INT. J. PROD. RES., 1995, VOL. 33, NO. 2, 293–318

A neural network-based cell formation algorithm in cellular manufacturing

S.-J. CHEN† and C.-S. CHENG†‡

The Adaptive Resonance Theory (ART) neural network is a novel method for the cell formation problem in group technology (GT). The advantages of using an ART network over other conventional methods are its fast computation and the outstanding ability to handle large scale industrial problems. One weakness of this approach is that the quality of a grouping solution is highly dependent on the initial disposition of the machine-part incidence matrix especially in the presence of bottleneck machines and/or bottleneck parts. The effort of this paper has been aimed at alleviating the above mentioned problem by the introduction of a set of supplementary procedures. The advantages of the supplementary procedures are demonstrated by 40 examples from the literature. The results clearly demonstrate that our algorithm is more reliable and efficient in cases of ill-structured data.

1. Introduction

Group Technology (GT) is an important management philosophy for improving the productivity of manufacturing systems. The application of GT promises reduction in material handling cost, set-up time, work-in-process, and many others. Cellular manufacturing is a successful application of GT concepts. One of the first problems encountered in the development and implementation of a cellular manufacturing system is that of cell formation. Cell formation involves identifying families of similar parts and forming the associated machine cells such that one or more part families can be processed within a single cell. A part family consists of those parts requiring similar machine operations. Those machines used for manufacturing a particular part family form a machine cell. In the past several years, numerous methods have been developed to identify part families and their associated machine cells. Generally, these methods can be classified as classification and coding procedures or direct analysis of process information. In this paper, we are concerned with the latter approach. The machine cells formation problem based on process information is often modelled by a binary machine-part incidence matrix $\{a_{ij}\}$ derived from route card data. This approach is referred to as the matrix formulation of the GT problem. Columns and rows of an incidence matrix represent parts and machines, respectively. A matrix element a_{ij} is '1' if machine i is used to process part j, and '0' if otherwise.

Once the machine and part incidence matrix is constructed from route card data, a clustering algorithm is often required to transform the initial matrix into a solution matrix to help identify clusters. Numerous algorithms for the construction of

Revision received May 1994.

†Department of Industrial Engineering, YUAN-ZE Institute of Technology, 135 Yuan-Tung Rd., Nei-Li, Taiwan, Republic of China.
‡To whom correspondence should be addressed.

machine cells and part families have been developed using a machine-part incidence matrix. The grouping of parts and machines can be done sequentially or simultaneously. In the sequential approach (also known as indirect approach), the machines (or parts) are grouped first, and then parts (or machines) are assigned to their appropriate machine cells (or part families). Examples are the methods of McAuley (1972), Seifoddini and Wolfe (1986). The simultaneous approach (also called the direct approach or machine-component group analysis) tries to achieve the grouping in one step. Some typical examples are the direct clustering algorithm (DCA) due to Chan and Milner (1982), rank order clustering (ROC) of King (1980) and the cluster identification algorithm (CIA) of Kusiak and Chow (1987). A comparison of these two approaches to cell formation problem was given by Seifoddini (1990).

Recently, the use of artificial neural networks in cell formation has been examined by numerous researchers with verying degrees of success. The following neural network models have been used to solve the machine and/or part grouping problems: backpropagation network (Kao and Moon 1991, Kaparthi and Suresh 1991), self-organizing network (Lee *et al.* 1992), simulated annealing (Lee and Wang 1992, Venugopal and Narendran 1992), Adaptive Resonance Theory (ART) (Dagli and Huggahalli 1991, Kusiak and Chung 1991, Burke and Kamal 1992, Dagli and Sen 1992, Kaparthi and Suresh 1992, Liao and Chen 1993), the Hebbian network (Malave and Ramachandran 1991), Grossberg's interactive activation and competitive network (Moon 1990), constraint satisfaction model (Moon and Chi 1992).

Among the approaches listed above, the ART network is found to be best suited for the cell formation problem. There are several variations of ART network, namely, ART1 (Carpenter and Grossberg 1986, 1988), ART2 (Carpenter and Grossberg 1987), and fuzzy ART (Carpenter *et al.* 1991). The ART1 can handle binary input patterns, while others can process both binary and analog.

The performance of the ART1 network has been investigated by several researchers (Dagli and Huggahalli 1991, Kusiak and Chung 1991, Dagli and Sen 1992, Kaparthi and Suresh 1992, Liao and Chen 1993). The most important advantage of the ART1 algorithm in cell formation is its extremely efficient computational performance. The approach was found to be appropriate for solving large-scale GT problems. The major disadvantage of using the ART1 algorithm in GT cell formation is that the order of presentation affects the performance to a large degree. Dagli and Huggahalli (1992) suggested several heuristic methods to overcome the limitations of ART1. The first method modified the weight update of ART1. Second, they presented inputs in decreasing order of the number of 1's. Kaparthi and Suresh (1992) provided several important comments about implementing the ART1 algorithm in cell formation. First, the matrix density (proportion of '1' elements) tends to be rather low. It is advantageous to reverse the ones and zeros in the incidence matrix for clustering purposes, and to restore the original values after clustering. Second, in cases of ill-structured data, different order of presentations might lead to different results. The authors suggested that imperfect data may require several presentations in the form of unsupervised-supervised learning. Burke and Kamal (1992) investigated the application of the fuzzy ART to GT problems. The performance of the fuzzy ART is dependent on the choice of network parameters.

The purpose of this paper is to develop a set of supplementary procedures used to overcome the above mentioned problems associated with the ART1 algorithm. The specific aim of this study is to improve the performance of ART1 in the presence of ill-structured data. In this context, an ill-structured data set refers to an incidence matrix that contains exceptional elements (EE) (i.e., elements not in the machine/part groups). Exceptional elements are attributed to bottleneck machines and/or bottleneck parts (Shafer *et al.* 1992). In the cell formation literature, a bottleneck machine is defined as a machine that processes parts belonging to more than one family (Chu and Tsai 1990, Cheng 1992, Kusiak and Cho 1992). A bottleneck part is one that is processed on machines belonging to two or more machine cells (Frazier and Gaither 1991, Cheng 1992, Kusiak and Cho 1992). The organization of this paper is as follows. Section 2 presents a detailed procedure of the ART1 neural network. Section 3 describes the supplementary procedures used to enhance the ART1. Section 4 describes the performance measures used to evaluate the cell formation algorithm. Section 5 presents the results of applying the extended ART1 on test problems from the literature. This is followed by the conclusions in §6.

2. Adaptive resonance theory

An artificial neural network is built on a number of simple processing elements called neurons. These neurons are often organized into a sequence of layers. All layers of the network are linked by weights, which are adapted using a learning algorithm. The structure of a neural network could be characterized by the interconnection architecture among neurons, the activation function for conversion of inputs into outputs, and the learning algorithm. There are a variety of different structures and learning algorithms useful for neural network application. For a discussion of the basic principles of neural networks, the reader is referred to Lippmann (1987), Hush and Horne (1993).

A number of neural network paradigms can be used as classifiers for the cell formation problem. In this study we are concerned with ART1, which is found to be well suited for the cell formation problem based on binary machine-part incidence matrix. The ART1 neural network is based on unsupervised learning. Learning in neural networks can be supervised, unsupervised or based on a combined unsupervised-supervised learning. In supervised learning, the correct output for an input pattern has to be specified when the input pattern is presented. In an unsupervised learning, the network has no knowledge about what the correct or desired outputs should be. The system learns on its own without external guidance. In the cell formation problem, unsupervised learning is more appropriate, because in practice, no information about correct group formation is known *a priori*.

The network includes two layers of neurons: the input (comparison) layer and the output (recognition) layer. The comparison layer elements accept inputs from the environment and the recognition layer elements each represents a pattern class. Every node in the input layer is totally connected to every node in the output layer with top-down and bottom-up connections. The ART1 algorithm employs a competitive learning approach in the sense that ART1 learns to cluster the input patterns by making the output neurons compete with each other for the right to react to a particular input pattern. The output neuron which has the weight vector that is most similar to the input vector claims this input pattern by producing an output of '1' and at the same time inhibits other output neurons by forcing them to produce 0's. In ART1, only the winning node is permitted to alter its weight vector, which is modified in such a way that it is brought even near to the representative input pattern in the cluster concerned. ART1 attempts to associate an input pattern to a cluster of patterns. The output of ART1 is an indication of membership of the input pattern in a group with similar characteristics.

The ART algorithm could be applied in two different ways in GT cell formation. In the first approach, ART is applied to machine vectors as well as part vectors to form machine cells and part families (Dagli and Huggahalli 1991, Liao and Chen 1993). With this approach an appropriate parameter must be selected such that the number of cells equals the number of part families. In the second approach (Kaparthi and Suresh 1992), machine vectors (part vectors) are presented to ART to form machine cells (part families). Next, parts (machines) are assigned to their appropriate machine cells (part families) based on some heuristic methods. In this paper, ART1 network is applied in the manner of the second approach.

