INT. J. PROD. RES., 2003, VOL. 41, NO. 14, 3185–3216

Performance of Fuzzy ART neural network and hierarchical clustering for part–machine grouping based on operation sequences

S. PARK† and N. C. SURESH‡*

The problem context for this study is one of identifying families of parts having a similar sequence of operations. This is a prerequisite for the implementation of cellular manufacturing, group technology, just-in-time manufacturing systems and for streamlining material flows in general. Given this problem context, this study develops an experimental procedure to compare the performance of a fuzzy ART neural network, a relatively recent neural network method, with the performance of traditional hierarchical clustering methods. For large, industry-type data sets, the fuzzy ART network, with the modifications proposed here, is capable of performance levels equal or superior to those of the widely used hierarchical clustering methods. However, like other ART networks, Fuzzy ART also results in category proliferation problems, an aspect that continues to require attention for ART networks. However, low execution times and superior solution quality make fuzzy ART a useful addition to the set of tools and techniques now available for group technology and design of cellular manufacturing systems.

1. Introduction

The design of cellular layouts (CL) has received considerable attention from researchers over the years. Production flow analysis (PFA) of Burbidge (1963, 1989) was one of the first procedures to be developed for this problem. This overall *cell formation problem* is a large problem requiring a hierarchical procedure involving heuristic procedures and subjective inputs at several stages. Within this large problem context, most of the methods developed to date have addressed the initial *part–machine grouping problem*. This problem attempts to identify families of parts that require the same set of machines without considering the sequence in which they are required. This addresses, in effect, the creation of job shop-like cells, or there is often a tacit assumption that material flows and minimization of backtracks within cells will be considered later in the overall cell formation problem.

A new research stream has emerged in recent years emphasizing the importance of considering operation sequences in part–machine grouping stage itself (Selvam and Balasubramanian 1985, Choobineh 1988, Tam 1988, Harhalakis *et al.* 1990, Vakharia and Wemmerlöv 1990, Kang and Wemmerlöv 1993, Dahel 1995, Kiang *et al.* 1995, Nair and Narendran 1998, Suresh *et al.* 1999). Considering sequences in part–machine grouping stage is desirable for several reasons:

- It is aimed directly at streamlining material flows and formation of flow lines. Flow line cells, with their streamlined work flows, enable a fuller realization of

Revision received March 2003.

† Belk College of Business Administration, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.
‡ School of Management, State University of New York, Buffalo, NY 14260, USA.
* To whom correspondence should be addressed. e-mail: ncsuresh@buffalo.edu

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2003 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0020754031000110277

the benefits of cellular manufacturing (CM), with less backtracking and material handling, easier use of conveyors within the cell, easier operation overlapping, and less lead time and work-in-process (WIP) inventory, compared with job shop-like cells.
- Ignoring operation sequences tends to distort the real extent of material-handling efforts within and outside the cells, as shown by Harhalakis *et al.* (1990).
- Identifying similar operation sequences facilitates implementation of just-in-time (JIT), business process reengineering, etc., serving to streamline materials flows in general.
- Clustering methods have advanced to a degree that sequences can now be considered directly, instead of resorting to separate, sequential (sub)optimizations within the overall, cell formation problem.

In recent years, artificial neural networks (ANNs) have emerged as viable methods for clustering large, industry-size routings data sets (Dagli and Sen 1992, Kaparthi and Suresh 1992, Kaparthi *et al.* 1993, Suresh and Kaparthi 1994). The pattern recognition capabilities of neural networks, and the ability to work with large, imperfect data sets, with low execution times render them a useful addition to the tools and techniques applicable for group technology (GT) and CM.

A notable development with neural networks in recent years is that they have been found to be applicable for sequence-based clustering, using networks such as Kohonen's self-organizing feature maps (SOFM) (Kiang *et al.* 1995) and Fuzzy ART neural network (Suresh *et al.* 1999). They have not been subjected to comprehensive and rigorous experimental comparisons with other methods. A key research question yet to be addressed is whether they outperform traditional methods such as hierarchical clustering methods.

Based on these promising developments, the objective here is to extend this new research stream further by introducing additional improvements to the use of Fuzzy ART network for sequence-based clustering, and comparing the clustering performance of this new methodology with traditional, hierarchical clustering methods. New representation schemes, clustering performance measures and experimental procedures are also developed in this process.

In this paper, Fuzzy ART neural network and traditional, hierarchical clustering procedures are used to address the part–machine grouping problem: (1) with consideration of operation sequences and (2) for problem sizes larger than those considered in past studies.

The problem addressed is a more inclusive version of the first stage, part–machine grouping problem by considering operation sequences in the clustering stage itself. Further analysis within the overall cell formation problem is conducted to assess the impact of various assignments of machine, labour and other resources to each candidate cell, specific layout decisions, evaluation of capacity, minimization of load imbalances, and ensuring adequate levels of machine use, material-handling considerations, labour-related issues such as reassignment of workers to cells, cross-training them, etc. Given the complexity, simulation methods are widely used in these latter stages.

This paper is organized as follows. A review of the literature devoted to sequence-based part–machine grouping, and neural network methods is provided in Section 2. Section 3 describes a representation scheme and performance measures adopted for sequence-based clustering. Section 4 provides a description of the algorithms tested.

Preliminary experimental verification based on small data sets from past literature, and a discussion of experimental results on industry-type data sets follow in Section 5. Finally, conclusions and research extensions appear in Section 6.

2. Sequence-based part–machine grouping procedures

The cell formation problem has been addressed extensively, using a wide range of methods, over the last two decades. A taxonomy of these works has been presented by many researchers (e.g. Chu 1989, Singh 1993, Shafer 1998).

A vast majority of the methods developed have addressed the conversion of the *part–machine incidence matrix* into a *block diagonal form*, from which more detailed cell design can be conducted. Only a limited number of studies have so far addressed part–machine grouping with the consideration of operation sequences. These studies are summarized in Section 2.1.

The use of ANNs for design of CM systems is also of relevance here. A summary of this research stream up to the point of their use for sequence-based clustering is presented in Section 2.2.

2.1. *Part–machine grouping based on operation sequences*

Table 1a provides a list of studies that have addressed part–machine grouping considering operation sequences. First, Selvam and Balasubramanian (1985) presented a three-step heuristic procedure involving a part-by-part similarity coefficient matrix. Each element in the matrix represents material-handling costs incurred by processing a part in a potential cell (column). Formulated as a *set covering problem*, a heuristic solution procedure is then developed.

Despite being an important development for sequence-based grouping, there are some drawbacks with this method. The use of a part-by-part matrix poses major

Reference	Contribution
Selvam and Balasubramanian (1985)	developed a heuristic procedure addressing sequence-dependent clustering as a set covering problem
Choobineh (1988)	a similarity coefficient to measure commonality in routing sequences for use with traditional clustering methods
Vakharia and Wemmerlöv (1990)	developed a heuristic procedure for cell formation based on commonality in material flows
Tam (1988)	a similarity coefficient based on Levenshtein distance measure and dynamic programming method to compute sequence-based similarity
Kiang *et al.* (1995)	applied the Kohonen's SOFM neural network using the similarity coefficient measure of Tam (1988)
Nair and Narendran (1998)	applied the non-hierarchical clustering procedure (ZODIAC) of Chandrasekharan and Rajagopalan (1987) for sequence-based clustering
Suresh *et al.* (1999)	developed a procedure for applying Fuzzy ART neural network for sequence-based clustering of large part–machine data sets

Table 1a. Studies on sequence-based clustering.

computational problems for practical situations involving hundreds or thousands of parts, and numerous integer variables in the formulation. Another problem relates to the matrix elements, which represent material-handling costs with some debatable assumptions behind their computation. These 'similarity coefficients' are also characterized by an asymmetry and unboundedness, as Vakharia and Wemmerlöv (1990) pointed out.

Choobineh (1988) presented a two-stage procedure, in which the first stage involved clustering into part families, and the second, the formation of machine cells considering other factors such as machine capacities. For the first clustering stage, the following sequence-based similarity coefficient was proposed:

$$S_{ik}(L) = \frac{1}{L}\left[S_{ik}(1) + \sum_{l=2}^{L} \frac{C_{ik}(l)}{N - l + 1}\right], \quad L \leq N \tag{1}$$

where $C_{ik}(l)$ is the number of common sequences of length l between parts i and k; and $S_{ik}(L)$ is the similarity coefficient of order L. This similarity coefficient needs to be computed for *every pair of parts* and used within well-known hierarchical clustering procedures like the single linkage method. Selection of the order L depends on the characteristics of the data set; the higher this value, the greater the computations involved. This method also requires the entire part–machine incidence matrix to be stored in memory. It also entails a part-by-part similarity coefficient method, and relies on other, traditional clustering methods for actual clustering.

Tam (1988) proposed a sequence-based similarity coefficient based on Levenshtein distance measure. This measures the similarity between two sequence vectors as the minimum number of elemental operations required to convert one part sequence to the other. A dynamic programming algorithm is presented to compute these coefficients. Consequently, dimensionality problems arise, and this method also involves a part-by-part similarity coefficient matrix.

Vakharia and Wemmerlöv (1990) presented the following similarity coefficient:

$$SO_{pq} = 0.5 \times \left[\left\{\sum_{i \in C_{pq}} A_{ip} \Big/ \sum_{i=1}^{M} A_{ip}\right\} + \left\{\sum_{i \in C_{pq}} A_{iq} \Big/ \sum_{i=1}^{M} A_{iq}\right\}\right], \tag{2}$$

where $i = 1, \ldots, M$ is the machine type index; $A_{ip} = 1$ if machine type i is required for part p; else $= 0$; and C_{pq} is the set of machine types required for p and q in the same sequence.

This method first involves scanning the routings to reduce them into a set of composite operation sequences. The similarity coefficients are based on this reduced set of composite sequences. This is computationally more amenable than part-by-part coefficient matrices. Following this, a procedure to allocate machines of each type is then presented to minimize backtracks within each cell. A similar workload-oriented procedure also considering possibilities of operation reallocation was presented by Kang and Wemmerlöv (1993). Both methods address the larger cell-formation problem within which the initial, sequence-based part–machine grouping may be streamlined by the use of neural network procedures suggested below.

Nair and Narendran (1998) adopted the non-hierarchical clustering procedures of Chandrasekharan and Rajagopalan (1987) and Srinivasan and Narendran (1991) for sequence-based clustering. Unlike hierarchical clustering methods, non-hierarchical methods do not necessarily require specifying the number of clusters beforehand.

However, they require identification of seeds (centroids) for each potential cluster. Machine cells are formed by progressively allotting other machines to these seeds based on rules of association (similarity) and clustering efficiency.

