such as LINDO) can only address relatively small problems [34, 37]. Further advancements have tackled the “fractional” grouping efficacy objective through approximately linearizing the grouping efficacy formulation [38] and strategically fixing the demonstrator values in a series of linear integer problems [39]. Brusco [6] has strengthened the potential of exact solution approach by strategically setting the lower and upper bounds in a branch-and-bound approach with clarifying the scalability of the proposed algorithm.

In contrast to exact solution approach, approximate solution approach does not guarantee the optimality of solutions. By relaxing this requirement, the algorithmic strategies based on metaheuristics and heuristics are generally applied to search for high-quality solutions without enumerative exploration. According to Blum and Roli [40], one important feature of metaheuristic algorithms is the control of the balance between diversification and intensification in the solution search process. The results of using metaheuristics for solving CF problems are abundant in literature, as reviewed in Papaioannou and Wilson [4]. Examples include tabu search [41], simulated annealing [42, 43], and genetic algorithm [44, 45]. Yet, the use of metaheuristics is not without challenges. As commented by Stawowy [46, p. 1], the approaches by metaheuristics have a tendency for more complex algorithms and are sometimes not easy for CF practitioners. Also, Renzi et al. [32, p. 415] have indicated that the choice of a metaheuristic method (mainly pointing to genetic algorithms) may not be informed by the research results from optimization literature. Brusco [7, p. 293] has remarked on some common metaheuristics that need to make certain

nontrivial “implementation decisions” in order to obtain desirable outcomes.

Compared to metaheuristics, heuristics does not carry the algorithmic feature of diversification and intensification. Instead, it mainly employs the greedy and hill-climbing algorithmic strategy to promptly improve the solution quality based on the short-term results. Without backtracking, one advantage is its simplicity and low computational effort. This explains that some early CF solution approaches (e.g., [47, 48]) are heuristic by nature as they can even be applied without using computers. By examining the procedure of hierarchical clustering (or SCM) closely, it should be classified as a heuristic solution approach since it progressively picks the object pair with highest similarity and forms intermediate groups. The nature of this strategy is close to the greedy algorithm [49, p. 64]. Thus, hierarchical clustering also inherits the characteristics of heuristic solution approach such as lower computational effort and weaker guarantee to high-quality solutions.

Even though the cost of computation is getting low, clustering methods and SCM have still been applied in CF in recent years, as reviewed in Section 2.1. One supporting reason is that SCM is more flexible to handle less-formalized information [2], while exact solution and metaheuristic approaches require a complete optimization model to operate the algorithms. Another reason is simplicity, where the complexity of some metaheuristic algorithms can hinder the practical applications. In this context, by inheriting the advantages of SCM, this paper will show that two-mode similarity can yield comparable solution quality for the CF problems considering production volumes and operation sequences.

3. Analysis of the CF Problem Using Nonbinary Matrix

3.1. Context of the CF Problem. In a cell formation (CF) problem, there exists a set of machines ($m_1, m_2, \dots$) and parts ($p_1, p_2, \dots$). The variations of CF problems stem from the production factors that are considered in the problems. This paper specifically considers operation sequences and part demands. Let $S = [s_{ij}]$ be the matrix capturing the information of operation sequences [50], and it is defined as follows:

$$s_{ij} = \begin{cases} g & \text{if } p_j \text{ requires } m_i \text{ to process at the } g\text{th step} \\ 0 & \text{if } p_j \text{ does not require } m_i. \end{cases} \quad (1)$$

Also, let v_j be the demands of part p_j . Using an example from Solimanpur et al. [9], Figure 1 presents a matrix form that captures the sequence information of five machines and seven parts with their demands. Generally, the purpose of CF is to form the machine groups and part families so that some measure of solution quality can be optimized. In this paper, we choose grouping efficacy and intercellular movements as the quality measures. While these measures are

	p_1	p_2	p_3	p_4	p_5	p_6	p_7
m_1	0	1	0	2	0	0	1
m_2	0	0	2	0	1	0	0
m_3	2	2	0	3	0	0	2
m_4	1	0	1	0	0	1	0
m_5	0	0	3	1	2	2	0
v_j	80	110	140	95	120	80	135

FIGURE 1: Sample matrix with sequence and demand information [9].

well recognized in CF literature, they will be discussed in the next subsection.

3.2. Measures of Solution Quality. To compare the quality of CF solutions, one common measure is grouping efficacy (denoted as μ) [6], and its formulation is provided below:

$$\mu = \frac{n_e - n_{\text{out}}}{n_e + n_{\text{in}}}, \quad (2)$$

where n_e , n_{out} , n_{in} are the total numbers of all operations (i.e., number of nonzero matrix entries), operations outside the cells (i.e., exceptional elements), and voids (i.e., zero matrix entries) inside the cells, respectively. In an ideal solution, there is no operation outside the cells (i.e., $n_{\text{out}} = 0$) and no void inside the cells (i.e., $n_{\text{in}} = 0$), leading to $\mu = 1$.

To incorporate the sequence and demand information in the solution comparison, the objective function by Solimanpur et al. [9] is employed, and it basically consists of two measures: intercellular movements (f_1) and total number of voids (f_2). To capture the intercellular movements, we need to distinguish whether the operations are at both ends or in the middle of the sequence. Particularly, if an operation is in the middle and the relevant part and machine are not in the same cell, this part needs to move out and then move in with respect to its own cell (i.e., 2 times). Let $D = [d_{ij}]$ be the weighted sequence matrix [3] to capture this idea, and it is defined as follows:

$$d_{ij} = \begin{cases} 0 & \text{if } s_{ij} = 0 \\ 1 & \text{if } p_j \text{ requires } m_i \text{ at the start or end of sequence} \\ 2 & \text{if } p_j \text{ requires } m_i \text{ in the middle of sequence.} \end{cases} \quad (3)$$

Let e_{ij} and o_{ij} be the binary parameters indicating the presence of exceptional and void elements with machine m_i and part p_j in the solution matrix, respectively. Also, let m and n be the number of machines and parts, respectively. Accordingly, f_1 and f_2 can be formulated in the following:

$$f_1 = \sum_{i=1}^m \sum_{j=1}^n v_j \cdot d_{ij} \cdot e_{ij}, \quad (4)$$

$$f_2 = \sum_{i=1}^m \sum_{j=1}^n v_j \cdot o_{ij}. \quad (5)$$

3.3. Formulation of Nonbinary Matrix. The notion of nonbinary matrix can be viewed as a generalization of the traditional incidence matrix in CF which indicates the binary relation between a machine and part (i.e., whether machine m_i is required to process part p_j). The “nonbinary” entries can be interpreted as “to what extent p_j needs m_i based on the production information”. The input of this step is the production information. Specifically in this paper, it refers to s_{ij} (operation sequences) and v_j (part demands).