To begin with, we must determine the size of the input and output layers. For the cell formation problem, either the part or machine characteristic vectors could be the set of input patterns presented to the input layer. The output nodes are modelled as machine cells or part families, each output unit representing a machine cell or a part family. In practice, the number of part types is generally much greater than the number of machine types. Using part characteristic vectors as inputs has an advantage that it involves fewer memory requirements. However, the neural network has to perform cluster analysis based on a smaller number of features. Throughout this paper, the analysis is for the set of machine characteristic vectors as input to the network unless otherwise specified. The input neurons are modelled as the characteristic of the machines, the number of neurons required in the input layer is set equal to the total number of parts. Output of neurons in the output layer represents the class of the input vector. Hence the number of neurons required in the output layer corresponds to the maximum expected number of machine cells.

The steps to implement ART1 algorithm for cell formation problem are as follows.

Step 0 Define the number of neurons in the input layer N_{in} and the number of neurons in the output layer N_{out} and select a value (between 0 and 1) for the vigilance parameter, ρ.
N_{in} = the number of columns (parts) of machine-part incidence matrix.
N_{out} = the maximum expected number of machine cells.

Step 1 Enable all the output units and initialize top-down weights W^t and bottom-up weights W^b.

$$W^t_{ij} = 1$$

$$W^b_{ij} = \frac{1}{1 + N_{\text{in}}}$$

W^t_{ij} = top-down weight from neuron j in the output layer to neuron i in the input layer.
W^b_{ij} = bottom-up weight from neuron i in the input layer to neuron j in the output layer.

Step 2 Present a machine vector X to the input layer, X consists of zero/one element x_i.

Step 3 Compute matching scores for all the enabled output nodes.

$$net_j = \sum_i W^b_{ij} \cdot x_i$$

where net_j is the output of neuron j in the output layer.

Step 4 Select a node with the largest value of matching score as best matching exemplar, let this node be j^*. In the event of a tie, the unit on the left is selected.

$$net_{j*} = \max_j \{net_j\}$$

Step 5 Perform vigilance test to verify that input pattern X belongs to cluster (cell) j^*.

$$||X|| = \sum_i x_i \text{ (norm of vector } X\text{)}$$

$$||W^t_{j*} \cdot X|| = \sum_i W^t_{ij*} \cdot x_i$$

$$V_{j*} = \frac{||W^t_{j*} \cdot X||}{||X||}$$

if $V_{j*} > \rho$, there is resonance, go to step 7. Otherwise, the cluster exemplar is rejected, go to step 6.

Step 6 Disable best matching exemplar
Since the vector X does not belong to cluster j^* the output of node j^* selected in step 4 is temporarily disabled and removed from future competitions; go to step 3.

Step 7 Adapt best matching exemplar

$$W^t_{ij*} = W^t_{ij*} \cdot x_i \qquad \text{(logical AND operation)}$$

$$W^b_{ij*} = \frac{W^t_{ij*} \cdot x_i}{0{\cdot}5 + \sum W^t_{ij*} \cdot x_i}$$

Step 8 Enable any nodes disabled in step 6 and go to step 2.

Some important features of the ART1 algorithm are discussed below. The weights updating procedures constitute the learning process in ART1 algorithm. The algorithm described here implements a fast-learning rule for the weights in the sense that any 1's that are not in the input pattern are removed from the exemplar template. It should be noted that the noisy patterns will degrade the exemplar template due to the logical AND operation performed during updating. The learning process of ART1 is very efficient when compared to other algorithms. In most algorithms, the input patterns have to be presented repeatedly until the weights stabilize to fixed values or a certain number of training iterations has been reached.

The operations involved in step 5 can be thought of as a template matching procedure. The similarity measure quantifies the degree of match between the input pattern and the exemplar template. The amount of mismatch tolerated between input pattern and exemplar is determined by the vigilance parameter ρ. The vigilance parameter functions as a similarity threshold. An ART1 with large vigilance will permit a small amount of mismatch and will result in a large number of separate clusters. On the other hand, if the value of ρ is set low, the patterns might be organized into a small number of clusters. In the similarity coefficient-based algorithms, a similarity threshold of zero implies a solution of one cluster. This might not be true for the ART1 algorithm. Unless the number of output node is set to one, the ART1 might not yield a solution of one cluster even with a ρ value of zero. This is due to the fact that an uncommitted node may win over an existing cluster during the search of the largest matching score.

As with the similarity coefficient method, ART1 could create a number of alternative solutions by simply adjusting the vigilance parameter. This feature increases the flexibility of designing manufacturing cells. However, the vigilance parameter is utilized during the learning procedure. A change of vigilance value implies that the ART1 algorithm has to be reiterated.

The ART1 algorithm described above is used as a clustering method to group machines into machine cells based on their manufacturing similarity. An ancillary part-assignment algorithm is needed to assign parts to their appropriate machine cells. In this research, the part-assignment algorithm proceeds as follows. For each part, find a machine cell which processes the part for a larger number of operations than any other machine cell. Ties are broken by choosing the machine cell which has the largest percentage of machines visited by that part. In the case of a tie again, the machine cell with the smallest identification number is selected. As will be shown later, reversing zeros and ones of the initial machine-part matrix often leads to a better solution. It is noted that before applying the part-assignment algorithm, zeros and ones should be restored if they were reversed previously.

3. Supplementary procedures

It was observed that ART1 algorithm performs well when the original matrix is well-structured. However, ART1 has limitations which become apparent when the method is applied to a grouping problem with an ill-structured incidence matrix. Before presenting the supplementary procedures, the potential problems with the ART1 are discussed below.

The limitations of the ART1 algorithm are illustrated with the example matrix presented in Fig. 1 (*a*). Figure 1 (*b*) shows the solution matrix after rearranging rows and columns. Three machine cells (clusters) $MC\text{-}1=\{1,2,3,9,10\}$, $MC\text{-}2=\{4,5\}$ and $MC\text{-}3=\{6,7,8\}$, and three corresponding part families $PF\text{-}1=\{1,2,6\}$, $PF\text{-}2=\{3,7,8\}$ and $PF\text{-}3=\{4,5,9,10\}$ are visible in the clustered matrix shown in Fig. 1 (*b*). The solution matrix shows bottleneck machines 9 and 10 with six exceptional elements.

Figure 1 (*c*) is the solution matrix when the ART1 algorithm ($\rho=0{\cdot}25$) is applied to the example matrix in Fig. 1 (*a*). The algorithm gives three clusters and eight exceptional elements. Three machine cells, $MC\text{-}1=\{1,2,3\}$, $MC\text{-}2=\{4,5,10\}$, and $MC\text{-}3=\{6,7,8,9\}$ and three corresponding part families $PF\text{-}1=\{1,2,6\}$, $PF\text{-}2=\{3,7,8\}$ and $PF\text{-}3=\{4,5,9,10\}$ are shown in Fig. 1 (*c*). The difference between solutions in Fig. 1 (*b*) and Fig. 1 (*c*) is the assignment of bottleneck machines 9 and 10. For the same number of clusters, the result of Fig. 1 (*c*) has more exceptional elements than that in Fig. 1 (*b*). The clustering result of Fig. 1 (*c*) is a typical example of the improper machine assignment problem identified by Seifoddini (1986, 1989 a). One can see that machines 9 and 10 have more common operations with machines in the first cell than with machines in other cells. They should be assigned to the first cell in order to reduce the number of exceptional elements.

In the ART1 algorithm, the improper assignment of bottleneck machines may be partially attributed to the decayed template. The matrix shown in Fig. 1 (*a*) can be used to illustrate this point. For ease of illustration, the number of clusters has been set at three. The first eight machines are classified into three machine cells. The connection weights after applying the first eight machine vectors are as follows.

	i	W^t_{ij} 1	2	3	4	5	6	7	8	9	10
j	1	1	0	0	0	0	0	0	0	0	0
	2	0	0	1	0	0	0	0	1	0	0
	3	0	0	0	1	0	0	0	0	1	0

	i	W^b_{ij} 1	2	3	4	5	6	7	8	9	10
j	1	0·67	0	0	0	0	0	0	0	0	0
	2	0	0	0·4	0	0	0	0	0·4	0	0
	3	0	0	0	0·4	0	0	0	0	0·4	0

Applying machine 9, the matching scores are

$$net_1 = 0{\cdot}67(1)+0(1)+0(1)+0(1)+0(0)+0(1)+0(1)+0(0)+0(1)+0(0)=0{\cdot}67$$
$$net_2 = 0(1)+0(1)+0{\cdot}4(1)+0(1)+0(0)+0(1)+0(1)+0{\cdot}4(0)+0(1)+0(0)=0{\cdot}4$$
$$net_3 = 0(1)+0(1)+0(1)+0{\cdot}4(1)+0(0)+0(1)+0(1)+0(0)+0{\cdot}4(1)+0(0)=0{\cdot}8$$

In step 4 of the ART1 algorithm, cluster 3 gets selected due to its highest matching score. Because $V_3 = 2/7 > 0{\cdot}25$, machine 9 is considered as belonging to the third cluster. Note that the connection weights are not changed after presenting machine 9. The matching scores for machine 10 can be computed as

$$net_1 = 0{\cdot}67(1)+0(1)+0(1)+0(0)+0(0)+0(1)+0(0)+0(1)+0(0)+0(0)=0{\cdot}67$$
$$net_2 = 0(1)+0(1)+0{\cdot}4(1)+0(0)+0(0)+0(1)+0(0)+0{\cdot}4(1)+0(0)+0(0)=0{\cdot}8$$
$$net_3 = 0(1)+0(1)+0(1)+0{\cdot}4(0)+0(0)+0(1)+0(0)+0(1)+0{\cdot}4(0)+0(0)=0{\cdot}0$$

The results indicate that the second cluster is the winner. Since $V_2 = 2/5 > 0{\cdot}25$, the ART1 algorithm assigns machine 10 to cluster 2. Further reduction of the vigilance parameter does not change the results of clustering. Although machines 9 and 10 have more common operations with machine 1 in the first cell than with machines in other cells, they will not be categorized under the first cell. This is due to the fact that there is only one bit left in the template of the first cluster after introducing machine 2. If machines 9 and 10 were presented before machine 2, the ART1 algorithm gives the clustering result as shown in Fig. 1 (*b*).