2.2. *Artificial neural networks for GT/CM*

Neural networks have emerged as viable methods for a wide range of applications in manufacturing (Zhang and Huang 1995). In the context of GT/CM, the pattern recognition and complexity reduction capabilities of ANNs are of particular

	Supervised learning	Unsupervised learning				
Application Area	BP/HF	CL	IA	SOFM	ART1	F-ART
Facilitate classification and coding						
Kaparthi and Suresh (1991)	•					
Design retrieval systems						
Kamarthi *et al.* (1990), Venugopal and Narendran (1992a, b)	•					
Part family formation						
Kao and Moon (1990, 1991), Moon and Roy (1992), Chakraborty and Roy (1993), Chung and Kusiak (1994)	•					
Liao and Lee (1994)					•	
Support GT-based design process						
Kusiak and Lee (1996)	•					
Block diagonalization						
Jamal (1993)	•					
Malave and Ramachandran (1991), Chu (1993), Malakooti and Yang (1995)		•				
Venugopal and Narendran (1992a, 1994)		•			•	•
Moon (1990a, b), Moon and Chi (1992), Currie (1992)			•			
Lee *et al.* (1992)				•		
Kusiak and Chung (1991), Dagli and Huggahalli (1991, 1995), Kaparthi and Suresh (1992, 1994), Dagli and Sen (1992), Kaparthi *et al.* (1993), Liao and Chen (1993), Chen and Cheng (1995)					•	
Burke and Kamal (1992, 1995), Suresh and Kaparthi (1994), Kaparthi and Suresh (1994), Kamal and Burke (1996)						•
Capacitated cell formation						
Rao and Gu (1994, 1995)					•	
Suresh *et al.* (1995)						•
Sequence-dependent clustering						
Kulkarni and Kiang (1995), Kiang *et al.* (1995)				•		
Suresh *et al.* (1999)						•

BP, back-propagation network; HF, hopfield network; CL, competitive learning model; IA, interactive activation model; SOFM, self-organizing feature maps; ART1, ART1 neural network; F-ART, Fuzzy ART neural network.

Table 1b. Studies on ANNs for group technology/cellular manufacturing.

importance and there is now a substantial body of literature devoted to the use of ANNs for GT/CM (table 1b).

In the evolution of GT, early methods for designing cells were based on identifying part families through classification and coding systems, based on engineering design information. Later methods were based more on a direct analysis of routings, as the part–machine grouping problem. The application of ANNs for GT/CM has undergone a similar evolution. A detailed review of this literature is presented by Venugopal (1998) and Suresh (2000).

A notable feature of ANNs is their ability to handle large part–machine data sets with low execution times (Dagli and Sen 1992, Kaparthi and Suresh 1992). The low execution times with ART are because they can be operated as *leader algorithms*, which do not require the entire part–machine matrix to be stored and manipulated.

The Fuzzy ART network represents an improvement in the neural network stream. It can handle both analogue and binary-valued inputs. For part–machine grouping, Fuzzy ART has been demonstrated to be superior to traditional algorithms such as BEA, ROC2 as well as those of ART1 (Burke and Kamal 1992, 1995, Suresh and Kaparthi 1994).

The use of ANNs for *sequence-dependent clustering* of routings represents the most recent development (table 1b). Kulkarni and Kiang (1995) and Kiang *et al.* (1995) used the Levenshtein distance measure of Tam (1988) (discussed in Section 2.1) and used Kohonen's SOFM to cluster parts. The applicability of this technique was demonstrated using small problem sizes. Since the similarity coefficient computations depend on dynamic programming logic, it is apparent that problems of larger size are difficult to solve.

The applicability of Fuzzy ART for sequence-dependent clustering was demonstrated by Suresh *et al.* (1999). To apply Fuzzy ART for sequence-based clustering, sequence information of each part is first captured in a matrix whose rows and columns comprise all the machine types. This matrix is made to interact with a two-dimensional array of neurons. Using this architecture, it was again shown that Fuzzy ART is capable of handling large, industry-size problems, with high levels of solution quality.

Much work remains to be done in this new stream of inquiry. The present paper extends and refines this methodology further and performs experimental comparisons to evaluate and compare the performance of Fuzzy ART neural network for sequence-based clustering with the performance of traditional, hierarchical clustering algorithms. To perform these comparisons, it also develops a suitable experimental procedure.

3. Methodology for experimental comparison

We first develop a methodology for experimental evaluation and a generalized platform for comparing the performance of neural network methods with other clustering procedures. In Section 3.1, a representation scheme is first introduced to capture machine requirements of each part, as well as the sequence in which they are required. In Section 3.2, a similarity coefficient based on these dual requirements is developed. As seen below, this is applicable for both neural network and other clustering methods. Likewise, in Section 3.3, a performance measure for sequence-based clustering is developed that is applicable for all the methods tested in this study.

3.1. *Representation scheme for routing sequences*

We first introduce the notion of a routing sequence, which is defined as an ordered listing of the machine types required by a part, in the sequence in which they are required. For example, if a part P1 is processed on machine types 3, 2, 5, 7, 8 and 4, respectively, its routing sequence is denoted as: <3 2 5 7 8 4>.

Next, the routing sequence for a part is converted into a *precedence matrix*. The precedence matrix is a *machine type-by-machine type matrix* consisting of zeros and ones. It captures the machine requirements of the part as well as the sequence pattern. The precedence matrix, $\mathbf{M}_{\mathrm{P}}$, for part P is specified such that:

- $\mathbf{M}_{\mathrm{P}}(i,i) = 1$, for every machine type i required in the routing sequence; $\mathbf{M}_{\mathrm{P}}(i,i) = 0$, for others.
- $\mathbf{M}_{\mathrm{P}}(i,j) = 1$ for $(i \neq j)$, if machine type j follows machine type i in the routing sequence; $\mathbf{M}_{\mathrm{P}}(i,j) = 0$, otherwise.

Thus, in the precedence matrix created: (1) diagonal elements with a value of 1 indicate the machine types required by the part; (2) in a given row, off-diagonal ones indicate downstream machine types; and (3) in a given column, off-diagonal ones indicate preceding machine types.

Figure 1 shows the precedence matrices for the part P1 with the routing sequence <3 2 5 7 8 4> and a part P2 with the routing sequence <7 8>. Given totally 10 machine types, part P1 uses machine types 3, 2, 5, 7, 8 and 4 in the stated order. In the precedence matrix for P1 (a 10 × 10 matrix), the diagonal elements for these machine types are ones; for the row corresponding to machine type 3, elements corresponding to successor machine types (columns 2, 5, 7, 8 and 4) are denoted as ones. Likewise, the precedence matrix for part P2 is constructed.

The precedence matrix defined here represents an improvement over the precedence matrix proposed by Suresh *et al.* (1999). The reason for the modification arose essentially due to certain requirements encountered with industry-type data sets. Industry-type data sets often involve parts with only one operation, and parts with very different sequence lengths (number of operations), and it was found that this modified version of the precedence matrix is more appropriate. These issues are discussed in Section 3.2.

Several other advantages of using this representation scheme can be mentioned. The use of a precedence matrix approach provides a convenient platform to capture

Part P1 <325784>	Machine Type
Machine Type	1 1234567890
1	....
2	.**1**.11.11..
3	.1**1**11.11..
4	...**1**......
5	...1**1**.11..
6	....
7	...1..**1**1..
8	...1...**1**..
9	....
10	....

Part P2 <78>	Machine Type
Machine Type	1 1234567890
1	....
2	....
3	....
4	....
5	....
6	....
7	......**1**1..
8	.......**1**..
9	....
10	....

Figure 1. Precedence matrices for Parts P1 and P2. Zeroes are shown as '.' for clarity.

machine requirements as well as sequences. As seen below, it is also amenable for use in neural network architectures such as fuzzy ART, and traditional methods based on hierarchical clustering. By comparing the precedence matrices of two parts, a variety of similarity coefficients can be used to assess the level of similarity in machine requirements and sequences. The use of the binary-valued precedence matrix offers some computational advantages as well.

3.2. *Sequence-based similarity coefficient*

Several similarity coefficients have been proposed in past studies for sequence-based clustering, as seen in Section 2.1. They have been used in a wide range of algorithms, including hierarchical and non-hierarchical clustering procedures.

In its simplest form, the similarity coefficient between two machines can be defined in terms of the commonality of parts processed by the two machines. For instance, McAuley (1972) defined the similarity coefficient between two machines as the number of parts visiting both machines divided by the number of parts visiting either of the two machines. This 'Jaccard coefficient' can be computed easily from a part–machine incidence matrix.

Similarly, the similarity coefficient between two parts can also be generated by comparing the machines required by the two parts. For sequence-based clustering, the Jaccard similarity coefficient between two parts can be based on their precedence matrices as:

$$S_{\mathrm{AB}} = \frac{\mathbf{M}_{\mathrm{A}} \cap \mathbf{M}_{\mathrm{B}}}{\mathbf{M}_{\mathrm{A}} \cup \mathbf{M}_{\mathrm{B}}}, \tag{3}$$

where $\mathbf{M}_{\mathrm{A}}$, $\mathbf{M}_{\mathrm{B}}$ are the precedence matrices of parts A and B, respectively. This may also be written in terms of fuzzy set operators as:

$$S_{\mathrm{AB}} = \frac{\|\mathbf{M}_{\mathrm{A}} \wedge \mathbf{M}_{\mathrm{B}}\|}{\|\mathbf{M}_{\mathrm{A}} \vee \mathbf{M}_{\mathrm{B}}\|}, \tag{4}$$

where ^ is the fuzzy AND operator: $(\mathbf{M}_{\mathrm{A}} \wedge \mathbf{M}_{\mathrm{B}})_{ij} = \min(\mathbf{M}_{\mathrm{A}ij}, \mathbf{M}_{\mathrm{B}ij})$; $\vee$ is the fuzzy OR operator: $(\mathbf{M}_{\mathrm{A}} \vee \mathbf{M}_{\mathrm{B}})_{ij} = \max(\mathbf{M}_{\mathrm{A}ij}, \mathbf{M}_{\mathrm{B}ij})$; and $\|\cdot\|$ is the number of elements in the matrix.

Shafer and Rogers (1993a, b) discussed a variety of similarity and distance measures along with desirable properties for these measures. More specific to sequence-dependent clustering, the similarity measures proposed by Tam (1988) and Choobineh (1988) were discussed in Section 2.1, along with some of their limitations.

The similarity coefficient proposed here is based directly on the precedence matrix. Given the routing sequences of two parts, their precedence matrices are computed and the Jaccard coefficient between the two precedence matrices is calculated first. This is then multiplied by a length adjustment factor (which is used for a set of reasons described below). Thus, the similarity coefficient, S_{AB}, between parts A and B is defined as:

$$S_{\mathrm{AB}} = \frac{\|\mathbf{M}_{\mathrm{A}} \wedge \mathbf{M}_{\mathrm{B}}\|}{\|\mathbf{M}_{\mathrm{A}} \vee \mathbf{M}_{\mathrm{B}}\|} \times \frac{\max(\|\mathbf{M}_{\mathrm{A}}\|, \|\mathbf{M}_{\mathrm{B}}\|)}{\min(\|\mathbf{M}_{\mathrm{A}}\|, \|\mathbf{M}_{\mathrm{B}}\|)}, \tag{5}$$

where $\mathbf{M}_{\mathrm{A}}$, $\mathbf{M}_{\mathrm{B}}$ are the precedence matrices of parts A and B, respectively.

The need for the length adjustment factor can be explained as follows. In figure 1, the routing sequence of part P1 is <3 2 5 7 8 4>, while the routing sequence of part

P2 is <7 8>. The length of routing sequence for P1 is 6, and that of P2 is 2. Given the precedence matrices shown in figure 1, the first Jaccard similarity coefficient term can be computed as 0.14. However, one may argue that the level of similarity should equal 1, since the routing sequence for P2 is a substring of the routing sequence for P1, resulting in very similar material flows. The only difference may be that P2 has to skip some operations that P1 has. After applying the length adjustment factor, it is seen that the similarity coefficient becomes 1.

Figure 2 shows the rationale behind this revised similarity coefficient, between two parts A and B, for four different cases. The routing sequence of A is given to be <1 2 3 4 5> and there are six machine types in total.

Part A

Part A <1 2 3 4 5>	Machine Type
M/c Type	123456
1	**1**1111.
2	.**1**111.
3	..**1**11.
4	...**1**1.
5	....**1**.
6	......