As discussed earlier, “middle” operations have higher priority than “start/end” operations to be put in the same cell in order to avoid the double counts of intercellular movements. To reflect this idea, the weighted sequence matrix $D = [d_{ij}]$ defined in (3) will be used to capture the “nonbinary” notion for operation sequences.

With regard to part demands, a high demand of p_j will encourage putting the required machines in the same cell so that we can avoid moving a large batch of p_j between cells. To capture this notion, let $V = [v_{ij}]$ be the part demand matrix. In the formulation, if m_i is required to produce p_j with demand v_j , then $v_{ij} = v_j$. Otherwise, if m_i is not required to make p_j , $v_{ij} = 0$. For illustration, the sample example from Figure 1 is used to generate the weighted sequence matrix D and the part demand matrix V , which are shown in Figures 2(a) and 2(b), respectively.

After obtaining $D = [d_{ij}]$ and $V = [v_{ij}]$ for operation sequences and part demands, respectively, the next process is normalization and weighting to yield the nonbinary matrix as the output of this step. Let $A = [a_{ij}]$ be the nonbinary matrix, and its formulation is provided below:

$$a_{ij} = w_1 \frac{d_{ij}}{\sum_{i=1}^m d_{ij}} + w_2 \frac{v_{ij}}{\sum_{j=1}^n v_j}, \quad (6)$$

where w_1 and w_2 are the weighting factors, and the denominators are applied for normalization. Particularly, the weighting factors are intended to balance the effects from both production factors. To do so, the following constraint is set:

$$w_1 \cdot \sum_{i,j} \left(\frac{d_{ij}}{\sum_i d_{ij}} \right) = w_2 \cdot \sum_{i,j} \left(\frac{v_{ij}}{\sum_j v_j} \right). \quad (7)$$

In addition, as we also set $w_1 + w_2 = 1$, the values of w_1 and w_2 can be determined with these two equations. Accordingly, the nonbinary matrix $A = [a_{ij}]$ is obtained. The resulting nonbinary matrix A of the sample example is shown in Figure 2(c).

4. Similarity Analysis

4.1. Background. In the context of SCM, similarity is often used to measure the potential of putting two objects in the same group before any formation of groups. While the direct interpretation of similarity is about the resemblance of two objects, Yin and Yasuda [2] have refined the notion of similarity as “appropriateness” (p. 335). That is, the similarity of two objects depends on the purpose of grouping, which defines the context of appropriateness. For example, suppose that we have three different pens: (Pen 1) blue ball pen, (Pen

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
$m1$	0	1	0	2	0	0	1
$m2$	0	0	2	0	1	0	0
$m3$	1	1	0	1	0	0	1
$m4$	1	0	1	0	0	1	0
$m5$	0	0	1	1	1	1	0

(a) Weighted sequence matrix D

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
$m1$	0	110	0	95	0	0	135
$m2$	0	0	140	0	120	0	0
$m3$	80	110	0	95	0	0	135
$m4$	80	0	140	0	0	80	0
$m5$	0	0	140	95	120	80	0

(b) Part demand matrix V

	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$
$m1$	0	0.23	0	0.21	0	0	0.25
$m2$	0	0	0.26	0	0.24	0	0
$m3$	0.2	0.23	0	0.15	0	0	0.25
$m4$	0.2	0	0.2	0	0	0.2	0
$m5$	0	0	0.2	0.15	0.24	0.2	0

(c) Nonbinary matrix A FIGURE 2: Sample matrices D , V , and A .

2) green ball pen, and (Pen 3) green highlighter. In view of the writing function, we may say that Pens 1 and 2 are more similar due to the width of a pen stroke. Yet, if we want to categorize pens based on colors for organization purpose, we may say that Pens 2 and 3 are more similar.

Despite the appropriateness notion, similarity in SCM is often used to compare two objects of the same type as if it follows the resemblance notion of similarity. For example, the similarity coefficients reviewed in Section 2.1 were developed for either any two machines or any two parts. We refer to these as one-mode similarity, where the term “mode” is defined as an object type in social network analysis [51]. Yet, the nature of CF problems is about the formation of groups for two sets of objects. Consequently, a clustering procedure needs to be conducted in two stages, either forming machine groups first or part families first. In this case, the solution quality may be impacted since the groups formed earlier restrict the possible results of second grouping or assignment decisions.

To extend the appropriateness notion of similarity in CF, this paper uses the two-mode similarity coefficient to assess how it is appropriate to group a machine and a part in the same group. While more details can be found in Section 4.3, the key feature of two-mode similarity is to support simultaneous hierarchical clustering so that both machines and parts can form the hierarchy in the same tree. Our numerical examples show that this approach can provide high-quality CF solutions at least comparable to literature results.

To regulate the notion of similarity and support the concatenation of their values, the following conditions are applied to regulate the numerical properties of similarity values. These conditions or axioms are based on the work by Sneath and Sokal [11, page 120]. Let $f_s(a, b)$ be the function that computes the similarity values between two objects a and b . Then, a valid similarity function needs to satisfy the following conditions:

Condition #1: $0 \leq f_s(a, b) \leq 1$.

Condition #2: $f_s(a, b) = f_s(b, a)$.

Condition #3: $f_s(a, a) = 1$.

Condition #1 is the boundary condition that any similarity value is bounded between 0 and 1. It sets the numerical range and tendency (e.g., “1” implies perfect similarity) for aggregation. Condition #2 is the symmetry condition that the order of comparison does not influence the similarity value. Condition #3 states that the comparison between two identical objects should yield the maximum similarity value.