The above example shows that the order of presentation has great influence on how the exemplar template is created and modified. This implies that the sequence of presentation affects the final solution to some extent. The problem bcomes more severe when many input vectors containing few 1's. This comes essentially from the fact that the learning rule can remove bits from the exemplar without adding any. Thus the exemplar can never cycle back to a previous value.

In some cases, the limitations of the ART1 algorithm can be overcome by reversing the zeros and ones of the machine-part incidence matrix, as suggested by Kaparthi and Suresh (1992). As an example, consider the machine-part incidence matrix in Fig. 2 (*a*). Three machine cells are identified by ART1, MC-$1 = \{1,2,3\}$,

(a)

Machines \ Parts	01	02	03	04	05	06	07	08	09	10
1	1	1	0	0	0	1	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0	0
5	0	0	1	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	1	0
7	0	0	0	1	1	0	0	0	1	1
8	0	0	0	1	1	0	0	0	1	1
9	1	1	1	1	0	1	1	0	1	0
10	1	1	1	0	0	1	0	1	0	0

(b)

Machines \ Parts	01	02	06	03	07	08	04	05	09	10
1	1	1	1							
2	1									
3	1		1							
9	1	1	1	1	1		1		1	
10	1	1	1	1		1				
4				1	1	1				
5				1		1				
6							1		1	
7							1	1	1	1
8							1	1	1	1

(c)

Machines \ Parts	01	02	06	03	07	08	04	05	09	10
1	1	1	1							
2	1									
3	1		1							
4				1	1	1				
5				1		1				
10	1	1	1	1		1				
6							1		1	
7							1	1	1	1
8							1	1	1	1
9	1	1	1	1	1		1		1	

Figure 1. Example A, (*a*) the initial matrix, (*b*) a possible solution and (*c*) the solution matrix by the ART1 algorithm.

(a)

Machines \ Parts	01	02	03	04	05	06
1	0	1	0	1	0	1
2	0	0	0	0	1	1
3	0	1	0	0	1	1
4	1	0	1	0	0	0
5	1	0	1	0	0	0
6	0	1	0	1	1	0

(b)

Machines \ Parts	02	05	06	01	03	04
1	1		1			1
2		1	1			
3	1	1	1			
4				1	1	
5				1	1	
6	1	1				1

(c)

Machines \ Parts	02	04	05	06	01	03
1	1	1		1		
2			1	1		
3	1		1	1		
6	1	1	1			
4					1	1
5					1	1

Figure 2. Example B, (*a*) the initial matrix, (*b*) solution matrix obtained by ART1 without reversing the zeros and ones and (*c*) solution matrix obtained by ART1 with reversing the zeros and ones.

MC-2 = {4,5}, and MC-3 = {6}. It is clear that machine 6 should be assigned to the first cell in order to reduce the number of exceptional elements. However, due to the decay of the template, machine 6 could not be assigned to the second cell (Fig. 2 (*b*)). The above problem can be resolved by reversing the zeros and ones (Fig. 2 (*c*)). It should be noted that reversing the zeros and ones does not always lead to a reasonable solution. This can be shown by example in Fig. 3 (*a*). After reversing the zeros and ones, ART1 produces two machine cells, namely MC-1 = {1,2,3,4}, and MC-2 = {5,6,7,8} (Fig. 3 (*b*)). The resulting matrix is not acceptable because machine 1 does not process any parts in the first machine cell. This example illustrates that merely reversing the zeros and ones may cause two machines to join together (based on the reversed 1's) even though they have no elements in common. This problem arises when input vectors containing few 1's (before reversing) are presented first.

In acknowledging the limitations of the ART1 algorithm in cases of ill-structured data, we propose a set of supplementary procedures to enhance the original algorithm, specifically for cell formation problems.

The first supplementary procedure is referred to here as rearrangement. Rearrangement is a procedure used to guide the presentation of input vectors. As was noted earlier, the top-down template is updated by removing 1's from it without adding any. The decay of the template becomes crucial when input patterns are noisy. The rearrangement process is used to provide a smooth weights change. The procedure for rearranging the machine-part incidence matrix is outlined in the following algorithm.

3.1. *Rearrangement algorithm*

Step 0 Apply the basic ART1 algorithm and the part-assignment method described above to identify machine/part groups.

Step 1 Identify those machine vectors with the number of '1' entries within the machine/part group smaller than the number of '1' entries remain outside the machine/part group and set them aside.

Step 2 For each cell, rearrange the machine vectors with less 1's outside in descending order of the number of 1's in the machine/part group. In the

Machines	01	02	03	04	05	06	07	08	09	10
1	0	0	0	0	0	0	1	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	1	0
4	1	0	1	1	0	0	0	0	1	0
5	0	1	0	0	1	0	0	1	0	1
6	0	0	0	0	1	1	1	0	0	1
7	0	1	0	0	1	1	0	1	0	0
8	0	1	0	0	1	1	0	1	0	1

(*a*)

Machines	01	03	04	09	02	05	06	07	08	10
1							1			
2	1									
3	1	1		1						
4	1	1	1	1						
5					1	1			1	1
6						1	1	1		1
7					1	1	1		1	
8					1	1	1		1	1

(*b*)

Figure 3. Example C, (*a*) the initial matrix and (*b*) solution matrix obtained by ART1 with reversing the zeros and ones.

Machines	01	02	03	04	05	06	07	08	09	10
1	0	0	0	0	0	1	1	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	1	0
4	1	0	1	1	0	0	0	0	1	0
5	0	1	0	0	1	0	0	1	0	1
6	0	0	0	0	1	1	1	0	0	1
7	0	1	0	0	1	1	0	1	0	0
8	0	1	0	0	1	1	0	1	0	1

(*a*)

Machines	01	03	04	07	09	02	05	06	08	10
1				1				1		
2	1									
3	1	1			1					
4	1	1	1		1					
5						1	1		1	1
6				1			1	1		1
7						1	1	1	1	
8						1	1	1	1	1

(*b*)

Machines	01	03	04	09	07	02	05	06	08	10
4	1	1	1	1						
3	1	1		1						
2	1									
1					1			1		
8						1	1	1	1	1
5						1	1		1	1
7						1	1	1	1	
6					1		1	1		1

(*c*)

Figure 4. Example D, (*a*) the initial matrix, (*b*) solution matrix obtained by ART1 without rearrangement and (*c*) solution matrix obtained by ART1 with rearrangement.

case of a tie, the machine with smaller number of 1's outside is presented first. In the case of a tie again, the machine with the smallest identification number is presented first.

Step 3 Present the machine vectors that were set aside in step 1 in descending order of the number of '1' entries. In the case of a tie, the machine with the smallest identification number will be presented first.

With above procedure similar machines are brought together and then presented to the ART1 network. The final order of presentation is machine vectors with fewer 1's outside, in descending order of the number of 1's in the machine group, then vectors with more 1's outside, in descending order of the number of 1's. The advantage of rearrangement procedure can be illustrated with machine-part incidence matrix shown in Fig. 4 (*a*). If reversing zeros and ones is carried out, ART1 gives two machine cells, $MC\text{-}1=\{1,2,3,4\}$, and $MC\text{-}2=\{5,6,7,8\}$ with EE = 2 (Fig. 4 (*b*)). Applying rearrangement procedure results in two machine cells, $MC\text{-}1=\{2,3,4\}$, and $MC\text{-}2=\{1,5,6,7,8\}$ with EE = 0 (Fig. 4 (*c*)). The improvement in solution is due to the sequence of presenting machine vectors as indicated in Fig. 4 (*c*).

An efficient alternative to the rearrangement procedure is presenting input vectors in descending order of the number of 1's as suggested by Dagli and Huggahalli (1991). The idea behind this approach is to prevent the exemplar from growing sparser as more inputs are applied. However, this approach does not guarantee a satisfactory result. The problem consisting of 15 machines and 15 parts, as shown in Fig. 5(*a*), is employed here to demonstrate that rearrangement works beter than simply presenting input vectors in descending order of the number of 1's. As can be seen from Fig. 5(*b*), the solution obtained by presenting machines in descending order of the number of 1's is not accceptable. The smallest number of machine cells identified is five, MC-1 = {1,2,4}, MC-2 = {5, 6, 7}, MC-3 = {8,9,10,11}, MC-4 = {3,12,13,15}, MC-5 = {14} with EE = 11. Applying the rearrangement procedure results in four machine cells, MC-1 = {5, 6, 7}, MC-2 = {2, 4, 14}, MC-3 = {3,11,12,13} and MC-4 = {1,8,9,10,15} with EE = 7 (Fig. 5(*c*)). The results suggested that the solution derived using the proposed method dominates the solution obtained by presenting inputs in descending order of the number of 1's.