Case 1

Part B <2 4 5>	Machine Type
M/c Type	123456
1	......
2	.**1**.11.
3	......
4	...**1**1.
5	....**1**.
6	......

Case 2

Part B <5 4 3 2>	Machine Type
M/c Type	123456
1	......
2	.**1**....
3	.1**1**...
4	.11**1**..
5	.111**1**.
6	......

Case 3

Part B < 4 >	Machine Type
M/c Type	123456
1	......
2	......
3	......
4	...**1**..
5	......
6	......

Case 4

Part B < 6 3 5 1 >	Machine Type
M/c Type	123456
1	**1**.....
2	......
3	1.**1**.1.
4	......
5	1...**1**.
6	1.1.1**1**

Alternative similarity coefficients	Case 1	Case 2	Case 3	Case 4
Jaccard coefficient (using precedence matrix of Suresh *et al.* 1999)	0.3	0.0	0.0	0.07
Jaccard coefficient (using proposed precedence matrix)	0.4	0.19	0.06	0.15
Proposed similarity coefficient (proposed precedence matrix and length adjustment)	1.0	0.29	1.0	0.23

Figure 2. Alternative computation of similarity coefficients.

- Case 1: Routing sequence of part B is a subset of that for part A. The routing sequence of B is given as <2 4 5>. This is similar to the case of parts P1 and P2 above. If the precedence matrices of A and B were defined as in Suresh *et al.* (1999), it would yield a Jaccard similarity coefficient of 0.3. Using the Jaccard coefficient on the modified precedence matrix proposed here would yield a similarity coefficient of 0.4. However, the desirable similarity coefficient of 1 is obtained only after applying the length adjustment factor.
- Case 2: Routing sequence of B is linearly opposite to that of part A. The routing sequence of B is <5 4 3 2>. In this instance, note that there is complete commonality in usage of machine requirements among parts A and B, but the sequence of machine usage is unidirectionally opposite to each other. If the precedence matrix is defined as in Suresh *et al.* (1999), the Jaccard similarity coefficient equals 0. Using the modified precedence matrix proposed here yields a Jaccard coefficient of 0.19. After applying the length adjustment factor the similarity coefficient equals 0.29.
- Case 3: Part B requires only one operation. The routing sequence of B is <4>. A Jaccard similarity coefficient of the precedence matrices of A and B, as defined in Suresh *et al.* (1999), would yield a value of 0.0 (not defined), and with the precedence matrices as defined here, a Jaccard coefficient of 0.06. Both these values are very low. However, after applying the length adjustment factor to the latter, the similarity increases to a value of 1.0.
- Case 4: Part B shares some machines and operation sequences with part A. The routing sequence of B is <6 3 5 1>. In this case, the similarity coefficient, computed based on the modified precedence matrix and the length adjustment factor, yields a value of 0.23, given some commonality in machine usage and a weak commonality in sequence of usage.

From the above examples, the need for both modification of the precedence matrix and the introduction of length adjustment factor is clear. A performance measure is now defined for evaluating the effectiveness of the two classes of clustering methods tested here.

3.3. *Performance measure for sequence-based clustering*

Many performance measures have been developed for clustering applications over the years. These include bond energy (McCormick *et al.* 1972), grouping efficacy (Kumar and Chandrasekharan 1990) and bond efficiency (Nair and Narendran 1998). Many of these measures are based on resulting block diagonal structures. The proportion of elements outside the blocks ('exceptional elements') and voids within blocks are used to evaluate clustering efficiency. The part–machine incidence matrix does not include sequence in which machines are used, rendering these measures inapplicable for sequence-based clustering.

A second drawback relates to the fact that the clustering problem encountered in practice is one of parts versus *machine types* (instead of parts versus *machines*). In most job shops, multiple copies of machines exist for several machine types. These pools of similar machines are partitioned and may be assigned to different cells. The definition of the traditional part–machine-grouping problem does not correspond well with these job-shop realities (Suresh *et al.* 1995).

For monitoring the effectiveness of sequence-based clustering, we use the *cohesion measure* developed by Kiang *et al.* (1995). The cohesion measure has been selected for several reasons: (1) it is a single, joint measure that succinctly captures clustering efficiency without relying on separate measures for exceptional elements and voids within blocks; (2) it is amenable to part versus machine-type (instead of part versus machine) situations which correspond to shop floor reality; and (3) it is amenable to clustering which includes operation sequences as well.

The cohesion measure essentially measures the level of similarity within each cluster identified. It is computed as follows. After applying a clustering algorithm, the resulting clusters (part families) are examined, and the cohesion of each cluster is computed. The cohesion of each cluster is defined as the average of the dissimilarity coefficients between every pair of parts within the cluster. This is calculated by summing up dissimilarities (defined as one minus similarity) and dividing by the number of possible part pairs in the cluster. Thus, the mean cohesion of cluster k, C_k, is defined as:

$$C_k = \frac{\sum\limits_{x<y} (1 - S_{xy})}{\binom{n_k}{2}}, \tag{6}$$

where part $x, y \in$ cluster k and n_k is the number of members in cluster k. Next, the overall cohesion measure for all clusters, Coverall, is calculated as:

$$C_{\text{overall}} = \frac{\sum\limits_{k} (n_k - 1) C_k}{n - c}, \tag{7}$$

where n is the number of all inputs (parts) and c is the number of clusters. This is simply the weighted average of mean cohesion values of all clusters.

Note that the lower the value of C_k, the better the cohesion of parts within the cluster, since the mean cohesion is the average of dissimilarities. One aspect became known in this study regarding this measure. When the number of clusters identified by an algorithm is large, the overall cohesion measure tends to decrease, indicating better clustering. This is because as the number of clusters increases, there are a greater number of smaller clusters consisting of very similar parts within. The extreme case occurs when only one part is assigned to each cluster. However, this effect was pronounced only when the number of clusters was very large. For the sake of comparability, this clustering effectiveness measure was adopted here, with the following requirement, however. To ensure proper comparison, several numbers of resulting clusters were prespecified and kept the same for the various clustering methods compared here.

4. Algorithms tested

Two classes of clustering algorithms, hierarchical clustering methods and Fuzzy ART/merge method, were compared. In order to initiate systematic comparisons of neural networks with other methods, we considered hierarchical methods that represent the earliest among clustering methods. They are now widely implemented in many statistical software packages. For this study, the hierarchical clustering algorithms within SAS (SAS Institute 1989) were used, and they included the single linkage (SLINK), complete linkage (CLINK), average linkage (ALINK), Flexible beta (FLEXIBLE) and Ward's minimum variance (WARD) methods. These are briefly described below, along with the Fuzzy ART/merge method.

4.1. *Hierarchical clustering methods*

Hierarchical (agglomerative) clustering has been widely used for GT part family formation problems. Studies such as DeWitte (1980), Waghodekar and Sahu (1984), Seifoddini and Wolfe (1987), Mosier (1989) and Gupta and Seifoddini (1991) provide a description of each method.

Hierarchical clustering involves two essential components: (1) a metric for defining the relationship between cluster entities (e.g. parts within a family) and (2) an algorithm for using the metric to measure relationships among clusters (e.g. part families). Typically, each entity (part) begins as a cluster by itself, and the two most similar entities are merged to form a new cluster that may replace prior clusters, and the merging process repeats until only one cluster is left. This is often viewed in terms of a dendogram (figure 3).

The similarity between two elements or clusters can be converted to a distance measure by setting distance as dissimilarity (or one minus similarity). The distance between two clusters can be defined either directly or combinatorially (Lance and Williams 1967), i.e. by an equation for updating a distance matrix when two clusters are joined. A combinatorial formula, which defines the next distance relationship after a merger has occurred, is often used to reduce computational redundancy. We next briefly describe the procedures adopted for the various hierarchical clustering methods tested in this study.

4.1.1. *Single linkage method (SLINK)*

SLINK defines the distance between two observations as the minimum distance between an observation in one cluster and an observation in the other cluster. Thus, distance between two clusters C_K and C_L, is defined by:

$$D_{KL} = \min_{i \in C_k, j \in C_L} d(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

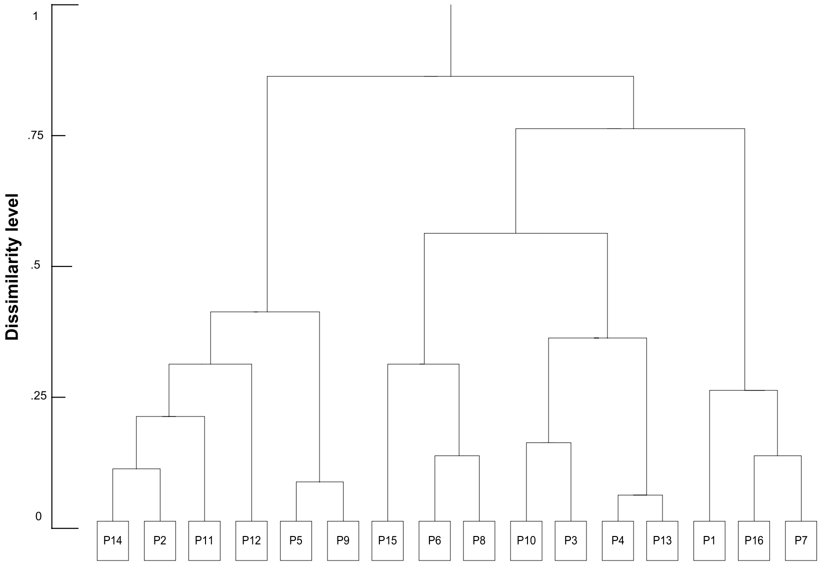

Figure 3. Dendogram for hierarchical clustering methods.

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between two observations $\mathbf{x}_i$, and $\mathbf{x}_j$. The definition of the distance measure is equivalent to the following combinatorial formula:

$$D_{JM} = \min(D_{JK}, D_{JL}). \tag{9}$$

In all the combinatorial formulae given here, it is assumed that clusters C_K and C_L are merged to form C_M, and the formula gives the distance between new cluster C_M and any other cluster C_J.

SLINK has many desirable properties including simplicity and minimal computation requirement (Hartigan 1981). Once pairwise dissimilarities are calculated and the distance (dissimilarity) matrix is constructed, the distance matrix can be used to develop the dendogram that represents the part families at different thresholds. SLINK is desirable when it is necessary to detect outliers, because outliers, with their large distances between other entities, are combined into an existing cluster at the very last stage. However, SLINK is often criticized because of 'the chaining problem': two part families may join because two of their members are similar while the remaining members may remain far apart.

4.1.2. *Complete linkage method (CLINK)*

In CLINK, the distance between two clusters is opposite of SLINK: the maximum distance between an observation in one cluster and an observation in the other:

$$D_{KL} = \max_{i \in C_k, j \in C_L} d(\mathbf{x}_i, \mathbf{x}_j). \tag{10}$$

The combinatorial formula for CLINK is:

$$D_{JM} = \max(D_{JK}, D_{JL}). \tag{11}$$

Like SLINK, CLINK is computationally simple and can use a distance matrix to develop a dendogram. With CLINK, in contrast to SLINK, moderate outliers can distort the clustering by merging them instead of merging other clusters (e.g. a cluster that consists of many similar observations with a few dissimilar ones).