4.2. Min/Max Similarity Coefficient. The purpose of the min/max coefficient is to assess the similarity between any two machines and any two parts using the nonbinary matrix $A = [a_{ij}]$ defined in Section 3.3. In particular, two machines have nonzero similarity if they need to produce some common parts. Let smm_{ij} be the min/max coefficient of two machines m_i and m_j , and its formulation is given below:

$$smm_{ij} = \frac{\sum_{k=1}^n \min(a_{ik}, a_{jk})}{\sum_{k=1}^n \max(a_{ik}, a_{jk})}. \quad (8)$$

Likewise, two parts have nonzero similarity if their productions require some common machines. Let spp_{ij} be the min/max coefficient of two parts p_i and p_j , and its formulation is given below:

$$spp_{ij} = \frac{\sum_{k=1}^m \min(a_{ki}, a_{kj})}{\sum_{k=1}^m \max(a_{ki}, a_{kj})}. \quad (9)$$

Notably, the min/max coefficient satisfies the three similarity conditions discussed in Section 4.1. In addition, if the input $A = [a_{ij}]$ is a binary matrix, the min/max coefficient will give the same results as the Jaccard coefficient. In other words, the min/max coefficient can be viewed as the generalization of the Jaccard coefficient, where the nonbinary values in $A = [a_{ij}]$ are treated as the weights, in addition to the binary information, to assess the degree of similarity.

4.3. Two-Mode Similarity Coefficient. The purpose of the two-mode coefficient is to assess the similarity between a machine and a part using the nonbinary matrix $A = [a_{ij}]$ defined in Section 3.3. As discussed earlier, the term two-mode is used to indicate the similarity assessment between two objects of different types. Along the notion of appropriateness, the similarity between machine m_i and part p_j is nonzero if m_i is used to produce p_j . Let smp_{ij} be the two-mode coefficient of m_i and p_j , and its formulation is given below:

$$smp_{ij} = \frac{2a_{ij}}{\sum_{k=1}^n a_{ik} + \sum_{k=1}^m a_{kj}}. \quad (10)$$

Regarding the numerical tendency, if the value of a_{ij} is high, the resulting similarity will also be high. The denominator basically normalizes the similarity value by summing the nonbinary entries a_{ij} of the row of m_i and the column of p_j . Also, it captures how exclusive m_i and p_j are related to each other. For example, if m_i is only used to produce p_j (but not other parts), the row of m_i in the nonbinary matrix has mainly zero values except the entry a_{ij} . The two-mode coefficient is intended to capture this tendency putting m_i and p_j in the same cell due to this exclusiveness.

For more elaboration, if p_j does not require m_i , the two-mode coefficient is equal to zero. If a_{ij} is the only nonzero entry for m_i and p_j (i.e., machine m_i is only used to produce p_j), the two-mode coefficient is equal to one (i.e., no reason to separate m_i and p_j in different cells). The numerator is set as $2 \times a_{ij}$ in order to satisfy the boundary condition of similarity discussed in Section 4.1. Notably, the symmetry condition is also satisfied, while the condition of identical objects is not applicable for the two-mode coefficient.

4.4. Concatenated Similarity Matrix. To execute simultaneous hierarchical clustering, the similarity information of both machines and parts are treated at the same time. Then, three types of similarity matrices are classified below:

- (i) Machine-machine similarity: an $m \times m$ symmetric matrix (denoted as $S_{mm} = [smm_{ij}]$) is used to capture the similarity values between any two machines. The min/max coefficient is used in this case.
- (ii) Part-part similarity: an $n \times n$ symmetric matrix (denoted as $S_{pp} = [spp_{ij}]$) is used to capture the similarity values between any two parts. The min/max coefficient is used in this case.
- (iii) Machine-part similarity: an $m \times n$ rectangular matrix (denoted as $S_{mp} = [smp_{ij}]$) is used to capture the similarity values between a machine and a part. The two-mode coefficient is used in this case.

To combine these three types of similarity matrices, the concatenated similarity matrix (CSM) is defined in (11). In this definition, the superscript T refers to the matrix transpose, and w_m , w_p , and w_{mp} are the scalars of weighting factors. Notably, CSM is a symmetric matrix with the dimension of $(m+n) \times (m+n)$. Since CSM is constructed by various types of similarity coefficients, the weighting factors (i.e., w_m , w_p ,

FIGURE 3: Illustration of the methodical workflow.

and w_{mp}) are used to adjust the grouping effects from S_{mm} , S_{pp} , and S_{mp} :

$$CSM = \begin{bmatrix} w_m \cdot S_{mm} & w_{mp} \cdot S_{mp} \\ w_{mp} \cdot S_{mp}^T & w_p \cdot S_{pp} \end{bmatrix}. \quad (11)$$

After obtaining CSM, both machines and parts are treated as if they belong to the same type of objects. Then, traditional procedures of hierarchical clustering can be applied using CSM as an input to construct a tree (or dendrogram) that can group both machines and parts at the same time. The next section will discuss the application of hierarchical clustering that takes CSM to solve the CF problem.

In this paper, we employ two approaches to set the weighting factors. The first approach is simply setting $w_m = w_p = w_{mp} = 1$, and it is intended to preserve the original grouping influence from the min/max and two-mode coefficients. The second approach is to set $w_m = w_p = 1$, and w_{mp} is set using (12) in order to balance the effect from the two-mode coefficient. After that, the concatenated similarity matrix (CSM) is normalized through division of its largest entry

$$w_{mp} = \frac{\max(S_{mm}) + \max(S_{pp})}{2 \max(S_{mp})}. \quad (12)$$

5. Methodical Procedure

As discussed by Yin and Yasuda [2], the similarity coefficient method (SCM) generally has a prescribed procedure. Essentially, this paper adapts this SCM procedure and proposes a 4-step method for problem solving. Figure 3 illustrates the overview of these procedural steps, with emphasis on the input and output information of each step [52].

Step 1 (process production information into nonbinary matrix). The purpose of this step is to convert the production information into a nonbinary matrix that indicates the relational strength between m_i and p_j . Higher relational strength

FIGURE 4: Tree result of the sample matrix constructed from hierarchical clustering.

implicates that the corresponding m_i and p_i should be put in the same cell. In this paper, the production information considered is operation sequences (i.e., $S = [s_{ij}]$) and part demands (i.e., v_j). The output of this step is the nonbinary matrix, $A = [a_{ij}]$, whose formulation has been discussed in Section 3.3.