Instead of merely counting the number of 1's, the rearrangement procedure considers the relative positioning of the 1's (based on initial grouping results). This explains why the rearrangement works better than presenting input vectors in descending order of the number of 1's.

The second supplementary procedure is called reassignment. Due to the decay of the exemplar template, the ART1 algorithm might not be able to bring the most similar machines together. This situation generally occurs in the presence of bottleneck machines. The reassignment procedure is used to refine the clustering result. The idea behind the procedure is that bottleneck machines should be reexamined and be reassigned to proper cells in order to reduce the number of exceptional elements. The algorithm includes the following steps.

3.2. *Reassignment algorithm*

Step 0 Apply the basic ART1 algorithm (and the rearrangement procedure if needed) together with the part-assignment algorithm to identify machine/part groups.

Step 1 Identify those machine vectors with the number of '1' entries within the machine/part group smaller than or equal to the number of '1' entries outside the machine/part group.

Step 2 Assign those machines identified in step 1 to their appropriate machine cells such that the number of exceptional elements is minimized. To do so, the number of parts in each group processed by each machine is determined (i.e., counting the number of 1's within machine/part groups) and the machine is assigned to the cell which has the largest number of parts processed by that machine. In the case of a tie, select the machine cell which processes the smallest number of parts. In the case of a tie again, the machine cell with the smallest identification number is selected.

Step 3 Delete those machine/part groups, which are empty.

Step 4 Assign each part to the appropriate machine cell using the part-assignment algorithm described in §2.

Note that the composition of the clusters might be changed after applying the reassignment procedure, therefore the procedure can iterate several times in order to yield a satisfactory result. Based on the experience of randomly generated problems of various sizes we found that the best solutions are usually obtained after two iterations of reassignment procedure.

4. Measures of performance

There are two measures frequently used in the literature to evaluate the quality of solutions given by a cell formation algorithm. The first one is the measure of effectiveness (ME) introduced by McCormick *et al.* (1972). The measure is defined as follows:

$$ME=\frac{1}{2}\cdot\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\cdot(a_{i,j-1}+a_{i,j+1}+a_{i-1,j}+a_{i+1,j})$$

where a_{ij} is the element of a clustered matrix. Generally speaking, a better clustering algorithm results in a higher ME. One major problem with the ME is that the relative positioning of rows and columns within a machine/part group has a great impact on the final result. The same grouping solutions (in terms of the cell composition) with different arrangements of rows and columns may give different values of the measure.

A better measure is the grouping efficiency (GE) proposed by Chandrasekharan and Rajagopalan (1986). This measure has been widely used in the literature (e.g., Askin *et al.* 1991, Boe and Cheng 1991, Kusiak and Cho 1992). GE is an aggregate measure which takes both the number of exceptional elements and machine utilization into consideration. The '1' elements remain outside the machine/part groups are often referred to as exceptional elements. Exceptional elements are the sources of intercellular moves and should be reduced to the minimum. Machine utilization indicates the percentage of times the machines within the cell are used in production. As a general rule, the higher the machine utilization, the better the clustering results. GE is the weighted average of these two components, it can be computed as:

$$\eta=q\eta_1+(1-q)\eta_2$$

where

$$\eta_1=\frac{e_d}{\sum_{r=1}^{k} M_rN_r}$$

$$\eta_2=1-\left[\frac{e_0}{mn-\sum_{r=1}^{k} M_rN_r}\right]$$

- m number of machines (rows)
- n number of parts (columns)
- M_r number of machines in the rth cell
- N_r number of parts in the rth family
- e_d number of 1's within the machine/part groups
- e_0 number of 1's outside the machine/part groups
- k number of clusters
- η grouping efficiency
- q weighting factor $(0\leq q\leq 1)$

GE ranges from 0 to 1. A GE with a value close to 1·0 means that the solution matrix has a perfect structure. Since most results reported in the literature were based on $q=0{\cdot}5$ as recommended by Chandrasekharan and Rajagopalan (1986), we will follow this convention to have a fair comparison with the existing algorithms. In this paper, the quality of solutions was evaluated in terms of GE and EE. When the GE measure is the same for two different solutions to a problem, the solution with lower exceptional elements is regarded as the better solution. This comparison approach has been accepted by most researchers (Chu and Tsai 1990, Boe and Cheng 1991, Kusiak and Cho 1992).

5. Computational results

In order to demonstrate that the ART1 with supplementary procedures is capable of yielding good solutions of cell formation problems, 40 data sets from the literature have been collected for the evaluation. The problems differ in size and density of machine-part incidence matrix. For ease of discussion, we divide the test data into three different groups: (1) problems that can be solved by the basic ART1 as well as the modified ART1 procedure; (2) problems that can be solved by the modified ART1 only; (3) problems for which the modified ART1 gives better results than reference algorithms. Details for each set are presented in Table 1.

Hereafter, we will refer to the ART1 with supplementary procedures as the extended ART1. For ease of reference, the extended ART1 will be denoted as ART1(a, b, c). For example, ART1(1,1,2) corresponds to reversing the zeros and ones followed by rearrangement and two iterations of reassignment. ART1(1, 0, 0) is equivalent to the algorithm proposed by Kaparthi and Suresh (1992). The ART1 algorithm applied in its basic form is denoted as ART1(0,0,0). It must be noted that the maximum number of iterations for each supplementary procedure is restricted to two. This comes from the fact that further iterations will not change the results for the 40 problems studied here. The results of applying the extended ART1 to the test problems are presented in Table 2. Data were input to the network in the same order as appeared in the literature. The best clustering results in terms of GE and EE from other studies are also included for comparison purposes. Each entry in Table 2 is the largest vigilance parameter at which ART1 can find a solution identical to the one shown on the left. Note that the vigilance parameter decreases from 0·99 to 0·0, with a decrement of 0·01. If ART1 cannot produce the solutions reported in the literature, it will be marked with '-'. Since the clustering results are dependent on the order of presentation, a '-' mark only indicates that ART1 cannot duplicate the reference solutions for that particular order of presentation. One should be aware that the grouping efficiency is highly dependent on the number of clusters produced by a cell formation algorithm. In order to have a fair comparison, we only consider the cases for which the ART1 produces the same number of clusters to that reported in the literature. In the following paragraphs, we review the solutions produced by the basic ART1 and the extended ART1 algorithms.

Table 2 (a) provides the grouping results for data sets in group I. The analysis of our computational results indicates that ART1(0,0,0) was able to duplicate the reference solutions without any difficulties. Comparing the results of ART1 algorithm with that of the extended ART1 shows that the extended ART1 algorithm could generally produce the best results at higher values of ρ.

Table 2 (b) summarizes the computational results for data sets in group II. Problems 25 and 28 are examples in the literature used to illustrate the improper assignment of bottleneck machines. In these problems, the basic ART1 could not exactly duplicate the best results reported in the literature. From Table 2 (b) one can see that the limitation of the basic ART1 can be overcome by applying reassignment algorithm.

Problem 32 was solved by Kumar and Vannelli (1987) with the constraints pertaining to the number of cells and cell size. ART1 cannot create the same number of cells as those of Kumar and Vannelli. This is because the ART1 does not set out to accommodate any constraints on the characteristics of the resultant cells. If the number of output nodes was initially set to 3, the solution of the extended ART1 compares exactly with the result obtained from Kumar and Vannelli (1987). The result from ART1(1,1,2) produces a higher value of GE than that reported in Kaparthi and Suresh (1992). Table 3 compares the alternative solutions obtained by ART1(1,0,0) and ART1(1,1,2).

Machines \ Parts	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
1			1	1	1		1			1	1	1	1	1	1
2	1	1										1			
3									1				1		
4		1								1		1	1		
5			1	1										1	
6				1		1								1	
7			1			1								1	
8					1		1				1				
9							1				1				1
10					1		1				1				
11							1				1				1
12								1					1		
13								1	1				1		
14	1	1													
15								1	1				1		

(*a*)

Machines \ Parts	02	10	12	03	04	06	14	05	07	11	15	08	09	13	01
1		1	1	1	1		1	1	1	1	1			1	
4	1	1	1											1	
2	1		1												1
5				1	1		1								
6					1	1	1								
7				1		1	1								
8								1	1	1					
9									1	1	1				
10								1	1	1					
11									1	1	1				
13												1	1	1	
15												1	1	1	
3													1	1	
12												1		1	
14	1														1

(*b*)

Machines \ Parts	03	04	06	14	01	02	10	12	08	09	13	05	07	11	15
5	1	1		1											
6		1	1	1											
7	1		1	1											
2					1	1		1							
4						1	1	1			1				
14					1	1									
11									1	1	1				
13									1	1	1				
3										1	1				
12									1		1				
8												1	1	1	
9													1	1	1
10												1	1	1	
15													1	1	1
1	1	1		1			1	1			1	1	1	1	1

(*c*)

Figure 5. Example D, (*a*) the initial matrix, (*b*) solution matrix obtained by ART1 in descending order of the number of 1's and (*c*) solution matrix obtained by ART1 with rearrangement.