4.1.3. *Average linkage method (ALINK)*

In some ways, ALINK attempts to overcome chaining problem in SLINK and clustering bias and distortion in CLINK. The distance between two clusters is the average distance between pairs of observations, one in each cluster:

$$D_{KL} = \frac{1}{n_K n_L} \sum_{i \in C_K, j \in C_L} d(\mathbf{x}_i, \mathbf{x}_j), \tag{12}$$

where n_K is the number of observations in C_K. The combinatorial formula for ALINK is:

$$D_{JM} = \frac{(n_K D_{JK} + n_L D_{JL})}{n_K + n_L}. \tag{13}$$

In ALINK, the distance matrix should be revised whenever a new cluster is formed. ALINK tends to join clusters with small variances and is slightly biased toward producing clusters with the same variance (SAS Institute 1989).

4.1.4. *Flexible beta method*

The Flexible beta method, developed by Lance and Williams (1967), defines the combinatorial formula as follows:

$$D_{JM} = (1-\beta)\frac{D_{JK}+D_{JL}}{2} + \beta D_{KL}. \tag{14}$$

Parameter β is chosen by default as -0.25 in the CLUSTER procedure of SAS if not specified (SAS Institute 1989), but for data with many outliers, smaller values have been suggested (Milligan 1989). It appears that Flexible beta method has not been used in cell formation study so far, hence it may be useful to test the performance of this method.

4.1.5. *Ward's minimum variance method (WARD)*

WARD does not merge two clusters based on the similarity or distance between them but merges them based on the difference of ANOVA sums of squares before and after merging, i.e. it minimizes the within-cluster sum of squares. When a new cluster is generated, the within-cluster sum of squares is minimized over all partitions obtainable by merging two clusters from the previous generation. This makes WARD the most time-consuming hierarchical clustering algorithm. It tends to join clusters with a small number of observations and is strongly biased toward producing clusters with roughly the same number of observations. (SAS Institute 1989). It is also sensitive to outliers because an extreme outlier can increase the within-cluster sum of squares significantly.

4.2. *Fuzzy ART/merge algorithm: a revised procedure*

The Fuzzy ART structure for sequence-based clustering, introduced in Suresh *et al.* (1999), was modified in the present study. We use the same basic structure, but it is followed by a new merging procedure. This is referred to here as the Fuzzy ART/merge method.

The merge procedure was developed for two basic reasons: (1) consistent with the experimental design adopted here, the merge procedure served to equalize the number of clusters identified by Fuzzy ART to a prespecified value and (2) the merge procedure also served to counter the category proliferation problem inherent to all ART networks. The network architecture is shown in figure 4 and a summary of the algorithm in figure 5.

In this architecture, two levels of neurons are required (figure 4). The upper layer consists of neurons ($k = 1, 2, \ldots, W$), each of which eventually comes to represent one part family identified. The lower layer consists of a M $\times$ M matrix of neurons ($i,j = 1, 2, \ldots, M$), which interfaces with the M $\times$ M precedence matrix for each part.

Step 1. Initialize network parameters: first, the weight matrix connecting upper and lower-level neurons is initialized to values of one: $\mathrm{W}_{1ij} \leftarrow 1$ for $i,j = 1, 2, \ldots, M$, where M is the number of machine types. In addition, values for the choice (α), learning (β) and vigilance (ρ) parameters are also specified: $\beta,\ \rho \in [0, 1]$. The choice parameter seems to have little impact for this application (Suresh and Kaparthi 1994). The values for β and ρ are between 0 and 1. The β value used in Step 7 specifies the speed with which the exemplars are adapted in response to new inputs, while ρ specifies

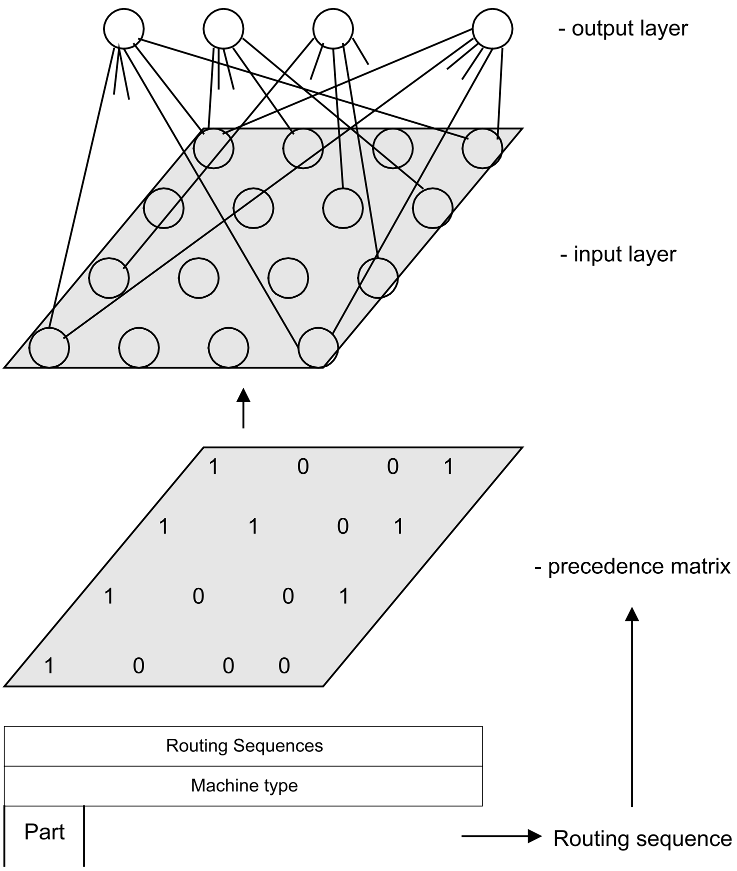

Figure 4. Fuzzy ART for sequence-based clustering.

the level of similarity between new inputs and existing exemplars. Higher ρ values result in more number of smaller families of parts, with high degrees of similarity within.

Step 2. *Input conversion*: the routing sequence for a part is read and converted into $\mathrm{M} \times \mathrm{M}$ precedence matrix **I**. The $\mathrm{M} \times \mathrm{M}$ matrices of lower-level neurons interact with these input values.

Step 3. *Calculate choice function*: for every upper-level neuron $k \le c$, we compute:

$$\mathbf{T}_k = \frac{\|\mathbf{I} \wedge \mathbf{W}_k\|}{\alpha + \|\mathbf{W}_k\|}.$$

Step 4. *Selection of best-matching node*: select the best-matching upper-level neuron, θ, whose exemplar: $\mathrm{T}_\theta = \max\{\mathbf{T}_k\}$.

Steps 5 and 6. *Resonance test and mismatch reset*: next, it is necessary to check whether this 'best-match' meets the specified level of similarity (ρ). If the selected exemplar passes this resonance test, the best-matching exemplar is updated using a learning function. Otherwise, T_θ is set to -1, so that during subsequent iterations, the input commits to the last output node, and a new node is created and initialized. Formally, Steps 5 and 6 are stated as:

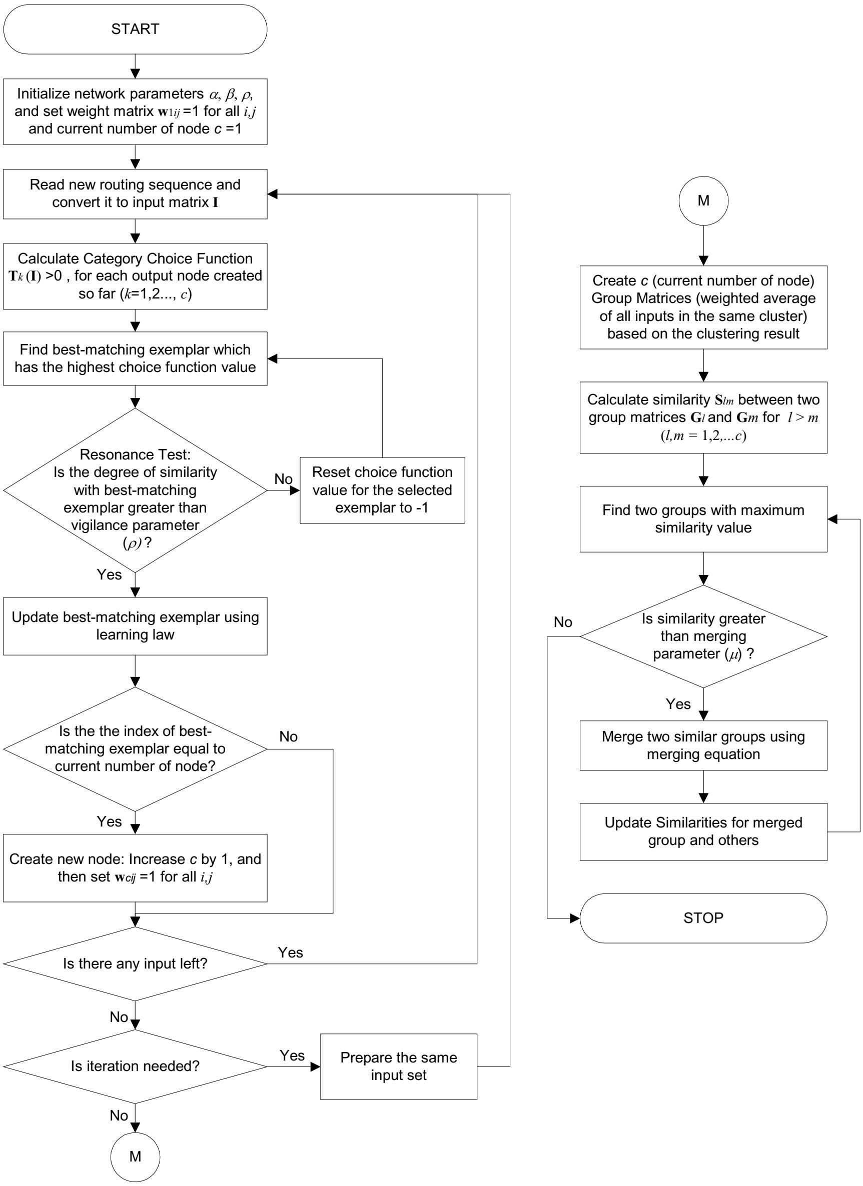
START
Initialize network parameters α, β, ρ, and set weight matrix w1ij =1 for all i,j and current number of node c =1
Read new routing sequence and convert it to input matrix I
Calculate Category Choice Function Tk (I) >0 , for each output node created so far (k=1,2..., c)
Find best-matching exemplar which has the highest choice function value
Resonance Test: Is the degree of similarity with best-matching exemplar greater than vigilance parameter (ρ)?
No
Reset choice function value for the selected exemplar to -1
Yes
Update best-matching exemplar using learning law
Is the the index of best-matching exemplar equal to current number of node?
No
Yes
Create new node: Increase c by 1, and then set wcij =1 for all i,j
Is there any input left?
Yes
No
Is iteration needed?
Yes
Prepare the same input set
No
M
M
Create c (current number of node) Group Matrices (weighted average of all inputs in the same cluster) based on the clustering result
Calculate similarity Slm between two group matrices Gl and Gm for l > m (l,m = 1,2,...c)
Find two groups with maximum similarity value
Is similarity greater than merging parameter (μ) ?
No
Yes
Merge two similar groups using merging equation
Update Similarities for merged group and others
STOP

Figure 5. Fuzzy ART/merge method.

$$\text{If } \frac{\|\mathbf{I} \wedge \mathbf{W}_\theta\|}{\|\mathbf{I}\|} \geq \rho, \text{ go to Step 7.}$$

$$\text{Set } \mathbf{T}_\theta = -1 \text{ and go to Step 4.}$$

Step 7. *Learning step*: the best-matching exemplar is updated using the learning law:

$$\mathbf{W}_\theta^{new} = \beta(\mathbf{I} \wedge \mathbf{W}_\theta^{old}) + (1 - \beta)\mathbf{W}_\theta^{old}.$$

Step 8. *If* $\theta = c$, *create new node*: if $\theta = c$, increase c by 1, and set $W_{cij} = 1$ for all i, j.