Step 2 (determine the similarity values). This step corresponds to the application of similarity coefficients in SCM. The purpose of this step is to evaluate the similarity values of machine-machine (i.e., S_{mm}), part-part (i.e., S_{pp}), and machine-part (i.e., S_{mp}) based on the nonbinary matrix. The similarity coefficients used in this step are the min/max coefficient for S_{mm} and S_{pp} (i.e., (8) and (9)) and the two-mode coefficient for S_{mp} (i.e., (10)). The output of this step is the concatenated similarity matrix (CSM), which has been discussed in Section 4.4.

Step 3 (build the tree and block diagonal matrix). This step corresponds to the application of hierarchical clustering algorithms to construct the tree (or dendrogram). Using the tree's nodes, the block diagonal matrix (BDM) is obtained by rearranging the rows and columns of the input matrix. The purpose of this step is to preliminarily sort the input matrix by putting similar objects close to each other.

In this paper, the average-linkage algorithm in hierarchical clustering [53] is applied to CSM in order to build the tree with both machines and parts. Taking the CSM as the input, the tree is constructed based on the similarity values between any two objects. If two objects have a high similarity value, they will be likely grouped to form a branch of the tree. As objects can be either machines or parts, one single tree is constructed for these two object types. Figure 4 shows the resulting tree using the sample matrix (i.e., Figure 1), where the nodes include both machines and parts (i.e., m_i and p_j), and a branch is created after joining two nodes (denoted as br_k). For example, m_2 and p_5 are joined to make br_2 in Figure 4.

Notably, the hierarchical clustering process already places the highly similar objects close to each other. Thus, if we reorder the rows and columns of the nonbinary matrix based on the node sequence from the tree, we can arrange the nonzero matrix entries getting close to the matrix's diagonal, forming a block diagonal matrix (BDM). To improve the quality of BDM, a node sequencing algorithm is applied [54]. The basic idea of this algorithm is to check whether switching tree branches (e.g., p_2 and p_5) can yield a better BDM or not.

	p_2	p_7	p_4	p_1	p_6	p_3	p_5
m_1	1	1	2	0	0	0	0
m_3	2	2	3	2	0	0	0
m_4	0	0	0	1	1	1	0
m_5	0	0	1	0	2	3	2
m_2	0	0	0	0	0	2	1
v_j	110	135	95	80	80	140	120

FIGURE 5: Sample of the block diagonal matrix.

If so, branch-switch is carried out. This process starts from the top branch, proceeding down to the lower branches. Figure 5 shows the sample of BDM, which is obtained by reordering the rows and columns of the sample matrix (i.e., Figure 1) based on the tree node sequence in Figure 4.

Step 4 (partition the block diagonal matrix). Note that structural constraints (e.g., number of cells and size limits) are not required to yield the block diagram matrix (BDM). Then, the purpose of this Step 4 is to determine machine groups and part families by dividing the BDM with the consideration of structural constraints. In the CF literature, various approaches have been proposed to yield final CF solutions based on the BDM such as the minimal spanning tree approach (Chen and Irani 1993) and the evolutionary algorithm approach [55]. In this paper, the tree structure is utilized for finalizing the cell formation process.

Using the previous sample example, Figure 6 illustrates the partitioning concept based on the tree and BDM. By breaking the top branch of the tree, two subsets of nodes are indicated. As each subset contains both machines and parts, it basically defines a block on the BDM. These blocks then define the contents of the cells in the final CF solution. In the algorithmic implementation, the tree's branches are broken from top to bottom to collect the positions to define the blocks over the BDM. These positions are applied to the BDF progressively until the structural criteria of cells (e.g., the number of cells) are satisfied.

6. Numerical Examples and Comparison

To examine the proposed method, three examples are first discussed in Section 6.1 to check (1) the use of hierarchical clustering to handle the sequence information, (2) the min/max similarity coefficient to handle the nonbinary information, and (3) the ability to handle a large matrix problem. Then in Section 6.2, benchmark results based on metaheuristics will be used to examine the solution quality from the proposed method.

6.1. Three Examples to Examine the Basic Functionality. The first example is obtained from Nair and Narendran [3], and it has 20 machines and 20 parts. Based on the previous discussion, the similarity analysis is conducted by first constructing the weighted sequence matrix using (3), and then the min/max coefficient is applied to evaluate S_{mm} and S_{pp} . The weights of the CSM are applied based on (12). As

TABLE 1: Comparison of numerical metrics of the nonbinary example.

	Number of cells	Exception (n_{out})	Void (n_{in})	Grouping efficacy (μ)
Original solution	4	53	55	0.3494
Our solution	5	62	19	0.3769

FIGURE 6: Illustration of the partitioning concept based on tree.

a result, the five-cell solution is obtained and shown in Figure 7. The solution in Figure 7 is essentially the same as the original solution by Nair and Narendran [3] in view of the composition of machines and parts in individual cells. Notably, their original solution was obtained by using non-hierarchical clustering, which requires the identification of “disjoint machines” to define centroids as initial inputs of the clustering process. In contrast, the proposed clustering method does not require the centroids as the method’s inputs to initiate the solution process. This feature is considered as one practical advantage of the proposed method as compared to the nonhierarchical approach.

The second example is obtained from Venugopal and Narendran [10], and it has 20 machines and 20 parts. The nonbinary value between m_i and p_j is interpreted as the normalized workload of m_i required to produce p_j , and it is calculated based on processing time, machine available time, and production volume. In this case, higher values imply stronger correlations; that is, the corresponding machines and parts should be grouped in one cell. Figure 8(a) shows the CF solution after applying the proposed method. In this solution, five cells are defined. Tentatively, we observe that the values of intercell entries are relatively small. Also, this solution generally shows the BDM, which arranges the blocks along the diagonal. This BDM can help users to view the overall structure easier and support the rearrangement of the block’s contents (if necessary).

For comparison, Figure 8(b) shows the original solution by Venugopal and Narendran [10] that was obtained by using simulated annealing. These two solutions are almost the same except that the original solution has only four cells. The large block of the original solution (whose matrix’s labels are highlighted in Figure 8(b)) is essentially the combination of two smaller blocks in our solution (whose labels are also highlighted in Figure 8(b)). Purely from the aspect of solution

quality, these two solutions are similar. Yet, the BDM of our solution has some advantage if we want to flexibly change the number of cells from four to five (or vice versa). Table 1 records the numerical results for both our solution and the original solution. As seen, our solution has a higher (better) value of grouping efficacy.