The advantage of reversing the zeros and ones can be illustrated through the computational results of problem 33. It can be seen that ART1 without reversing the zeros and ones cannot produce an acceptable structure. The major reason is due to the extremely low density of the original incidence matrix. The number of clusters identified was too numerous to form machine cells. The usefulness of reversing zeros and ones depends on the characteristic of the machine-part matrix. For instance, ART1(1, 0, 0) performs well in problems 23 and 24, however, it encountered difficulties in problems 5, 8, 17 and 21.

It is worthy to point out that some of the solutions generated by the extended ART1 algorithm are of better quality than those provided by other algorithms. Table 2 (*c*) summarizes the computational results for test problems in group III. For problem 34 the solution by the extended ART1 resulted in the same number of machine cells and exceptional elements but higher GE than the solution in the literature. For problem 35, the solution of the extended ART1 is identical to the solution (with machine group size set equal to 4) given by Askin *et al.* (1991). Additionally, three alternative solutions by the extended ART1 resulted in the same number of machine cells but have lower exceptional elements and higher GE than the reference solution.

Harhalakis *et al.* (1990) gave two possible solutions for problem 36. In the case of 4 machine cells, Harhalakis *et al.* reported 11 exceptional elements with GE = 79·62. The extended ART1 yields the same number of exceptional elements but the GE is higher than that of Harhalakis *et al.* When the number of cells equals 5, Harhalakis *et al.* provided a solution with 15 exceptional elements and GE = 85·72. For the same number of cells, the extended ART1 produced 14 exceptional elements and GE = 88·96. The solutions to this problem are summarized in Table 5.

For problem 37, the extended ART1 gives four clusters, as in the solution provided by Srinivasan *et al.* (1990). However, the extended ART1 provided a better grouping efficiency and smaller number of exceptional elements than the solution given by Srinivasan *et al.* The extended ART1 identified four machine cells, $MC\text{-}1=\{1,4,7,8,11,12\}$, $MC\text{-}2=\{2,13\}$, $MC\text{-}3=\{5,10,14,16\}$ and $MC\text{-}4=\{3,6,9,15\}$ and the corresponding part families $PF\text{-}1=\{2,4,7,9,12,18,22,30\}$, $PF\text{-}2=\{1,3,10,13,16,20\}$, $PF\text{-}3=\{6,8,11,14,15,17,21,24,26\}$ and

Problem number	Source	Number of machines	Number of parts	Density
Group I				
1	King and Nakornchai (1982)	5	7	0·4
2	Waghodekar and Sahu (1984, Fig. 2(*a*))	5	7	0·457
3	Waghodekar and Sahu (1984, Fig. 3(*a*))	5	7	0·457
4	Kusiak and Cho (1992)	6	8	0·458
5	Kusiak and Chung (1991)	7	8	0·268
6	Kusiak (1992)	8	7	0·268
7	Seifoddini (1986)	5	12	0·5
8	Seifoddini and Wolfe (1986)	8	12	0·365
9	Seifoddini (1989 b, Fig. 1)	9	12	0·343
10	Seifoddini (1989 b, Fig. 2)	9	12	0·333
11	Seifoddini (1989 b, Fig. 3)	9	12	0·333
12	Askin *et al.* (1991)	10	15	0·327
13	Chan and Milner (1982, Fig. 2(*a*))	15	10	0·307
14	Chandrasekharan and Rajagopalan (1986)	8	20	0·381
15	Seifoddini (1989 c)	11	22	0·322
16	Stanfel (1985)	14	24	0·182
17	Burbidge (1975) excluding machines 6 and 18	14	43	0·145
18	Carrie (1973)	20	35	0·193
19	Burbidge (1975)	22	43	0·133
20	Chandrasekharan and Rajagopalan (1989)	24	40	0·136
21	Chandrasekharan and Rajagopalan (1989)	24	40	0·135
22	Chandrasekharan and Rajagopalan (1987)	40	100	0·105
Group II				
23	Waghodekar and Sahu (1984, Fig. 4(*a*))	5	7	0·571
24	Waghodekar and Sahu (1984, Fig. 5(*a*))	5	7	0·571
25	Chow and Hawaleshka (1992)	5	11	0·455
26	Kusiak and Chow (1987)	7	8	0·232
27	Gongaware and Ham (1981)	9	9	0·395
28	Seifoddini (1989 a, Table 1)	5	18	0·511
29	McAuley (1972)	12	10	0·317
30	Chan and Milner (1982, Fig. 3(*a*))	15	10	0·327
31	Chandrasekharan and Rajagopalan (1989)	24	40	0·136
32	Kumar and Vannelli (1987)	30	41	0·104
33	Stanfel (1985)	30	50	0·103
Group III				
34	Kusiak and Chow (1987)	7	11	0·3
35	de Witte (1980)	12	19	0·329
36	Harhalakis *et al.* (1990)	20	20	0·198
37	Srinivasan *et al.* (1990)	16	30	0·242
38	Burbidge (1975)	16	43	0·183
39	Burbidge (1975) (using part vectors as inputs)	16	43	0·183
40	Boe and Cheng (1991)	20	35	0·219

Table 1. Basic information of test problems.

No.	Reference solution	NG	EE	GE	000	010	020	001	002	011	021	012	022	100	110	120	101	102	111	121	112	122
1	King and Nakornchai (1982)	2	0	91·18	0·49	0·49	0·49	0·49	0·49	0·49	0·49	0·49	0·49	0·74	0·74	0·74	0·79	0·79	0·79	0·79	0·79	0·79
2	Waghodekar and Sahu (1984)	2	2	85·62	0·33	0·33	0·33	0·33	0·33	0·33	0·33	0·33	0·33	0·49	0·49	0·49	0·49	0·49	0·49	0·49	0·49	0·49
3	Waghodekar and Sahu (1984)	2	2	85·62	0·24	0·33	0·33	0·24	0·24	0·33	0·33	0·33	0·33	0·66	0·49	0·49	0·66	0·66	0·49	0·49	0·49	0·49
4	Kusiak and Cho (1992)	2	2	87·50	0·59	0·59	0·59	0·99	0·99	0·99	0·99	0·99	0·99	0·19	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99
5	Kusiak and Chung (1991)	3	1	85·49	0·33	0·33	0·33	0·33	0·33	0·33	0·33	0·33	0·33	–	0·79	0·79	0·66	0·66	0·79	0·79	0·79	0·79
6	Kusiak (1992)	3	1	85·49	0·33	0·33	0·66	0·33	0·33	0·99	0·99	0·99	0·99	0·66	0·66	0·66	0·79	0·79	0·79	0·79	0·79	0·79
7	Seifoddini (1986)	2	4	83·48	0·49	0·49	0·49	0·99	0·99	0·74	0·74	0·74	0·74	–	–	–	0·87	0·87	0·87	0·87	0·87	0·87
		2	5	85·04	0·37	0·37	0·37	0·37	0·37	0·99	0·99	0·99	0·99	0·66	0·66	0·66	0·99	0·99	0·99	0·99	0·99	0·99
8	Seifoddini and Wolfe (1986)	3	7	85·53	0·42	0·42	0·42	0·99	0·99	0·99	0·99	0·99	0·99	–	0·59	0·59	0·87	0·87	0·87	0·87	0·87	0·87
9	Seifoddini (1989c)	3	4	89·41	0·59	0·59	0·59	0·99	0·99	0·99	0·99	0·99	0·99	0·71	0·71	0·71	0·99	0·99	0·99	0·99	0·99	0·99
10	Seifoddini (1989c)	3	3	92·48	0·49	0·49	0·49	0·99	0·99	0·99	0·99	0·99	0·99	0·83	0·62	0·62	0·99	0·99	0·99	0·99	0·99	0·99
11	Seifoddini (1989c)	3	3	93·75	0·59	0·59	0·79	0·99	0·99	0·99	0·99	0·99	0·99	0·87	0·77	0·77	0·99	0·99	0·99	0·99	0·99	0·99
12	Askin *et al.* (1991)	3	5	91·50	0·49	0·49	0·49	0·99	0·99	0·99	0·99	0·99	0·99	0·88	0·79	0·79	0·99	0·99	0·99	0·99	0·99	0·99
13	Chan and Milner (1982)	3	0	96·00	0·24	0·33	0·33	0·99	0·99	0·99	0·99	0·99	0·99	0·85	0·85	0·85	0·99	0·99	0·99	0·99	0·99	0·99
14	Chandrasekharan and Rajagopalan (1986)	3	9	95·83	0·71	0·71	0·71	0·99	0·99	0·99	0·99	0·99	0·99	0·71	0·71	0·71	0·99	0·99	0·99	0·99	0·99	0·99
15	Cheng (1992)	3	10	87·82	0·45	0·45	0·45	0·99	0·99	0·79	0·79	0·99	0·99	0·64	0·64	0·64	0·94	0·94	0·94	0·94	0·94	0·94
16	Askin *et al.* (1991)	4	2	83·90	0·19	0·19	0·19	0·19	0·19	0·19	0·19	0·19	0·19	0·69	0·69	0·69	0·9	0·9	0·86	0·86	0·86	0·86
17	Askin *et al.* (1991)	5	3	83·02	0·19	0·42	0·42	0·42	0·42	0·42	0·42	0·42	0·42		0·78	0·78	0·99	0·99	0·83	0·83	0·99	0·99
18	Carrie (1973)	4	2	87·81	0·22	0·28	0·28	0·83	0·99	0·99	0·85	0·99	0·99	0·83	0·83	0·83	0·93	0·96	0·96	0·96	0·96	0·96
19	(*a*) King (1980)	4	2	74·95	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) Kusiak and Cho (1992)	5	3	80·7	0·12	0·12	0·12	0·24	0·24	0·24	0·24	0·24	0·24	0·78	0·78	0·78	0·99	0·99	0·99	0·99	0·99	0·99
20	Chandrasekharan and Rajagoplan (1989)	7	0	100·00	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·99
21	Chandrasekharan and Rajagopalan (1989)	7	10	95·20	0·49	0·49	0·49	0·99	0·99	0·99	0·99	0·99	0·99	–	0·91	0·91	0·99	0·99	0·99	0·99	0·99	0·99
22	Chandrasekharan and Rajagopalan (1987)	10	36	95·10	0·24	0·24	0·24	0·99	0·99	0·99	0·99	0·99	0·99	0·99	0·24	0·24	0·99	0·99	0·99	0·99	0·99	0·99