Step 9. *Repeat*: go to Step 2 unless no more input is left.

Step 10. *Iteration step*: if clustering result converges (i.e. cluster membership solution for the current iteration equals that of the previous solution) or current iteration reaches a predetermined maximum number of iterations, go to the merge procedure. Before we go to the next iteration with the same set of inputs, weights do not change for the nodes that contain more than one part in its cluster. Nodes with only one part are deleted.

4.2.1. *Merge procedure*

Next, the steps involved in the merge procedure are summarized below.

Step 1. Create group matrices: from the clusters resulting from Fuzzy ART, create group matrices as follows:

$$\mathbf{G}_{lij} = \frac{\sum \mathbf{I}_{ij}}{n_l} \text{ for all } \mathbf{I} \in \mathrm{C}_l \text{ (cluster } l = 1, 2, \dots, c),$$

where n_l is the number of members in cluster. The group matrix is simply an average of the input vectors in the category.

Step 2. *Calculate similarities between groups*: after creating group matrices, similarities between groups are calculated so that the merging may occur based on these. Similarity between group l and m is defined as:

$$S_{lm} = \frac{\|\mathbf{G}_l \wedge \mathbf{G}_m\|}{\|\mathbf{G}_l \vee \mathbf{G}_m\|} \times \frac{\max(\|\mathbf{G}_l\|, \|\mathbf{G}_m\|)}{\min(\|\mathbf{G}_l\|, \|\mathbf{G}_m\|)} \text{ for } l > m \ (l, m = 1, 2, \dots, c).$$

Step 3. Find maximum similarity: next, we find two groups, r and s, having the maximum similarity as candidates for merging: $$

$$S_{rs} = \max_{(l>m)} \{S_{lm}\}.$$

Step 4. *Similarity test*: if $S_{rs} > \mu$, proceed to Step 5; else, stop the algorithm. Merging occurs with two most similar groups if the similarity between two groups $S_{rs} > \mu$.

Step 5. *Merge the groups*: group s is merged into group r, such that:

$$\mathbf{G}_{rij} = \frac{\|\mathbf{G}_r\|}{\|\mathbf{G}_r\| + \|\mathbf{G}_s\|} \times \mathbf{G}_{rij} + \frac{\|\mathbf{G}_s\|}{\|\mathbf{G}_r\| + \|\mathbf{G}_s\|} \times \mathbf{G}_{sij}$$

$$\mathbf{G}_{sij} = 0, \text{ for all } i, j.$$

Step 6. *Update similarities*: the similarity values, S_{rk} are updated, and S_{sk} values are set to zero for $k = 1, 2, \dots, c$. Go to Step 3.

Thus, the above merging process continues until the maximum similarity between groups does not meet the desired minimum merging similarity (μ).

5. Experimental evaluation

Before developing the experimental procedure on large, industry-type data sets, preliminary validation using the few, prior data sets is first performed. Section 5.2 first summarizes the characteristics of industry-type routings data sets, providing the rationale behind the data sets generated for this series of experiments. This is followed by a discussion of the experimental design and experimental results in Section 5.3.

5.1. *Preliminary validation using prior data sets*

Prior data sets for sequence-dependent grouping are somewhat limited. All three data sets available from the literature (from Tam 1988, Harhalakis *et al.* 1990, Nair and Narendran 1998) were used for preliminary validation. Both hierarchical clustering methods and Fuzzy ART/merge were tested on these data sets, and the results are shown in tables 2–4.

Table 2 shows the results using the data set assumed by Tam (1988), which were also used by Kiang *et al.* (1995) to test the performance of Kohonen's SOFM neural network. This is the smallest of the three data sets, consisting of 19 parts and 12

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	.	.	2	.	.	.	3	4	.	.	.
2	1	2	.	4	.	3	6	5	.	.	.	.
3	1	2	.	3	.	.	4	5	6	.	.	.
4	1	.	.	2	.	.	3	.	4	.	.	.
5	1	.	.	.	.	2	4	.	5	3	.	.
6	.	.	.	.	.	1	3	4	5	2	.	.
7	.	.	.	2	.	1	.	3	4	.	.	.
8	.	3	1	5	2	4	.	6	7	.	.	.
9	.	.	1	4	2	3	.	5	6	.	.	.
10	.	.	1	3	.	2	.	4	.	.	.	.
11	.	.	.	.	.	1	.	.	.	.	.	2
12	.	.	.	.	.	.	2	.	.	.	1	3
13	.	.	.	.	.	.	3	.	.	2	1	4
14	.	.	.	.	.	.	2	.	.	3	1	.
15	.	.	.	.	.	.	.	.	.	2	1	.
16	.	.	.	.	.	.	.	.	.	.	1	2
17	.	.	.	.	.	.	2	.	.	.	1	3
18	.	.	.	.	.	1	2	.	.	3	.	.
19	.	.	.	.	.	.	2	.	.	1	.	.

Cluster	Kiang *et al.* solution	Fuzzy ART/merge solution	Best hierarchical clustering solution (flexible beta)
Cluster membership (two clusters)			
1	1 2 3 4 5 6 7 8 9 10 11 18	1 2 3 4 5 6 12 13 14 15 16 17 18 19	1 2 3 4 5 6 7 8 9 10 11 18 19
2	12 13 14 15 16 17 19	7 8 9 10 11	12 13 14 15 16 17
Cluster membership (three clusters)			
1	1 2 3 4 5 6 11 18	1 2 3 4 5 6 18	1 2 3 4 7 8 9 10 11
2	7 8 9 10	7 8 9 10 11	5 6 18 19
3	12 13 14 15 16 17 19	12 13 14 15 16 17 19	12 13 14 15 16 17
Cluster membership (four clusters)			
1	1 2 3 4	1 2 3 4	1 2 3 4
2	5 6 11 18	5 6 18	5 6 18 19
3	7 8 9 10	7 8 9 10 11	7 8 9 10 11
4	12 13 14 15 16 17 19	12 13 14 15 16 17 19	12 13 14 15 16 17

Table 2. Results for the data set of Tam (1988).

machines, with each part requiring two to seven operations. The clustering results are shown for number of clusters equal to 2, 3 and 4. The order of feeding inputs was randomly selected since clustering results often may vary with the order of inputs in many algorithms. For Fuzzy ART, the parameters assumed were: $\alpha = 0.1$, $\beta = 0.2$ and $\rho = 0.7$. Table 2 shows the clustering solutions resulting from the method of Kiang *et al.* (1995), the Fuzzy ART method and hierarchical clustering using the Flexible beta method. There are minor differences in the clustering solutions, but there is generally good agreement among the solutions.

Table 3 shows the clustering results for the data set of Harhalakis *et al.* (1990), with the number of clusters being 4 and 5. This data set consists of 20 parts and 20 machines. The solutions shown include those of Harhalakis *et al.*, the Fuzzy ART/merge and the hierarchical clustering using the Ward method. The parameters for

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

Cluster	Harhalakis *et al.* solution	Fuzzy ART/merge solution	Best hierarchical clustering solution (Ward)
Cluster membership (four clusters)			
1	1, 9, 12, 14, 17, 20	1, 9, 12, 17, 20	1, 9, 12, 17, 20
2	2, 4, 6, 7, 11, 15, 19	2, 4, 6, 7, 11, 15, 19	2, 4, 6, 7, 11, 15, 19
3	5, 8, 13, 16	5, 8, 13, 16	5, 8, 13, 16
4	3, 10, 18	3, 10, 14, 18	3, 10, 14, 18
Cluster membership (five clusters)			
1	1, 9, 12, 14, 17, 20	1, 9, 12, 17, 20	1, 9, 12, 17, 20
2	2, 4, 11, 19	2, 4, 11, 19	2, 4, 11, 19
3	5, 8, 13, 16	5, 8, 13, 16	5, 8, 13, 16
4	6, 7, 15	6, 7, 15	6, 7, 15
5	3, 10, 18	3, 10, 14, 18	3, 10, 14, 18

Table 3. Results for the data set of Harhalakis *et al.* (1990).

Fuzzy ART/merge were set to $\alpha = 0.1$, $\beta = 0.2$ and $\rho = 0.5$, with μ ranging from 0.10 to 0.41, to produce the desired number of clusters. The results show good agreement among the solutions of the three methods. The cluster memberships show that part 14 has been classified either into group $\{1, 9, 12, 17, 20\}$ or group $\{3, 10, 18\}$ with very small changes in clustering performance among the three methods.

Table 4 shows the clustering solutions of Nair and Narendran (1998), the Fuzzy ART/merge and the hierarchical clustering using Flexible beta method, assuming the number of clusters to be 8. This data set, consisting of 40 parts and 25 machines, represents the largest data set from past literature (with the exception of the 1400-part data set of Suresh *et al.* 1999). The results show good agreement among the three solutions, with only minor differences in the part groups identified. It is noteworthy that part 11 did not join any clusters in Fuzzy ART/merge solution. The most similar part based on our similarity measure is part 22 having similarity of 0.11 with part 11. Thus, in this case, it may be more appropriate to treat part 11 as an outlier and increase the desired number of clusters to 9 or more.

For these small data sets, no apparent differences were evident among the three methods, although the hierarchical clustering methods appeared to fare marginally better than Fuzzy ART. However, more generalizable conclusions can be made based on systematic comparisons on larger, more realistic data sets considered below.

5.2. *Industry data sets*

Past research studies on cell formation for the most part have assumed data sets that do not correspond too well with routings data one encounters in practice. Many points of divergence can be noted.

- Grouping problem has been addressed in the past as a matrix of parts versus *machines*, and converted into a block diagonal structure. The resulting blocks are viewed as potential cells. However, in most job shops, many work centres have multiple copies of similar machines. These pools of *machine types* are partitioned, and these machines may need to be allocated to different blocks.
- Most part–machine incidence matrices have assumed a similar number of parts and machines; in practice, the number of parts tends to be far greater than the number of machines, resulting in matrices that are far from being square matrices.
- Assuming block diagonal structures, it is also tacitly assumed that the number of operations needed for every part is approximately the same. In practice, the number of operations tends to vary greatly among parts. A significant number of parts also have a single operation, which can distort clustering results significantly. The block sizes also tend to vary greatly.
- Problem sizes assumed in most cell formation studies have also tended to be much smaller than those encountered in practice.

As an example, table 5 summarizes the characteristics of real-world routings data sets from two Dutch manufacturing firms. The first data set consists of 4415 parts (processed on 64 machine types). The number of operations is seen to vary from one to nine, with 34% of the parts requiring only one operation. The routings data set of the second firm consists of 841 parts (processed on 31 machine types). The number of operations is seen to range from one to five. Thus, in this study, the data sets assumed for the following experiments were generated with these characteristics in mind.