The third example comes from Chandrasekharan and Rajagopalan [56], which was originally a CF problem of 40 machines and 100 parts with binary information only. To examine the handling of sequence and demand information, the operation sequence is generated for each part based on the order of the machine indices (a method used by Spiliopoulos and Sofianopoulou [57]). The demand of each part was randomly generated using a discrete uniform distribution with a range of [1, 100] (a method used by Solimanpur et al. [9]). By applying the proposed method, a solution is obtained and represented in Figure 9, where the parts’ demands are shown on the last row. The grouping efficacy of this solution is 0.8403, which matches the result in Chandrasekharan and Rajagopalan [56] and other recent results [7, 39]. Also, the execution time of this problem is less than 0.5 second based on Matlab implementation using a typical computer (speed: 3.1 GHz, RAM: 8 GB). Thus, the solution time is considered fast as compared to other optimization approaches.

6.2. Comparison with the Results from Metaheuristics. The benchmark results were obtained from the work by Solimanpur et al. [9], where nine examples of CF problems are used for the comparison in this study. Notably, the original work by Solimanpur et al. [9] mainly adopted the CF matrices from literature, and they added the information of part demands and operation sequences if necessary. To conduct an even-ground comparison, we adopted the information of CF problems purely from Solimanpur et al. [9] using the original problem

	m6	m15	m7	m4	m13	m19	m20	m8	m10	m9	m1	m18	m12	m3	m2	m11	m5	m14	m17	m16
p16	3	1	2	0	0	4														
p13	1	3	2	0	0														4	
p5	3	2	4	1	0															
p8	0	1	3	5	2					4										
p3						3	2	1												
p10						1	2	3												
p18						2	3	1	4											
p14								1	2	0	3	0	0		4					
p20									2	0	3	1	4							
p17									0	1	2	0	3							
p1							5		0	3	2	4	1							
p9									0	2	4	1	5			3				
p12									0	1	5	2	4				3			
p11								1						3	0	0		2		
p4									4					1	3	2				
p19														1	2	3	4			
p2														3	2	1				
p6																1	5	2	4	3
p15					1												0	2	4	3
p7																	1	0	3	2

FIGURE 7: Solution of the example with sequence information only.

	p1	p9	p7	p18	p15	p20	p3	p5	p10	p2	p11	p16	p17	p6	p8	p19	p14	p4	p13	p12	
m1	0.38	0	0.32	0.1		0.12															
m9	0.36	0.36	0.24		0.06											0.02			0.04		
m6	0.27	0.34	0.21													0.04	0.06		0.07		
m16	0.21	0.31	0.32	0.02	0.04		0.01					0.01	0.03							0.07	
m19	0	0.36	0.33				0.08			0.05								0.09			
m20	0	0	0.52								0.14						0.16				
m7	0	0	0.49	0.18							0.1										
m8				0.32	0.24	0.21	0.17				0.08				0.12	0.09					
m3				0	0.27	0.23	0.34	0.11		0.09											
m13	0.01					0.02	0.29	0.18	0.21	0.26								0.03		0.06	
m15							0.22	0.27	0.27	0.18					0.06			0.09			
m5							0	0	0.31	0.34	0.14	0.03									
m14		0.03	0.01				0.1				0.02	0.11	0.2	0.17	0.13	0.14	0.19			0.01	
m10								0.04	0.02							0.17	0.14		0.23	0.38	0
m2										0.11									0	0.52	0
m11	0.03					0.01		0.02				0.26							0	0.37	0.32
m18			0.04			0.02				0.01					0.14		0.12		0	0.35	0.33
m17											0.12							0.18		0	0.31
m4						0.09	0.07				0.05		0.18		0.21				0	0	0.36
m12																			0	0	0.58

(a) Our solution

	p1	p7	p9	p5	p2	p10	p11	p12	p13	p4	p6	p8	p14	p16	p17	p19	p15	p3	p18	p20	
m1	0.38	0.32	0																0.1	0.12	
m6	0.27	0.21	0.34							0.07	0.04	0.06									
m7	0	0.49	0		0.1						0.09					0.12			0.18		
m9	0.36	0.24	0.26							0.04		0.02						0.06			
m16	0.21	0.32	0.31					0.07							0.01	0.03					
m19	0	0.33	0.36				0.05	0.1							0.09			0.04	0.01	0.02	
m20	0	0.52	0																	0.08	
m5					0	0.31	0	0.34						0.16							
m13	0.01				0.29	0.21	0.18	0.26		0.06					0.14	0.03		0.11			
m15					0.22	0.27	0.27	0.18											0.02		
m12								0.58	0	0	0	0	0	0	0	0					
m17								0.39	0.31	0	0	0	0.18	0	0	0					
m10								0	0.38	0.23	0	0.17	0	0	0	0.14					
m11	0.03							0.32	0.37	0	0	0	0	0	0.26	0		0.01			
m2								0	0.52	0	0	0	0	0	0	0					
m4								0.36	0	0	0	0.21	0	0	0.18	0		0.07		0.09	
m18		0.04						0.33	0.35	0	0	0.14	0.12	0	0	0		0.02			
m14		0.01	0.03					0.01	0.03	0.35	0	0	0.14	0.12	0	0		0.01			
								0.01	0	0	0.17	0.13	0.19	0.11	0.2	0.14					
m3					0.11	0.09												0.27	0.34	0	0.23
m8						0.08												0.24	0.17	0.32	0.21

(b) Solution from Venugopal and Narendran [10]

FIGURE 8: Solution comparison of the nonbinary example.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 9: Solution of the large example with sequence and demand information.

TABLE 2: Comparison with the results by the ACO method [9].

Problem	Size ($m \times n$)	Number of cells	Solution from the ACO method			Proposed clustering method			Total improvement (%)
			f_1	f_2	$f_1 + f_2$	f_1	f_2	$f_1 + f_2$	
P0	15×25	3	863	803	1666	863	803	1666	0.00
P1	8×20	3	800	0	800	800	0	800	0.00
P2	20×35	4	2058	2584	4642	1671	2911	4582	1.31
P3	30×50	13	2102	3268	5370	3619	1228	4847	10.79
P4	24×40	9	3772	2172	5944	2585	2067	4652	27.77
P5	9×10	2	377	1039	1416	309	1005	1314	7.76
P6	7×10	3	74	453	527	222	153	375	40.53
P7	10×12	3	589	288	877	236	368	604	45.20
P8	25×40	6	1685	4247	5932	1394	3421	4815	23.20

codes (e.g., P1, P2, P3,...) with the acknowledgment of original matrices and relevant works as follows.