NG: number of groups
EE: number of exceptional elements
GE: grouping efficiency

Table 2 (*a*). Summary of computational results—group I.

No.	Reference solution	NG	EE	GE	000	010	020	001	002	011	021	012	022	100	110	120	101	102	111	121	112	122
23	Waghodekar and Sahu (1984)	2	3	77·10	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99	0·33	0·33	0·33	0·99	0·99	0·99	0·99	0·99	0·99
24	Waghodekar and Sahu (1984)	2	3	77·10	–	–	–	0·99	0·99	0·49	0·49	0·49	0·49	0·33	0·33	0·33	0·99	0·99	0·33	0·33	0·33	0·33
25	Chow and Hawaleshka (1992)	2	3	82·16	–	–	–	0·59	0·74	0·42	0·42	0·74	0·74	–	–	–	0·42	0·42	0·42	0·42	0·42	0·42
26	Kusiak and Chow (1987)	3	0	82·5	–	–	–	–	–	–	–	–	–	–	0·57	0·57	–	–	0·57	0·57	0·57	0·57
27	Chu and Tsai (1990)	3	6	89·06	–	0·33	0·33	0·99	0·99	0·99	0·99	0·99	0·99	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99
28	Seifoddini (1989 a)	2	5	86·76	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99
		2	7	89·14	0·79	0·33	0·33	–	–	–	–	–	–	0·38	0·83	0·83	–	–	–	–	–	–
29	McAuley (1972)	3	5	88·13	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99	–	–	–	0·87	0·87	0·87	0·87	0·87	0·87
30	Chan and Milner (1982)	3	5	91·50	–	–	–	0·74	0·99	0·99	0·99	0·99	0·99	–	–	–	0·99	0·99	0·99	0·99	0·99	0·99
31	Chandrasekharan and Rajagopalan (1989)	7	20	91·16	–	–	–	0·16	0·16	0·19	0·19	0·19	0·19	–	0·82	0·82	0·85	0·85	0·91	0·91	0·91	0·91
32	(*a*) Kumar and Vannelli (1987)	3	6	66·89	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) ART1	4	8	68·92	–	–	–	–	–	–	–	–	–	–	0·7	0·7	–	–	0·74	0·72	0·74	0·72
	(*b*) ART1	5	10	72·18	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0·77	0·77
	(*c*) ART1	5	10	71·01	–	–	–	–	–	–	–	–	–	0·69	–	–	–	–	–	–	–	–
33	Stanfel (1985)	4	0	68·33	–	–	–	–	–	–	–	–	–	–	0·69	0·69	–	–	0·74	0·69	0·74	0·69

NE: number of groups
EE: number of exceptional elements
GE: grouping efficiency

Table 2(*b*). Summary of computational results—group II.

No.	Reference solution	NG	EE	GE	000	010	020	001	002	011	021	012	022	100	110	120	101	102	111	121	112	122
34	(*a*) Kusiak and Chow (1987)	2	3	70·95	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) ART1	2	3	72·47	–	–	–	–	–	–	–	–	–	0.66	–	0·49	0·66	0·66	–	0·49	0·49	0·49
35	(*a*) Askin *et al.* (1991)	3	23	76·64	–	–	–	–	–	–	0·99	–	0·99	–	–	–	0·9	–	0·9	0·9	0·9	0·9
	(*b*) ART1	3	22	77·13	–	–	–	0·18	0·33	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*c*) ART1	3	21	77·13	–	–	–	–	–	–	–	–	–	–	–	–	–	0·9	–	–	–	–
	(*d*) ART1	3	20	76·69	–	–	–	–	0·18	–	–	–	–	–	–	–	–	0·94	–	–	0·94	0·94
36	(*a*) Harhalakis *et al.* (1990)	4	11	79·62	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) ART1	4	11	80·52	–	–	–	–	–	–	–	–	–	0·56	0·56	0·56	0·58	0·58	0·58	0·58	0·58	0·58
	(*c*) ART1	5	14	88·96	–	–	–	0·24	0·24	–	–	–	–	–	–	–	0·81	0·81	0·81	0·81	0·81	0·81
	(*d*) Harhalakis *et al.* (1990)	5	15	85·72	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
37	(*a*) Srinivasan *et al.* (1990)	4	21	85·36	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) ART1	4	19	86·44	–	0·28	–	0·99	0·99	0·99	0·99	0·99	0·99	–	0·69	0·69	0·99	0·99	0·99	0·99	0·99	0·99
38	(*a*) Seifoddini (1990)	5	27	75·49	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) Askin *et al.* (1991)	5	29	79·86	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*c*) ART1	5	27	79·40	–	–	–	0·28	0·28	–	–	0·24	0·24	–		–	–	0·83	0·83	0·83	0·99	0·99
39	(*a*) Seifoddini (1990)	5	27	75·49				–	–	–	–	–	–							–	–	–
	(*b*) Askin *et al.* (1991)	5	29	79·86	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*c*) ART1	5	27	79·40	–	–	–	–	–	–	–	0·24	0·24	–	–	–	–	0·78	0·84	0·84	0·91	0·91
40	(*a*) Boe and Cheng (1991)	4	40	77·36	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*b*) ART1	4	35	79·79	–	–	–	–	–	–	–	–	–		0·61	0·55	–	0·86	0·61	0·55	0·92	0·92
	(*c*) ART1	4	34	79·79	–	–		0·19	0·85	0·22	–	0·88	0·88	–	–	–	–	0·93	0·77	0·66	0·93	0·93
	(*d*) ART1	4	35	80·38	–	0·19	0·14	0·16	0·16	0·74	0·66	0·59	0·79	–	–	0·49	–	0·96	0·51	0·75	0·51	0·81

NG: number of groups
EE: number of exceptional elements
GE: grouping efficiency

Table 2(*c*). Summary of computational results—group III.

Cluster	Machines
ART1(1,0,0)	Kaparthi and Suresh (1992)
EE=10, GE=71·01	
1	1, 2, 3, 10, 11, 12, 21, 22, 23
2	4, 13, 16, 24
3	5, 6, 7, 14, 15, 17, 18, 25, 26
4	8, 27, 28
5	9, 19, 20, 29, 30
ART1(1,1,2)	
EE=10, GE=72·18	
1	1, 2, 3, 10, 11, 12, 21, 22, 23
2	6, 14, 15, 16, 25
3	4, 13, 24, 27, 28
4	5, 7, 17, 18, 26
5	8, 9, 19, 20, 29, 30

Table 3. Solution to Kumar and Vannelli (1987) problem.

Cluster	Machines	Parts
1	3, 5, 6, 8	11, 12, 18, 19
2	1, 2, 4, 9	9, 10, 14, 16, 17
3	7, 10, 11, 12	1, 2, 3, 4, 5, 6, 7, 8, 13, 15

(*a*) Askin *et al.* (1991), GE=76·74%, EE=23.

Cluster	Machines	Parts
1	1, 2, 9	5, 6, 8, 9
2	3, 4, 5, 6, 8	1, 2, 3, 4, 7
3	7, 10, 11, 12	10, 11, 12, 13, 14, 15, 16, 17, 18, 19

(*b*) ART1(0,0,1), GE=77·13%, EE=22.

Cluster	Machines	Parts
1	1, 2, 4, 8, 9	1, 4, 5, 6, 7, 8
2	7, 10, 11, 12	9, 10, 11, 12, 13, 14, 15, 16, 17, 18
3	3, 5, 6	2, 3, 19

(*c*) ART1(1,0,2), GE=77·13%, EE=21.

Cluster	Machines	Parts
1	1, 9	5, 8, 9
2	2, 3, 4, 5, 6, 8	1, 2, 3, 4, 6, 7
3	7, 10, 11, 12	10, 11, 12, 13, 14, 15, 16, 17, 18, 19

(*d*) ART1(1,0,2), GE=76·69%, EE=20.

Table 4. Part families and machine cells for problem 35.