	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
1	.	.	.	5	.	.	3	.	.	1	.	.	.	.	.	4	.	2	.	.	.	6	.	.	.
2	2	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	.	.	.	.	.	.	.	1
3	.	.	2	.	.	.	.	.	.	.	3	.	.	.	.	.	.	.	.	1	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	2	.	.
5	.	.	.	3	.	.	.	.	.	.	.	2	.	.	.	.	.	1	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.	.	3	.	.	.	2	.	.	.	.	.	.	1	.	.
7	.	.	.	3	.	.	2	.	.	5	.	.	.	.	.	4	.	1	.	.	.	.	.	.	.
8	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	3	.	.	2	.	.	.	.	.	.
9	.	.	3	.	.	.	.	.	.	.	4	.	.	.	.	.	.	.	.	1	.	.	.	.	2
10	.	.	.	.	.	.	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3
11	.	.	.	.	.	.	.	2	.	.	.	.	3	.	.	.	.	.	.	.	1	.	.	.	.
12	1	.	4	.	.	.	.	.	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	2	5
13	.	.	3	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	1	.	.	.	.	.
14	.	.	4	.	1	.	.	.	.	.	2	.	.	.	.	.	.	.	.	3	.	.	.	.	.
15	.	4	.	.	3	.	.	.	.	5	.	.	.	.	.	1	.	.	2	.	.	.	.	.	.
16	.	.	.	1	.	.	3	.	.	.	.	.	.	.	.	2	.	4	.	.	.	.	.	.	.
17	.	.	.	.	.	.	1	.	.	3	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.
18	.	.	.	.	.	.	.	.	.	.	.	.	.	3	2	.	.	.	.	.	.	1	.	.	.
19	.	.	.	.	.	.	.	1	3	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
20	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	.	1	.	.
21	.	.	.	.	.	.	.	1	3	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
22	.	.	3	.	.	.	.	4	2	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.
23	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	3	.	.	1	.	.	.	.	.	.
24	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.
25	.	.	.	.	.	1	.	.	.	.	.	.	.	.	3	.	.	.	.	.	2	.	.	.	.
26	.	.	.	2	.	.	.	.	.	.	.	3	.	.	4	.	.	.	.	.	.	.	1	.	.
27	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	3	2	.	.	.
28	.	.	.	.	.	.	.	2	1	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
29	.	.	.	.	3	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.
30	.	.	.	4	.	.	2	.	.	.	.	.	.	.	.	3	.	1	.	.	.	.	.	.	.
31	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	.	1	.	3	.	.	.	.	.	.
32	.	.	.	.	.	.	.	.	.	.	.	.	2	1	3	.	.	.	.	.	.	4	.	.	.
33	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	3	.	.	.	.	2
34	.	.	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	1	3	.
35	.	.	.	.	.	2	.	.	.	.	.	.	.	.	4	.	.	.	.	.	1	3	.	.	.
36	2	3	.	.	.	.	.	.	.	.	4	.	.	.	.	.	1	.	.	.	.	.	.	.	.
37	.	.	.	.	.	.	3	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	1	.	.
38	.	.	.	.	.	.	.	2	3	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.
39	.	.	.	.	.	.	2	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
40	.	.	.	.	.	2	.	.	.	.	.	.	.	.	3	.	.	.	.	.	1	.	.	.	.

Cluster	Nair and Narendran solution	Fuzzy ART/merge solution	Best hierarchical clustering solution (flexible beta)
Cluster membership (eight clusters)			
1	1 5 7 16 17 30	1 4 5 6 7 16 17 20 26 30 34 37 39	1 5 7 16 17 30
2	2 12 36	2 12 36	2 12 31 36
3	3 9 13 14 33	3 9 13 14 33	3 9 13 14 33
4	4 6 20 26 34 37 39	11	4 6 20 26 34 37 39
5	8 15 23 24 31	8 15 23 24 31	8 15 23 24
6	10 19 21 22 28 38	10 19 21 22 28 38	10 19 21 22 28 38
7	11 25 27 29 35 40	25 27 29 35 40	11 18 27 29 32
8	18 32	18 32	25 35 40

Table 4. Results for the data set of Nair and Narendran (1998).

	Data set 1		Data set 2	
Length	Frequency	%	Frequency	%
1	1504	34.07	363	43.16
2	1279	28.97	363	38.05
3	921	20.86	114	13.56
4	496	11.23	42	4.99
5	143	3.24	2	0.24
6	46	1.04		
7	19	0.43		
8	3	0.07		
9	4	0.09		
Total	4415	100.00	841	100.00

Table 5. Characteristics of two real-world part routings.

5.3. *Experimental design*

The experiments were based on a *replicated clustering* procedure (Mosier 1989, Kaparthi and Suresh 1994, Suresh and Kaparthi 1994). In replicated clustering, a known solution is generated first and randomly reordered, and these scrambled data are presented to an algorithm. The clusters resulting from the algorithm are then compared and evaluated with the known starting solution. This is a procedure particularly valid for large data sets for which the optimal solution is not known before hand. Replicated clustering also enables an evaluation of the robustness of the solution to random reordering of the same data set. Many clustering algorithms are known to be sensitive to the order in which the data is presented. Past studies have often assumed a single ordering sequence of inputs and have not assessed this aspect systematically.

The case of sequence-dependent clustering poses some additional problems compared with conventional experimental designs assuming block-diagonal structures. The '*blocks*', which are parts with similar sequences, are not easily defined with uniqueness, especially when using data with a prespecified degree of imperfection. In these cases, the generated data sets may not exactly be optimal solutions to start with.

The data sets were generated as follows. Several '*base sequences*' were generated instead of a block diagonal structure. The base sequences form the basis for generating specific part routing sequences. Data sets belonging to three problem sizes were generated. The first problem size consisted of 200 parts, 20 machine types and 10 base sequences (200 × 20 × 10); the second involved 400 parts, 40 machine types and 20 base sequences; and the third 1000 parts, 80 machine types and 20 base sequences. The number of parts, the number of machine types and the number of base sequences were determined based on the literature dealing with part–machine grouping problem. However, unlike the past study, the number of base sequences is not the same as the number of part family to be expected because even with parts generated from a same base sequence may form a different part family if the operation sequence of one part is significantly different from another part. The base sequences were generated as follows. First, the length of the base sequences was set to five for every problem size. Then, depending on the problem size, five numbers of 20, 40 or 80 machine types were generated randomly and listed as the basic form for the routing sequences. Base sequences allowed for overlaps in machine requirements in order to correspond to industry-type data sets.

Surrounding each base sequence, individual part routing sequences were generated as follows. First, the number of operations for the part were determined as a randomly generated value between one and eight operations. This was based on the following probabilities: 5% for one operation; 30% for two and three operations; 20% for four operations; 10% for five operations; 10% for six to eight operations. It can be easily found that clustering parts with one operation is a trivial problem although a significant number of parts require only one operation in both of the industry data sets examined above. After clusters of other parts with more than one operation are formed, it is clear that clustering an additional part with one operation into a cluster where that operation has a slack will not change or affect the existing solution. Thus, we did not consider as many portions of parts with one operation as it may exist in real world data. The focus rather was on the decreasing pattern for the number of operations required by a part, which was seen in both industry data sets.

The part routing sequence was generated surrounding a given base sequence and a specified *dispersion level*. The dispersion level is a filtering criterion that determines if the generated part routing sequence is similar to the base sequence, within a specified level of similarity. The data sets were generated using three levels of dispersion: high (similarity level of 0.3), medium (0.5) and low (0.7). Note that the dispersion level only refers to the limiting similarity level between a part and its associated base sequence. The dispersion level does not equal the average similarity value or cohesion measure within the clusters. This unique approach to generate data sets was used mainly due to the complexity of covering too many cases that a part with many operations can cause. For example, a part with five operations can have 120 variations in sequences. In addition, unlike the replicated clustering procedure used in traditional sequence-independent clustering, this approach allows the desired solution to make more clusters than the number of base sequence.

For a given parameter combination (number of parts, number of machine types and number of base sequences), 10 different data sets were generated that formed 10 replications for each parameter combination. Thus, 30 specific data sets were generated and clustered by various algorithms. Each specific data set was clustered using four hierarchical clustering algorithms (ALINK, CLINK, Flexible beta, Ward methods) and Fuzzy ART with one iteration and five iterations. Among hierarchical methods, SLINK was omitted because initial experiments showed that it performed poorly due to the chaining problem. Most solution results showed one or two very large clusters, with numerous one-part clusters. Fuzzy ART was operated for either one or five iterations to test the proposition that the solution quality of this leader algorithm could be improved further with additional iterations.

The mean overall cohesion measures for the 10 data replications for each parameter combination are shown in figures 6–8.

An additional experimental factor was the number of clusters to be formed by each algorithm. For the first parameter combination, which was based on 10 base sequences, the number of clusters to be formed by the algorithms was set at 10, 20 and 40. For the second and third parameter combinations, which were based on 20 base sequences, the number of clusters to be formed was set at 20, 40 and 80. Thus, for each parameter combination, the number of clusters to be formed equalled the number of base sequences to start with, followed by an increased number of clusters.

	Hierarchical clustering methods				Fuzzy ART/merge			
Dispersion level	Base value	No. of clusters	Average linkage	Complete linkage	Flexible beta	Ward method	One iteration	Five iterations
High	0.6819	10	**0.7234**	0.7756	**0.7151**	**0.7248**	0.7854	0.7583
(similarity		20	0.6467	0.6657	0.6317	0.6389	0.6285	**0.5884**
= 0.3)		40	0.5480	0.5878	0.5242	0.5409	0.4386	**0.4249**
Medium	0.5292	10	0.6560	0.7218	0.6375	0.6462	**0.5530**	**0.5446**
(similarity		20	0.5616	0.5910	0.5339	0.5392	0.4562	**0.4428**
= 0.5)		40	0.4385	0.4773	0.4196	0.4272	0.3350	**0.3063**
Low	0.4309	10	0.6145	0.6861	0.5810	0.5888	0.4527	**0.4424**
(similarity		20	0.4803	0.5295	0.4644	0.4583	0.3660	**0.3517**
= 0.7)		40	0.3406	0.3897	0.3226	0.3342	0.2493	**0.2202**

Results for the best algorithm(s) are in **bold**; no significant difference with the best solution if two or more in **bold**.

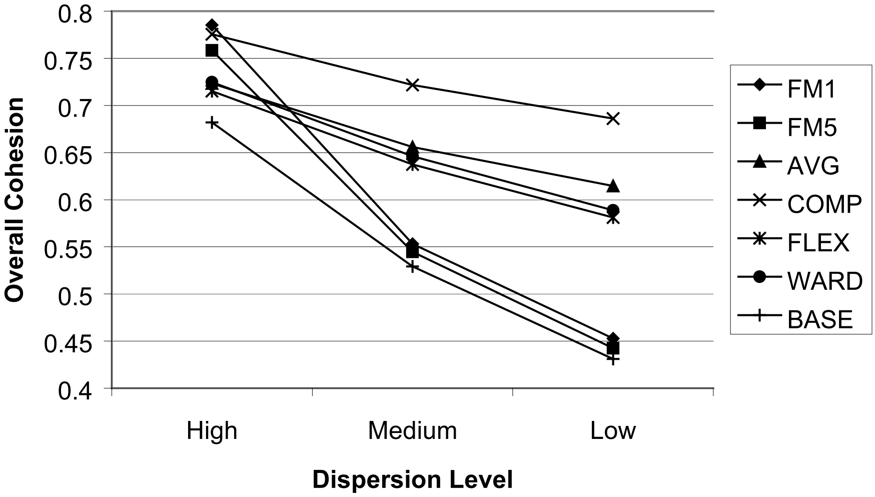

Figure 6. Overall cohesion measure (200 × 20 × 10): means from 10 data sets.

This was done to provide an expanded view of the clustering performance, over a wider range of the number of clusters formed, and to assess the robustness of well-performing algorithms over a wider parameter range.