- (i) P1: Mahdavi and Mahadevan [58] developed a heuristic algorithm, namely, CLASS.
- (ii) P2: Boe and Cheng [59] developed a (heuristic) close neighbour algorithm to obtain a block diagonal structure.
- (iii) P3: Stanfel [60] used a (heuristic) divisive clustering approach.
- (iv) P4: Srinivasan [61] used the minimum spanning tree method.
- (v) P5: Wu et al. [62] used genetic algorithms.
- (vi) P6: Tsai and Lee [63] used mathematical programming.
- (vii) P7 and P8: Sudhakara Pandian and Mahapatra [64] used neural networks.

Solimanpur et al. [9] applied the ant colony optimization (ACO) method to solve the CF problems. Notably, ACO is one type of metaheuristic methods, and it should exhibit better capability to search for high-quality solutions than heuristic approaches. Adopting the same measure, the quality of solutions is assessed by intercellular movements (i.e., f_1 in (4)) and the number of voids as machine utilization (i.e., f_2 in (5)). While both f_1 and f_2 measures are smaller-the-better, the summation of f_1 and f_2 (i.e., $f_1 + f_2$) is applied to compare the overall quality of solutions.

Table 2 shows the comparison results. For clarification, P0 is the test problem in Solimanpur et al. [9]. These results are based on the weighting of the CSM using (12) for all problems here except P1, which uses $w_m = w_p = w_{mp} = 1$. In this study, we actually run both sets of weighting and report the better results. Admittedly, how to set the weights of the CSM toward the high-quality solutions remains an open question. Yet, our observations over the reported examples indicate that the solutions are not highly sensitive to the weight values. Given that the computational effort of the proposed clustering method is low, it is still feasible to check two different sets of weights suggested in Section 4.4 for each problem.

As observed, the proposed clustering method can yield better solutions for most problems (i.e., from P2 to P8). Beyond the simple comparison of the value of f , it is noted that the values of f_1 and f_2 are often in the trade-off situation (i.e., better value of f_1 traded with weaker value of f_2), and no solutions are dominated in view of Pareto optimality. Concerning our solution results that are the same (i.e., P0 and P1), we consider that Solimanpur et al. [9] have already obtained very high-quality solutions for P0 and P1. In this view, our solutions can at least match these best solutions reported in literature.

6.3. Remarks on the Advantages of the Proposed Method. In summary of this section, the main advantage of the proposed method is its ability to yield high-quality CF solutions (comparable to the results by metaheuristics) without compromising the computational efficiency of heuristic solution approach. In other words, the proposed method can be viewed as one that improved heuristics based on hierarchical clustering to search for high-quality solutions. Table 3 compares the proposed method (on the last row) with the major types of algorithmic approaches that have been discussed in Section 2.1. The columns of this table list the comparison aspects, which are discussed below:

- (i) Solution search strategy: this aspect has been discussed in Section 2.1. Its intent is to recall the basic difference of various strategies in searching for CF solutions. Such information serves the foundation to explain their performance differences.
- (ii) Guarantee of optimal solutions: only the exact solution approach can guarantee optimal solutions. The traditional view considers that metaheuristics has better potential for high-quality solutions than heuristics. Yet, the proposed method of this paper shows that an improved heuristic approach can yield high-quality results comparable to the results by metaheuristics (i.e., results in Section 6.2) due to the use of the two-mode coefficient.
- (iii) Need optimization model? this is a practical issue concerning the skills required by users to apply the relevant algorithms. Notably, the quality of an

TABLE 3: Comparison of algorithmic approaches to CF problems.

	Solution search strategy	Guarantee of optimal solutions	Need optimization model?	Computational time
Exact solution approach	Branch-and-bound approach for exhaustive search	Yes	Yes	Long, not feasible for a large problem
Metaheuristics	Systematic diversification and intensification	No, good potential for high-quality solutions	Yes	Medium
Heuristics	Greedy and hill-climbing strategy, short-term results without backtracking	No, fair potential for high-quality solutions	No	Short
Proposed method (improved heuristics)	Same as heuristics	No, comparable to metaheuristics	No	Short, example: 0.5 second for 40×100 matrices

optimization model is essential to implement metaheuristics for process convergence and high-quality solutions. Arguably, similarity should be an easier concept, where users can focus on the pairwise relations between two objects. The consideration of production factors can be aggregated individually in a nonbinary matrix (i.e., (6)).

- (iv) Computational time: the exact solution approach would take long computational time, and it is not practical for large problems. For example, Brusco [6, page 662] has stated that they did not test the exact solution approach for a problem with the total number of machines and parts equal to or more than 65. Concerning metaheuristics, we have checked the times required to solve the problems in Section 6.2 from Solimanpur et al. [9], where the times range from 0.87 second (for P6) to 106.94 seconds (for P3). In contrast, as reported in Section 6.1, the proposed method takes 0.5 second for the problem with 40 machines and 100 parts. It verifies that the solution search by the proposed method (as one heuristics) takes short times.

In sum, the proposed method is considered an improved heuristic approach since it can yield solutions of comparable quality with the solutions by metaheuristics. At the same time, it can keep the benefit of short computational times in most heuristic approaches.

7. Conclusions

Although metaheuristics has been widely applied to solve CF problems, the development of SCM can still be found in recent literature due to its simplicity and efficiency to obtain reasonable solutions. By considering hierarchical clustering

as a heuristic approach (rather than a separate class of solution approaches), this paper tries to explore its capability to obtain high-quality solutions as compared to metaheuristics. In the proposed method, the nonbinary matrix is used to incorporate the information of part demands and operation sequences and the two-mode similarity coefficient to simultaneously group machines and parts. In the numerical experiments, we can demonstrate that the proposed method can yield better solutions than those by metaheuristics in some cases reported in literature.