PF-4={5,19,23,25,27,28,29}. The only difference between the extended ART1 and the reference solution is the assignment of parts 13 and 17. In the reference solution, part 13 was assigned to PF-4 and part 17 was assigned to PF-2.

In the case of problem 38, machines selected for each of the five cells are identical to those in Seifoddini's solution (1990). The extended ART1 algorithm yields a higher value of GE than that of Seifoddini's solution. This is mainly due to the difference in part-assignment procedure.

Problem 39 is the same as problem 38, except that part vectors were the inputs to the network. The basic ART1 cannot duplicate the best results reported using conventional methods. As already mentioned, either machine vectors or part vectors could be the inputs to the ART1 algorithm. For this particular problem, there are many parts being processed on one machine only. In other words, there is only one '1' entry in the part characteristic vector. It is very difficult to cluster parts into families based on a small number of features. For this problem, ART1(1,2,2) produced results slightly better than the result given by Seifoddini (1990).

For problem 40, four clusters have been identified, the solutions obtained by the extended ART1 are strikingly different from the solution provided by Boe and Cheng (1991). The extended ART1 algorithm surpasses Boe and Cheng's solution, not only in terms of GE measure but for the number of exceptional elements. Table 6 gives the final machine cells and part families for problem 40.

The common feature (with the exception of problem 34) of the test problems in group III is that a large number of exceptional elements exist in the machine-part matrix. For all these problems, the extended ART1 outperforms other reference algorithms in terms of GE as well as EE. The results indicate that the proposed approach could lead to improved solutions. The average grouping efficiency of the extended ART1 is 2·02% higher than the best results produced by various algorithms.

The quality of solution of the extended ART1 has been demonstrated through the computational results presented above. Our studies showed a definite improvement in grouping results with the supplementary procedures. Next, we would like to

Harhalakis *et al.* (1990) solution			
Cluster 1:	machines 1, 9, 10, 12, 18 parts 1, 9, 12, 14, 17, 20	Cluster 1:	machines 1, 9, 10, 12, 18 parts 1, 9, 12, 14, 17, 20
Cluster 2:	machines 2, 3, 5, 11, 14, 16, 17 parts 2, 4, 6, 7, 11, 15, 19	Cluster 2:	machines 2, 3, 11, 14 parts 2, 4, 11, 19
Cluster 3:	machines 4, 6, 7, 13, 15 parts 5, 8, 13, 16	Cluster 3:	machines 4, 6, 7, 13, 15 parts 5, 8, 13, 16
Cluster 4:	machines 8, 19, 20 parts 3, 10, 18	Cluster 4:	machines 5, 16, 17 parts 6, 7, 15
		Cluster 5:	machines 8, 19, 20 parts 3, 10, 18
ART1(1,1,1)			
Cluster 1:	machines 1, 9, 12, 18 parts 1, 9, 12, 17, 20	Cluster 1:	machines 1, 9, 12, 18 parts 1, 9, 12, 17, 20
Cluster 2:	machines 2, 3, 5, 11, 14, 16, 17 parts 2, 4, 6, 7, 11, 15, 19	Cluster 2:	machines 2, 3, 11 parts 2, 4, 11, 19
Cluster 3:	machines 4, 6, 7, 13, 15 parts 5, 8, 13, 16	Cluster 3:	machines 4, 6, 7, 15 parts 5, 8, 13, 16
Cluster 4:	machines 8, 10, 19, 20 parts 3, 10, 14, 18	Cluster 4:	machines 5, 13, 14, 16, 17 parts 6, 7, 15
		Cluster 5:	machines 8, 10, 19, 20 parts 3, 10, 14, 18

Table 5. Solution to problem 36.

know if the extended ART1 could produce a consistent solution regardless of the order of presentation. Problems 15, 31, 38, 39 and 40 were employed here to verify if the deficiencies with the ART1 algorithm regarding the order of presentation could be alleviated by the proposed procedures. These matrices are good testing problems because they contain exceptional parts and bottleneck machines. In this study, 1000 different order of presentations are randomly generated for each matrix. The ρ value will be decreased from 0·99 to 0·1 in steps of 0·01 until the solution produced by the ART1 is identical (in terms of the contents of each cluster) to the solution obtained from the reference algorithm. The results are summarized in Table 7. The entries in Table 7 represent the percentage of times that an ART1 procedure will give the identical solution to that of the reference algorithm. Examining Table 7 reveals that the problem associated with the order of presentation is well addressed by the rearrangement and reassignment procedures. For the five problems tested, ART1(1, 1, 2) and ART1(1, 2, 2) produce consistent results irrespective of the sequence of machines or parts in the initial incidence matrix.

From the above discussions, it has been demonstrated that the extended ART1 could perform as well or better than other existing algorithms. The supplementary procedures eliminate the sensitivity of the basic ART1 to the configuration of the initial matrix. The computational requirements for the extended ART1 are higher

Cluster	Machines	Parts
1	1, 7, 11, 12, 15, 16, 19	4, 6, 9, 11, 21, 28, 30, 32, 33, 34, 35
2	3, 8, 17	1, 3, 5, 15, 17 20, 25, 29
3	2, 4, 13, 14, 18	2, 7, 10, 12, 13, 18, 24, 27, 31
4	5, 6, 9, 10, 20	8, 14, 16, 19, 22, 23, 26

(*a*) Boe and Cheng's results, GE = 77·36%, EE = 40.

Cluster	Machines	Parts
1	3, 7, 8, 17	1, 3, 5, 15, 17, 20, 25, 29
2	2, 4, 13, 14, 18	2, 7, 10, 12, 13, 18, 24, 27, 31
3	1, 11, 12, 15, 16, 19	4, 6, 9, 11, 21, 28,30, 32, 33, 34, 35
4	5, 6, 9, 10, 20	8, 14, 16, 19, 22, 23, 26

(*b*) ART1(1,2,2), GE = 79·79%, EE = 35.

Cluster	Machines	Parts
1	2, 4, 13, 14, 18	2, 7, 10, 12, 13, 18, 23, 27, 31
2	1, 3, 5, 7, 8, 17	1, 3, 5, 15, 17, 20, 23, 25, 29, 34, 35
3	6, 9, 10, 20	8, 14, 16, 19, 22, 26
4	11, 12, 15, 16, 19	4, 6, 9, 11, 21, 28, 30, 32, 33

(*c*) ART1(1,2,2), GE = 79·79%, EE = 34.

Cluster	Machines	Parts
1	2, 4, 13, 14, 18	2, 7, 10, 12, 13 18, 24, 27, 31
2	1, 3, 7, 8, 17	1, 3, 5, 15, 17 20, 23, 25, 29, 34, 35
3	5, 6, 9, 10, 20	8, 14, 16, 19, 22, 26
4	11, 12, 15, 16, 19	4, 6, 9, 11, 21, 28, 30, 32, 33

(*d*) ART1(1,2,2), GE = 80·38%, EE = 35.

Table 6. Part families and machine cells for problem 40.

than those of the basic ART1 algorithm. For ART1(1,2,2), the computational requirement is approximately 2·5 times as large as that of the basic ART1 algorithm. As was mentioned previously, the ART1-based cell formation algorithm must work iteratively (by varying the value of vigilance) until an acceptable solution is achieved. With the extended procedures, the best solutions are generally found at higher values of ρ. This is an indication that the extended procedure requires less time to obtain an acceptable solution. This argument can be verified by inspecting the results shown in Table 2. The overall computational effort of the extended procedure might be less than the basic ART1 algorithm.

6. Conclusions

ART1 is a novel approach to machine cells' formation problems. One weakness of this algorithm is that the quality of grouping solution is influenced by the sequence of machines or parts in the initial machine-part incidence matrix. In this paper, we have discussed the potential sources of such a problem and proposed a set of supplementary procedures as the solution. The effectiveness of the supplementary procedures is demonstrated through its application to 40 problems from the literature. We have shown that the problem regarding the order of presentation could be alleviated by utilizing the described supplementary procedures. The results suggest that the extended ART1 could consistently produce a quality result to a cell formation problem.

For the described ART1 algorithm, the number of cells is an outcome of the solution procedure instead of an input data. The algorithm can be modified to accommodate any possible constraints on the characteristics of the resultant cells (e.g., the total number of cells, the number of machines in each cell, etc.).

Procedure	Problem 15 (NG = 3, EE = 10)	31 (NG = 7, EE = 20)	38 (NG = 5, EE = 27)	39 (NG = 5, EE = 27)	40 (NG = 4, EE = 34)
(0, 0, 0)	0·52	0·0	0·0	0·0	0·39
(0, 1, 0)	0·55	0·0	0·0	0·0	0·85
(0, 2, 0)	0·55	0·0	0·0	0·05	0·99
(0, 0, 1)	0·98	0·88	0·5	0·02	0·77
(0, 0, 2)	0·97	0·94	0·78	0·26	0·94
(0, 1, 1)	1·0	1·0	0·1	0·17	0·86
(0, 2, 1)	1·0	1·0	0·0	0·0	0.89
(0, 1, 2)	1·0	1·0	0·47	0·30	1·0
(0, 2, 2)	1·0	1·0	0·33	0·28	1·0
(1, 0, 0)	0·36	0·11	0·0	0·0	0·19
(1, 1, 0)	0·32	0·62	0·0	0·0	0·55
(1, 2, 0)	0·42	0·83	0·0	0·0	0·72
(1, 0, 1)	0·96	0·92	0·27	0·57	0·78
(1, 0, 2)	0·97	0·97	0·70	0·93	0·94
(1, 1, 1)	1·0	1·0	0·87	0·95	1·0
(1, 2, 1)	1·0	1·0	0·87	1·0	1·0
(1, 1, 2)	1·0	1·0	1·0	1·0	1·0
(1, 2, 2)	1·0	1·0	1·0	1·0	1·0

Table 7. Experimental results.