It must be reiterated that normally the number of clusters to be formed is a input parameter only for hierarchical clustering algorithms; this is not required for Fuzzy ART-based algorithms. This requirement was enforced on Fuzzy ART merely for comparability and to suit the performance measure and the experimental design adopted.

5.4. *Results and discussion*

First, figures 6–8 show that for all three data sizes, overall cohesion measures were lower (i.e. the clustering performance was better) for lower levels of dispersion in the input data. This is consistent for all three data sizes and within every algo-

	Hierarchical clustering methods				Fuzzy ART/merge			
Dispersion level	Base value	No. of clusters	Average linkage	Complete linkage	Flexible beta	Ward method	One iteration	Five iterations
High	0.6922	20	0.7830	0.8337	**0.7604**	**0.7572**	**0.7454**	**0.7501**
(similarity		40	0.6978	0.7049	0.6701	0.6727	**0.6250**	**0.6231**
= 0.3)		80	0.5956	0.6284	0.5718	0.5807	**0.4709**	**0.4612**
Medium	0.5382	20	0.6786	0.7660	0.6753	0.6696	**0.5503**	**0.5482**
(similarity		40	0.5786	0.6157	0.5641	0.5649	0.4634	**0.4527**
= 0.5)		80	0.4638	0.4990	0.4453	0.4557	0.3353	**0.3158**
Low	0.4357	20	0.6050	0.7042	0.6029	0.6034	0.4481	**0.4388***
(similarity		40	0.4908	0.5319	0.4774	0.4755	**0.3640**	**0.3595**
= 0.7)		80	0.3535	0.4010	0.3460	0.3503	0.2437	**0.2263**

Results for the best algorithm(s) are in **bold**; no significant difference with the best solution if two or more in **bold**; * no significant difference with the base solution.

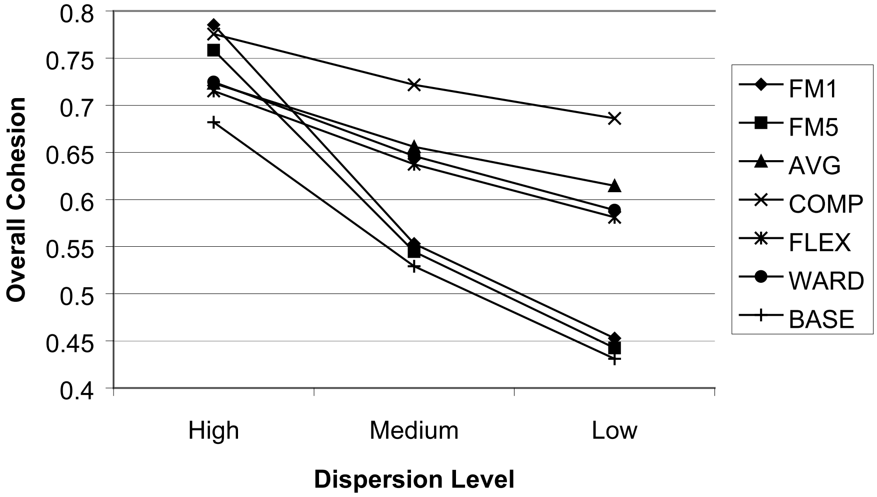

Figure 7. Overall cohesion measure (400 × 40 × 20): means from 10 data sets.

rithm. Thus, for all the methods tested, clustering effectiveness was greater when the input data was more perfect.

Note that for all three parameter combinations (figures 6–8) and for a given level of dispersion, the overall cohesion measures decreased when the required number of clusters was increased. This can been seen consistently for all the algorithms tested. Thus, it is clear that when more number of clusters were required, both hierarchical clustering and Fuzzy ART resulted in a greater number of smaller, more cohesive clusters.

Figures 6–8 also show the results from statistical comparisons among the means. Values shown in **bold** were significantly lower than other values, indicating superior clustering performance. Two or more values shown in bold indicate the absence of significant differences among the values shown in bold, based on *t*-tests.

In figures 6–8, the overall cohesion measure for the input data generated is indicated as the '*base value*'. For instance, in figure 8 (the largest data size), for

	Hierarchical clustering methods				Fuzzy ART/merge			
Dispersion level	Base value	No. of clusters	Average linkage	Complete linkage	Flexible beta	Ward method	One iteration	Five iterations
High	0.7060	20	0.8084	0.8937	0.7846	0.7787	**0.7428**	**0.7441**
(similarity		40	0.7165	0.7921	0.7002	0.6966	**0.6761**	**0.6791**
= 0.3)		80	0.6489	0.6632	0.6212	0.6242	**0.5690**	**0.5743**
Medium	0.5439	20	0.6791	0.8446	0.6545	0.6569	**0.5481**	**0.5493**
(similarity		40	0.5858	0.7032	0.5638	0.5667	**0.4948**	**0.4969**
= 0.5)		80	0.4990	0.5483	0.4692	0.4746	**0.3951**	**0.3933**
Low	0.4407	20	0.5706	0.7927	0.5610	0.5551	**0.4440**	**0.4435***
(similarity		40	0.4724	0.6151	0.4590	0.4603	**0.3928**	**0.3980**
= 0.7)		80	0.3718	0.4375	0.3625	0.3653	**0.2945**	**0.2937**

Results for the best algorithm(s) are in **bold**; no significant difference with the best solution if two or more in **bold**; * no significant difference with the base solution.

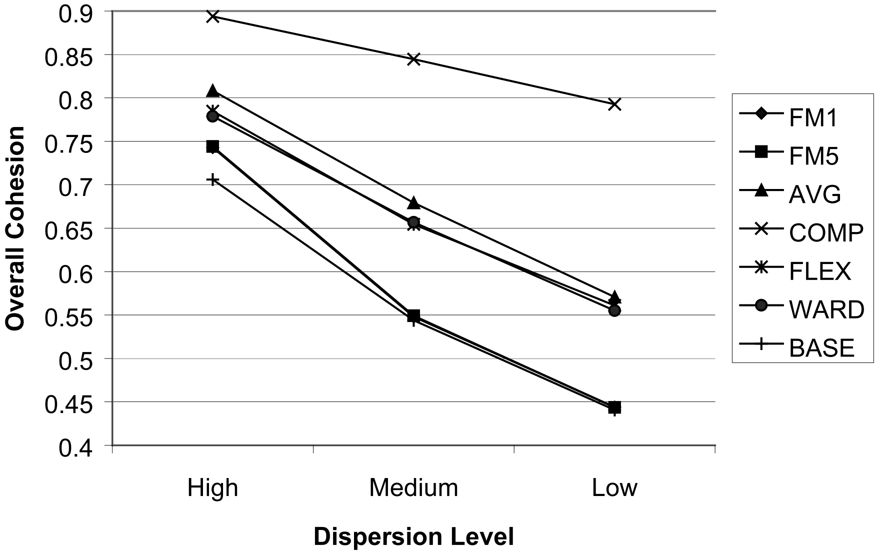

Figure 8. Overall cohesion measure (1000 × 80 × 20): means from 10 data sets.

the high dispersion level, the base value amounts to 0.7060. The number of base sequences for this parameter combination equals 20. When the required number of clusters equals this value, the overall cohesion measures obtained using all the algorithms are seen to be above it. The best values are obtained using Fuzzy ART (0.7428 and 0.7441) for one and five iterations, respectively. Since there was no significant difference between these two values, they are both shown in bold. When the number of clusters required is increased to 40 and 80, it results in cohesion measures even lower than the base value itself, indicating that smaller, much more cohesive clusters have been formed. Note that Fuzzy ART continues to outperform hierarchical clustering for higher number of clusters. Within Fuzzy ART, no significant difference is seen between the results from one and five iterations.

Based on the means shown in figures 6 and 7, it is apparent that Fuzzy ART again significantly outperforms hierarchical clustering, except in the case of high dispersion level and number of clusters equalling the number of base sequences. From the graphs of figures 6–8, it is clear that the relative advantage with Fuzzy ART over hierarchical clustering is greater at low and medium levels of dispersion.

Within Fuzzy ART, no significant difference was found between the results obtained with one iteration and five iterations. This finding tends to dispel the notion that as a leader (single-pass) algorithm, the clustering effectiveness can be improved further by using more than one pass or iteration. This is an aspect that needs to be tested systematically on all ART-based networks as well as other unsupervised neural networks.

Among the hierarchical clustering methods, Flexible beta and Ward's minimum variance methods are both comparable in performance, regardless of the dispersion level and number of clusters. It was also found that for low and medium levels of dispersion, Flexible beta, Ward and average linkage methods consistently outperformed the complete linkage method. This is apparent from figures 6–8. CLINK, as discussed in Section 3.4.2, tended to inhibit many pairs of clusters, which consisted of many similar observations with a few dissimilar ones, from joining together.

When comparing the performance of hierarchical clustering with that of Fuzzy ART, some additional factors need to be considered. Specification of the distance measure or similarity coefficient is a prerequisite for applying hierarchical clustering algorithms. This is not a problem in itself. However, with hierarchical clustering, the major drawback is that it requires computation of similarities for all possible pairs of parts. Normally, the number of parts is far greater than the number of machines in a real-world part–machine grouping problem. Thus, computational efforts are much greater with hierarchical clustering compared with unsupervised neural networks.

In this study, execution times were not monitored since the hierarchical clustering algorithms were executed using SAS on a mainframe computer, while Fuzzy ART was executed on a laptop computer. However, the significantly low execution times of fuzzy ART, compared with traditional algorithms, were demonstrated in past studies (Suresh and Kaparthi 1994, Suresh *et al.* 1999). It may be argued that execution times may not be too relevant for the part–machine grouping problem, which is often a one-time exercise. However, algorithms with low execution times are still preferred in the context of real-time, interactive optimization on the part of analysts and users.

Likewise, in addition to the difficulty of calculating the part-by-part similarity values at every step, storage requirements can be a problem with hierarchical clustering. With Fuzzy ART, the entire part–machine matrix is not stored in memory; the only storage required is for the input part matrix, and exemplar vectors associated with each high-level neuron. Neural network-based algorithms, in particular, ART-based algorithms, take advantage of the exemplar vectors, which change interactively with the stream of inputs by changing connection weights between upper layer neurons and lower layer neurons. The number of clusters should also be prespecified when hierarchical clustering method is used, whereas with fuzzy ART, it is the network parameters that need to be specified.

The category proliferation problem inherent in all ART networks was again encountered with Fuzzy ART in this study. It was primarily in response to this aspect, encountered during preliminary experimentation, that the merge feature was developed. In light of the above experiments, it is evident that this merge procedure can also be applied to other ART-based networks to counter the category proliferation problem. It appears that this has not been suggested in past neural network research, and represents another contribution of this study.

6. Conclusions

This paper addressed the problem of identifying families of parts with similar routing sequences. The problem context is one of designing CL or merely streamlining material flows, as part of the implementation of CM, JIT systems, business process re-engineering, focused factories, etc. The use of operation sequence information in the design of CM systems helps in the formation of flow-line type of cells as opposed to job-shop type of cells.

Based on promising new developments about the use of the Fuzzy ART neural network for sequence-based clustering, the objective here was to develop this methodology further by introducing additional improvements to the use of Fuzzy ART network and by comparing the clustering performance of this new methodology with traditional, hierarchical clustering methods. New representation schemes, clustering performance measures and experimental procedures were developed in this process. Both Fuzzy ART neural network and traditional, hierarchical clustering procedures were used to address the part–machine grouping problem: (1) with consideration of operation sequences and (2) for problem sizes larger than those considered in past studies.