In future work, it is planned to revisit different types of CF problems in literature based on the notion of the two-mode coefficient. The overall goal is to establish a general framework based on SCM that can tackle various types of CF problems and yield reasonable solutions for large problems in a limited time. By acknowledging the heuristic nature of SCM, we will also explore the limits of SCM to obtain high-quality solutions and develop some similarity-based guidance to improve the capability of the clustering heuristics.

Data Availability

The data related to this research is available upon request by contacting the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the NSERC Discovery Grants (Canada).

Supplementary Materials

One file of supplementary material is included with the manuscript. This file contains the images of the matrix solutions for the problems presented in Section 6.2. (*Supplementary Materials*)

References

- [1] J. McAuley, "Machine grouping for efficient production," *Production Engineering Research and Development*, vol. 51, no. 2, pp. 53–57, 1972.
- [2] Y. Yin and K. Yasuda, "Similarity coefficient methods applied to the cell formation problem: a taxonomy and review," *International Journal of Production Economics*, vol. 101, no. 2, pp. 329–352, 2006.
- [3] G. J. Nair and T. T. Narendran, "CASE: A clustering algorithm for cell formation with sequence data," *International Journal of Production Research*, vol. 36, no. 1, pp. 157–180, 1998.
- [4] G. Papaioannou and J. M. Wilson, "The evolution of cell formation problem methodologies based on recent studies (1997–2008): review and directions for future research," *European Journal of Operational Research*, vol. 206, no. 3, pp. 509–521, 2010.
- [5] C. Liu, Y. Yin, K. Yasuda, and J. Lian, "A heuristic algorithm for cell formation problems with consideration of multiple production factors," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 9–12, pp. 1201–1213, 2010.
- [6] M. J. Brusco, "An exact algorithm for maximizing grouping efficacy in part-machine clustering," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 47, no. 6, pp. 653–671, 2015.
- [7] M. J. Brusco, "An iterated local search heuristic for cell formation," *Computers & Industrial Engineering*, vol. 90, pp. 292–304, 2015.
- [8] S. Li and H. Mehrabadi, "Generation of block diagonal forms using hierarchical clustering for cell formation problems," in *Proceedings of the 47th CIRP Conference on Manufacturing Systems, CMS 2014*, pp. 44–49, can, April 2014.
- [9] M. Solimanpur, S. Saeedi, and I. Mahdavi, "Solving cell formation problem in cellular manufacturing using ant-colony-based optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 9–12, pp. 1135–1144, 2010.
- [10] V. Venugopal and T. T. Narendran, "Cell formation in manufacturing systems through simulated annealing: An experimental evaluation," *European Journal of Operational Research*, vol. 63, no. 3, pp. 409–422, 1992.
- [11] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: the Principles and Practice of Numerical Classification*, Freeman, San Francisco, 1973.
- [12] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.
- [13] H. Seifoddini and P. M. Wolfe, "Selection of a threshold value based on material handling cost in machine-component grouping," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 19, no. 3, pp. 266–270, 1987.
- [14] T. Gupta, "Design of manufacturing cells for flexible environment considering alternative routeing," *International Journal of Production Research*, vol. 31, no. 6, pp. 1259–1273, 1993.
- [15] C.-C. Chang, T.-H. Wu, and C.-W. Wu, "An approach for multiple attribute group decision making problems with interval-valued intuitionistic trapezoidal fuzzy numbers," *Computers & Industrial Engineering*, vol. 66, no. 2, pp. 438–450, 2013.
- [16] F. Choobineh, "A framework for the design of cellular manufacturing systems," *International Journal of Production Research*, vol. 26, no. 7, pp. 1161–1172, 1988.
- [17] A. J. Vakharia and U. Wemmerlov, "Designing a cellular manufacturing system: a materials flow approach based on operation sequences," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 22, no. 1, pp. 84–97, 1990.
- [18] R. G. Askin and M. Zhou, "Formation of independent flow-line cells based on operation requirements and machine capabilities," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 30, no. 4, pp. 319–329, 1998.
- [19] K. K. Goyal, P. K. Jain, and M. Jain, "A comprehensive approach to operation sequence similarity based part family formation in the reconfigurable manufacturing system," *International Journal of Production Research*, vol. 51, no. 6, pp. 1762–1776, 2013.
- [20] Y.-C. Ho, C.-E. C. Lee, and C. L. Moodie, "Two sequence-pattern, matching-based, flow analysis methods for multi-flowlines layout design," *International Journal of Production Research*, vol. 31, no. 7, pp. 1557–1578, 1993.
- [21] B. R. Sarker and Y. Xu, "Designing multi-product lines: Job routing in cellular manufacturing systems," *Institute of Industrial Engineers (IIE). IIE Transactions*, vol. 32, no. 3, pp. 219–235, 2000.
- [22] F. Alhourani, "Clustering algorithm for solving group technology problem with multiple process routings," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 781–790, 2013.
- [23] L. Wu and S. Suzuki, "Improved similarity coefficient method for cell formation with considering operation sequences and times," *Innovation and Supply Chain Management*, vol. 8, no. 2, pp. 48–57, 2014.
- [24] L. Wu and S. Suzuki, "Cell formation design with improved similarity coefficient method and decomposed mathematical model," *The International Journal of Advanced Manufacturing Technology*, vol. 79, no. 5–8, pp. 1335–1352, 2015.
- [25] Y. Won and S. Kim, "Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings," *Computers & Industrial Engineering*, vol. 32, no. 1, pp. 207–220, 1997.
- [26] Y. Yin and K. Yasuda, "Manufacturing cell's design in consideration of various production factors," *International Journal of Production Research*, vol. 40, no. 4, pp. 885–906, 2002.
- [27] Y. Won and K. C. Lee, "Group technology cell formation considering operation sequences and production volumes," *International Journal of Production Research*, vol. 39, no. 13, pp. 2755–2768, 2001.
- [28] Z. Yuguang, X. Kai, and S. Dongyan, "Clustering and group selection of interim product in shipbuilding," *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1393–1401, 2014.
- [29] F. Alhourani, "Cellular manufacturing system design considering machines reliability and parts alternative process routings," *International Journal of Production Research*, vol. 54, no. 3, pp. 846–863, 2016.
- [30] J. Navaei and H. ElMaraghy, "Grouping part/product variants based on networked operations sequence," *Journal of Manufacturing Systems*, vol. 38, pp. 63–76, 2016.
- [31] B. Boutsinas, "Machine-part cell formation using biclustering," *European Journal of Operational Research*, vol. 230, no. 3, pp. 563–572, 2013.
- [32] C. Renzi, F. Leali, M. Cavazzuti, and A. O. Andrisano, "A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems," *The*