Finally, it should be mentioned that this paper only deals with the machine cell formation based on the machine-part matrix without considering other manufacturing data such as sequences of operations, production volume and so on. To take these additional factors into account, other neural network models should be investigated.

References

ASKIN, R. G., CRESSWELL, S. H., GOLDBERG, J. B., and VAKHARIA, A. J., 1991, A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research*, **29**, 1081–1100.

BOE, W. J., and CHENG, C. H., 1991, A close neighbour algorithm for designing cellular manufacturing systems. *International Journal of Production Research*, **29**, 2097–2116.

BURBIDGE, J. L., 1975, *The Introduction of Group Technology* (London: Heinemann).

BURKE, L. I., and KAMAL, S., 1992, Fuzzy ART for cellular manufacturing. In *Intelligent Engineering Systems Through Artificial Neural Networks*, **2**, Dagli *et al.* (eds) (New York: ASME Press), pp. 779–784.

CARPENTER, G. A., and GROSSBERG, S., 1986, Neural dynamics of category learning and recognition: attention, memory consolidation, and amnesia. In *Brain Structure, Learning, and Memory*, R. N. Davis and E. Wegman (eds) (AAAS Symposium Series).

CARPENTER, G. A., and GROSSBERG, S., 1987, ART2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26**, 4919–4930.

CARPENTER, G. A., and GROSSBERG, S., 1988, The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, **21**(3), 77–88.

CARPENTER, G. A., GROSSBERG, S., and ROSEN, D. B., 1991, Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, **4**, 759–771.

CARRIE, A. S., 1973, Numerical taxonomy applied to group technology and plant layout. *International Journal of Production Research*, **11**, 399–416.

CHAN, H. M., and MILNER, D. A., 1982, Direct clustering algorithm for group formation in cellular manufacturing. *Journal of Manufacturing Systems*, **1**, 65–75.

CHANDRASEKHARAN, M. P., and RAJAGOPALAN, R., 1986, An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, **24**, 451–464.

CHANDRASEKHARAN, M. P., and RAJAGOPALAN, R., 1987, ZODIAC—an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research*, **25**, 835–850.

CHANDRASEKHARAN, M. P., and RAJAGOPALAN, R., 1989, Groupability: an analysis of the properties of binary data matrices for group technology. *International Journal of Production Research*, **27**, 1035–1052.

CHENG, C. H., 1992, Algorithms for grouping machine groups in group technology. *OMEGA International Journal of Management Science*, **20**, 493–501.

CHOW, W. S., and HAWALESHKA, O., 1992, An efficient algorithm for solving the machine chaining problem in cellular manufacturing. *Computers and Industrial Engineering*, **22**, 95–100.

CHU, C. H., and TSAI, M., 1990, A comparison of three array-based clustering techniques for manufacturing cell formation. *International Journal of Production Research*, **28**, 1417–1433.

DAGLI, C., and HUGGAHALLI, R., 1991, Neural network approach to group technology. In *Knowledge Based Systems and Neural Networks: Techniques and Applications*, R. Sharda *et al.* (eds) (New York: Elsevier).

DAGLI, C., and SEN, C.-F., 1992, ART1 neural network approach to large scale group technology problems. In *Robotics and Manufacturing: Recent Trends in Research, Education and Applications*, **4**, (New York: ASME Press), pp. 787–792.

DE WITTE, J., 1980, The use of similarity coefficients in production flow analysis. *International Journal of Production Research*, **18**, 503–514.

GONGAWARE, T., and HAM, I., 1981, Cluster analysis applications for group technology manufacturing systems. *Proceedings of the 9th National North American Metalworking Research Conference* (Dearborn, Minnesota: SME).

HARHALAKIS, G., NAGI, R., and PROTH, J. M., 1990, An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research*, **28,** 185-198.

HUSH, D. R., and HORNE, B. G., 1993, Progress in supervised neural network. *IEEE Signal Processing Magazine*, January, 8-39.

KAO, Y., and MOON, Y. B., 1991, A unified group technology implementation using the backpropagation learning rule of neural networks. *Computers and Industrial Engineering*, **20,** 425-437.

KAPARTHI, S., and SURESH, N. C., 1991, A neural network system for shape-based classification and coding of rotational parts. *International Journal of Production Research*, **29,** 1771-1784.

KAPARTHI, S., and SURESH, N. C., 1992, Machine-component cell formation in group technology: a neural network approach. *International Journal of Production Research*, **30,** 1353-1367.

KING, J. R., 1980, Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, **18,** 213-232.

KING, J. R., and NAKORNCHAI, V., 1982, Machine-component group formation in group technology—review and extension. *International Journal of Production Research*, **20,** 117-133.

KUMAR, K. R., and VANNELLI, A., 1987, Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research*, **25,** 1715-1728.

KUSIAK, A., and CHO, M., 1992, Similarity coefficient algorithms for solving the group technology problem. *International Journal of Production Research*, **30,** 2633-2646.

KUSIAK, A., and CHOW, W. S., 1987, Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, **6**(2), 117-124.

KUSIAK, A., and CHUNG, Y., 1991, GT/ART: Using neural networks to form machine cells. *Manufacturing Review*, **4,** 293-301.

KUSIAK, A. (ed.), 1992, *Intelligent Design and Manufacturing* (New York: Wiley).

LEE, H., MALAVE, C. O., and RAMACHANDRAN, S., 1992, A self-organizing neural network approach for the design of cellular manufacturing systems. *Journal of Intelligent Manufacturing*, **3,** 325-332.

LEE, S., and WANG, H. P., 1992, Manufacturing cell formation: a dual-objective simulated annealing approach. *International Journal of Advanced Manufacturing Technology*, **7,** 314-320.

LIAO, T. W., and CHEN, L. J., 1993, An evaluation of ART1 network models for GT part family and machine cell forming. *Journal of Manufacturing Systems*, **12,** 282-290.

LIPPMANN, R. P., 1987, An introduction to computing with neural nets. *IEEE ASSP Magazine*, 36-54.

MALAVE, C. O., and RAMACHANDRAN, S., 1991, Neural network-based design of cellular manufacturing systems. *Journal of Intelligent Manufacturing*, **2,** 305-314.

MCAULEY, J., 1972, Machine grouping for efficient production. *The Production Engineer*, **52,** 53-57.

MCCORMICK, W. T. JR., SCHWEITZER, P. J., and WHITE, T. W., 1972, Problem decomposition and data reorganization by a cluster technique. *Operations Research*, **20,** 993-1009.

MOON, Y. B., 1990, Forming part-machine families for cellular manufacturing: a neural-network approach. *International Journal of Advanced Manufacturing Technology*, **5,** 278-291.

MOON, Y. B., and CHI, S. C., 1992, Generalized part family formation using neural network techniques. *Journal of Manufacturing Systems*, **11,** 149-159.

SEIFODDINI, H., 1986, Improper machine assignment in machine-component grouping in group technology. *Proceedings of the Fall Industrial Engineering Conference* (Boston, Massachusetts: AIIE), pp. 406-409.

SEIFODDINI, H., and WOLFE, P. M., 1986, Application of the similarity coefficient method in group technology. *IIE Transactions*, **18,** 271-277.

SEIFODDINI, H., 1989a, A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. *International Journal of Production Research*, **27,** 1161-1165.

SEIFODDINI, H., 1989 b, A probabilistic approach to machine cell formation in group technology. *Proceedings of the International Industrial Engineering Conference and Societies' Manufacturing and Productivity Symposium Proceedings*, Toronto, Canada, pp. 625–629.

SEIFODDINI, H., 1989 c, Single linkage versus average linkage clustering in machine cells formation applications. *Computers and Industrial Engineering*, **16,** 419–426.

SEIFODDINI, H., 1990, Machine-component group analysis versus the similarity coefficient method in cellular manufacturing applications. *Computers and Industrial Engineering*, **18,** 333–339.

SHAFER, S. M., KERN, G. M., and WEI, J. C., 1992, A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. *International Journal of Production Research*, **30,** 1029–1036.

SRINIVASAN, G., NARENDRAN, T. T., and MAHADEVAN, B., 1990, An assignment model for the part-families problem in group technology. *International Journal of Production Research*, **28,** 145–152.

STANFEL, L. E., 1985, Machine clustering for economic production. *Engineering Costs and Production Economics*, **9,** 73–81.

VENUGOPAL, V., and NARENDRAN, T. T., 1992, Cell formation in manufacturing systems through simulated annealing: an experimental evaluation. *European Journal of Operational Research*, **63,** 409–422.

WAGHODEKAR, P. H., and SHAU, S., 1984, Machine component cell formation in group technology: MACE. *International Journal of Production Research*, **22,** 937–948.