The experiments showed the superior performance of Fuzzy ART over hierarchical clustering for large, industry-type data sets. The category proliferation problem inherent to all ART networks was again encountered, for which a merge procedure was developed. It appears that this merge routine can be applied to other unsupervised neural networks as well.

This study provides additional evidence to view Fuzzy ART neural network as a viable approach for sequence-based clustering of parts and machines. Given good performance and fast execution times for large data sets, it may be operated within a decision-support system context for cell design. It may be operated interactively and alternate configurations may be generated with different vigilance thresholds.

This study opens additional avenues for future research. It appears that further improvements in computational efficiency are possible through the use of sparse matrix representations. Other neural networks, particularly Kohonen's (1984) SOFM, may also be tested for this problem. The representation scheme developed in this study may be used for other algorithms as well in addition to developing new similarity coefficients based on the representation schemes developed for capturing machine requirements and sequences.

The clustering procedure developed is aimed at the first stage of part–machine grouping within the overall cell formation problem. Within the overall procedure, researchers have so far focused on subproblems, striving to develop good procedures for each phase. There has been a separation of methods used for the early stage and later stages. With further advances in research methods and for future research to be meaningful to practitioners, there is a need for integrating these procedures into a coherent methodology. For instance, it is conceivable that in due course the impact of using various part–machine grouping procedures can be analysed on shop performance measures such as flow time, tardiness, work in process and cost through simulation-based procedures. Much work remains to be done in these areas, however.

Part–machine grouping procedures also need to be tested further on a wide range of large, imperfect, real-world routings, for cell formation research to be of greater relevance to practitioners. This study, along with a few other past works, has made a modest advance in this regard.

References

Burbidge, J. L., 1963, Production flow analysis. *Production Engineer*, December, 742–752.

Burbidge, J. L., 1989, *Production Flow Analysis* (Oxford: Clarendon).

Burke, L. I. and Kamal, S., 1992, Fuzzy ART and cellular manufacturing. *ANNIE 92 Conference Proceedings*, pp. 779–784.

Burke, L. I. and Kamal, S., 1995, Neural networks and the part family/machine group formation problem in cellular manufacturing: a framework using Fuzzy ART. *Journal of Manufacturing Systems*, **14**, 148–159.

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.

Chakraborty, K. and Roy, U., 1993, Connectionist models for part–family classifications. *Computers and Industrial Engineering*, **24**, 189–198.

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.

Chen, S.-J. and Cheng, C.-S., 1995, A neural network-based cell formation algorithm in cellular manufacturing. *International Journal of Production Research*, **33**, 293–318.

Choobineh, F., 1988, A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, **26**, 1161–1172.

Chu, C. H., 1989, Clustering analysis in manufacturing cell formation. *OMEGA: International Journal of Management Science*, **17**, 289–295.

Chu, C. H., 1993, Manufacturing cell formation by competitive learning. *International Journal of Production Research*, **31**, 829–843.

Chung, Y. and Kusiak, A., 1994, Grouping parts with a neural network. *Journal of Manufacturing Systems*, **13**, 262–275.

Currie, K. R., 1992, An intelligent grouping algorithm for cellular manufacturing. *Computers and Industrial Engineering*, **23**, 109–112.

Dagli, C. and Huggahalli, R., 1991, *Neural Network Approach to Group Technology, Knowledge-based Systems and Neural Networks* (New York: Elsevier), pp. 213–228.

Dagli, C. and Huggahalli, R., 1995, Machine–part family formation with the adaptive resonance theory paradigm. *International Journal of Production Research*, **33**, 893–913.

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.

Dahel, N. E., 1995, Design of cellular manufacturing systems in tandem configuration. *International Journal of Production Research*, **33**, 2079–2095.

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

Gupta, T. and Seifoddini, H., 1991, Clustering algorithms for the design of a cellular manufacturing system—an analysis for their performance. *Computers and Industrial Engineering*, **20**, 461–468.

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.

Hartigan, J. A., 1981, Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association*, **76**, 388–394.

Jamal, A. M. M., 1993, Neural network and cellular manufacturing. *Industrial Management and Data Systems*, **93**, 21–25.

Kamal, S. and Burke, L. I., 1996, FACT: a new neural network-based clustering algorithm for group technology. *International Journal of Production Research*, **34**, 919–946.

Kamarthi, S. V., Kumara, S. T., Yu, F. T. S. and Ham, I., 1990, Neural networks and their applications in component design data retrieval. *Journal of Intelligent Manufacturing*, **1**, 125–140.

Kang, S. L. and Wemmerlöv, U., 1993, A work load-oriented heuristic methodology for manufacturing cell formation allowing reallocation of operations. *European Journal of Operational Research*, **69**, 292–311.

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.

Kaparthi, S. and Suresh, N. C., 1994, Performance of selected part–machine grouping techniques for data sets of wide ranging sizes and imperfection. *Decision Sciences*, **25**, 515–539.

Kaparthi, S., Suresh, N. C. and Cerveny, R. P., 1993, An improved neural network leader algorithm for part–machine grouping in group technology. *European Journal of Operational Research*, **69**, 342–356.

Kiang, M. Y., Kulkarni, U. R. and Tam, K. Y., 1995, Self-organizing map network as an interactive clustering tool—an application to group technology. *Decision Support Systems*, **15**, 351–374.

Kohonen, T., 1984, *Self-Organization and Associative Memory* (Berlin: Springer).

Kulkarni, U. R. and Kiang, M. Y., 1995, Dynamic grouping of parts in flexible manufacturing systems—a self-organizing neural networks approach. *European Journal of Operational Research*, **84**, 192–212.

Kumar, C. S. and Chandrasekharan, M. P., 1990, Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, **28**, 233–243.

Kusiak, A. and Chung, Y., 1991, GT/ART: using neural networks to form machine cells. *Manufacturing Review*, **4**, 293–301.

Kusiak, A. and Lee, H., 1996, Neural computing based design of components for cellular manufacturing. *International Journal of Production Research*, **34**, 1777–1790.

Lance, G. N. and Williams, W. T., 1967, A general theory of classification sorting strategies. I. Hierarchical systems. *Computer Journal*, **9**, 373–380.

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.

Liao, T. W. and Chen, J. L., 1993, An evaluation of ART1 neural models for GT part family and machine cell forming. *Journal of Manufacturing Systems*, **12**, 282–289.

Liao, T. W. and Lee, K. S., 1994, Integration of feature-based CAD system and an ART1 neural model for GT coding and part family forming. *Computers and Industrial Engineering*, **26**, 93–104.

Malakooti, B. and Yang, Z., 1995, A variable-parameter unsupervised learning clustering neural network approach with application to machine–part group formation. *International Journal of Production Research*, **33**, 2395–2413.

Malave, C. O. and Ramachandran, S., 1991, A neural network-based design of cellular manufacturing system. *Journal of Intelligent Manufacturing*, **2**, 305–314.

McAuley, J., 1972, Machine grouping for efficient production. *Production Engineer*, **51**, 53–57.

McCormick, W. T., Schweitzer, P. J. and White, T. W., 1972, Problem decomposition and data reorganization by a clustering technique. *Operations Research*, **20**, 992–1009.

Milligan, G. W., 1989, A study of the beta-flexible clustering method. *Multivariate Behavioral Research*, **24**, 163–176.

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

Moon, Y. B., 1990b, Forming part 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.

Moon, Y. B. and Roy, U., 1992, Learning group technology part families from solid models by parallel distributed processing. *International Journal of Advanced Manufacturing Technology*, **7**, 109–118.

Mosier, C. T., 1989, An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell formation problem. *International Journal of Production Research*, **27**, 1811.

Nair, G. J. and Narendran, T. T., 1998, CASE: a clustering algorithm for cell formation with sequence data. *International Journal of Production Research*, **36**, 157–179.

Rao, H. A. and Gu, P., 1994, Expert self-organizing neural network for the design of cellular manufacturing systems. *Journal of Manufacturing Systems*, **13**, 346–358.

Rao, H. A. and Gu, P., 1995, A multi-constraint neural network for the pragmatic design of cellular manufacturing systems. *International Journal of Production Research*, **33**, 1049–1070.

SAS Institute, 1989, *SAS/STAT User's Guide, Vol. 1, Version 6, Fourth Ed.* (Cary: SAS Institute, Inc.).

Seifoddini, H. and Wolfe, M. P., 1987, Selection of the threshold value based on material handling cost in machine-component grouping. *IIE Transactions*, **19**, 266–270.

Selvam, R. P. and Balasubramanian, K. N., 1985, Algorithmic grouping of operation sequences. *Engineering Costs and Production Economics*, **9**, 125–134.

Shafer, S. M. and Rogers, D. F., 1993a, Similarity and distance measures for cellular manufacturing. Part I. A survey. *International Journal of Production Research*, **31**, 1133–1142.

Shafer, S. M. and Rogers, D. F., 1993b, Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison. *International Journal of Production Research*, **31**, 1315–1326.

Shafer, S. M., 1998, Part–machine–labor grouping: the problem and solution methods. In N. C. Suresh and J. M. Kay (eds), *Group Technology and Cellular Manufacturing: State-of-the-Art Synthesis of Research and Practice* (Boston: Kluwer).

Singh, N., 1993, Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, **69**, 284–291.

Srinivasan, G. and Narendran, T. T., 1991, GRAFICS—a non-hierarchical clustering algorithm for group technology. *International Journal of Production Research*, **29**, 463–478.

Suresh, N. C. and Kaparthi, S., 1994, The performance of Fuzzy ART neural network for group technology cell formation. *International Journal of Production Research*, **32**, 1693–1713.

Suresh, N. C. and Kay, J. M. (eds), 1998, *Group Technology and Cellular Manufacturing: State-of-the-Art Synthesis of Research and Practice* (Boston: Kluwer).

Suresh, N. C., Slomp, J. and Kaparthi, S., 1995, The capacitated cell formation problem: a new hierarchical methodology. *International Journal of Production Research*, **33**, 1761–1784.

Suresh, N. C., Slomp, J. and Kaparthi, S., 1999, Sequence-dependent clustering of parts and machines: a Fuzzy ART neural network approach. *International Journal of Production Research*, **37**, 2793–2816.

Tam, K. Y., 1988, An operation sequence based similarity coefficient for part family formations. *Journal of Manufacturing Systems*, **9**, 55–68.

Vakharia, A. J. and Wemmerlöv, U., 1990, Designing a cellular manufacturing system: a materials flow approach based on operations sequences. *IIE Transactions*, **22**, 84–97.

Venugopal, V., 1998, Artificial neural networks and Fuzzy models: new tools for part–machine grouping. In N. C. Suresh and J. M. Kay (eds), *Group Technology and Cellular Manufacturing: State-of-the-Art Synthesis of Research and Practice* (Boston: Kluwer).

Venugopal, V. and Narendran, T. T., 1992a, A neural network approach for designing cellular manufacturing systems. *Advances in Modeling and Analysis, AMSE Press,* **32**, 13–26.

Venugopal, V. and Narendran, T. T., 1992b, Neural network model for design retrieval in manufacturing systems. *Computers in Industry*, **20**, 11–23.

Venugopal, V. and Narendran, T. T., 1994, Machine–cell formation through neural network models. *International Journal of Production Research*, **32**, 2105–2116.

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

Zhang, H.-C. and Huang, S. H., 1995, Applications of neural networks in manufacturing: a state-of-the-art survey. *International Journal of Production Research*, **33**, 705–728.

Copyright © 2003 EBSCO Publishing