- International Journal of Advanced Manufacturing Technology*, vol. 72, no. 1-4, pp. 403–418, 2014.
- [33] A. Kusiak, "The generalized group technology concept," *International Journal of Production Research*, vol. 25, no. 4, pp. 561–569, 1987.
 - [34] Y. Won and K. C. Lee, "Modified p-median approach for efficient GT cell formation," *Computers & Industrial Engineering*, vol. 46, no. 3, pp. 495–510, 2004.
 - [35] H. M. Selim, R. G. Askin, and A. J. Vakharia, "Cell formation in group technology: review, evaluation and directions for future research," *Computers & Industrial Engineering*, vol. 34, no. 1, pp. 3–20, 1998.
 - [36] F. M. Defersha and M. Chen, "A comprehensive mathematical model for the design of cellular manufacturing systems," *International Journal of Production Economics*, vol. 103, no. 2, pp. 767–783, 2006.
 - [37] Z. Zhu, R. B. Heady, and S. Reiners, "An efficient zero-one formulation of the cell formation problem," *Computers & Industrial Engineering*, vol. 28, no. 4, pp. 911–916, 1995.
 - [38] B. Elbenani and J. A. Ferland, "An exact method for solving the manufacturing cell formation problem," *International Journal of Production Research*, vol. 50, no. 15, pp. 4038–4045, 2012.
 - [39] I. Bychkov, M. Batsyn, and P. M. Pardalos, "Exact model for the cell formation problem," *Optimization Letters*, vol. 8, no. 8, pp. 2203–2210, 2014.
 - [40] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
 - [41] D. Sun, L. Lin, and R. Batta, "Cell formation using tabu search," *Computers & Industrial Engineering*, vol. 28, no. 3, pp. 485–494, 1995.
 - [42] S. Lee and H. P. Wang, "Manufacturing cell formation: A dual-objective simulated annealing approach," *The International Journal of Advanced Manufacturing Technology*, vol. 7, no. 5, pp. 314–320, 1992.
 - [43] R. Tavakkoli-Moghaddam, 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, pp. 443–454, 2008.
 - [44] C. Zhao and Z. Wu, "A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives," *International Journal of Production Research*, vol. 38, no. 2, pp. 385–395, 2000.
 - [45] F. M. Defersha and M. Chen, "A linear programming embedded genetic algorithm for an integrated cell formation and lot sizing considering product quality," *European Journal of Operational Research*, vol. 187, no. 1, pp. 46–69, 2008.
 - [46] A. Stawowy, "Evolutionary strategy for manufacturing cell design," *Omega*, vol. 34, no. 1, pp. 1–18, 2006.
 - [47] J. L. Burbidge, "Production flow analysis," *Production Engineer*, vol. 50, no. 4-5, pp. 139–152, 1971.
 - [48] J. R. King, "Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm," *International Journal of Production Research*, vol. 18, no. 2, pp. 213–232, 1980.
 - [49] C. Moore and S. Mertens, *The Nature of Computation*, Oxford University Press, New York, NY, USA, 2011.
 - [50] G. Harhalakis, R. Nagi, and J. M. Proth, "An efficient heuristic in manufacturing cell formation for group technology applications," *International Journal of Production Research*, vol. 28, no. 1, pp. 185–198, 1990.
 - [51] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.
 - [52] Y. Zhu, *Hierarchical Clustering and Similarity Statistics for Solving and Investigating Cell Formation Problems (MSc Thesis)*, University of Calgary, 2017, <http://dx.doi.org/10.5072/PRISM/26356>.
 - [53] H. Seifoddini, "Single linkage versus average linkage clustering in machine cells formation applications," *Computers & Industrial Engineering*, vol. 16, no. 3, pp. 419–426, 1989.
 - [54] L. Chen, Z. Ding, and S. Li, "Tree-based dependency analysis in decomposition and Re-decomposition of complex design problems," *Journal of Mechanical Design*, vol. 127, no. 1, pp. 12–23, 2005.
 - [55] J. F. Gonçalves and M. G. C. Resende, "An evolutionary algorithm for manufacturing cell formation," *Computers & Industrial Engineering*, vol. 47, no. 2-3, pp. 247–273, 2004.
 - [56] M. P. Chandrasekharan and R. Rajagopalan, "Zodiac—an algorithm for concurrent formation of part-families and machine-cells," *International Journal of Production Research*, vol. 25, no. 6, pp. 835–850, 1987.
 - [57] K. Spiliopoulos and S. Sofianopoulou, "An efficient ant colony optimization system for the manufacturing cells formation problem," *The International Journal of Advanced Manufacturing Technology*, vol. 36, no. 5-6, pp. 589–597, 2008.
 - [58] 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.
 - [59] W. J. Boe and C. H. Cheng, "A close neighbour algorithm for designing cellular manufacturing system," *International Journal of Production Research*, vol. 29, no. 10, pp. 2097–2116, 1991.
 - [60] L. E. Stanfel, "Machine clustering for economic production," *Engineering Costs and Production Economics*, vol. 9, no. 1-3, pp. 73–81, 1985.
 - [61] G. Srinivasan, "A clustering algorithm for machine cell formation in group technology using minimum spanning trees," *International Journal of Production Research*, vol. 32, no. 9, pp. 2149–2158, 1994.
 - [62] X. Wu, C.-H. Chu, Y. Wang, and D. Yue, "Genetic algorithms for integrating cell formation with machine layout and scheduling," *Computers & Industrial Engineering*, vol. 53, no. 2, pp. 277–289, 2007.
 - [63] C. C. Tsai and C. Y. Lee, "Optimization of manufacturing cell formation with a multi-functional mathematical programming model," *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 3, pp. 309–318, 2006.
 - [64] R. Sudhakara Pandian and S. S. Mahapatra, "Manufacturing cell formation with production data using neural networks," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1340–1347, 2009.

Submit your manuscripts at
www.hindawi.com