Int J Adv Manuf Technol (2006) 28: 761–765
DOI 10.1007/s00170-004-2421-z

ORIGINAL ARTICLE

P. Venkumar · A. Noorul Haq

Fractional cell formation in group technology using modified ART1 neural networks

Received: 2 March 2004 / Accepted: 16 September 2004 / Published online: 18 May 2005

Abstract Group technology (GT) is a manufacturing philosophy that attempts to reduce production cost by reducing the material handling and transportation cost. The GT cell formation by any known algorithm/heuristics results in much intercell movement known as exceptional elements. In such cases, fractional cell formation using reminder cells can be adopted successfully to minimize the number of exceptional elements. The fractional cell formation problem is solved using modified adaptive resonance theory1 network (ART1). The input to the modified ART1 is machine-part incidence matrix comprising of the binary digits 0 and 1. This method is applied to the known benchmarked problems found in the literature and it is found to be equal or superior to other algorithms in terms of minimizing the number of the exceptional elements. The relative merits of using this method with respect to other known algorithms/heuristics in terms of computational speed and consistency are presented.

Keywords Adaptive resonance theory networks · Fractional cell formation · Group technology

1 Introduction

Group theory (GT) is applied in a cellular manufacturing system to identify part families and their associated machine groups so that each part family is processed within a machine group. The identification of a part family and its associated machine groups are called cell formation. The advantage of a cellular manufacturing system is that it reduces material handling time, work-in-process, throughput time, setup time, delivery time, and space. Furthermore, it provides the operational benefits of flow line production, simplification of quality control and increased job satisfaction.

Cell formation is often recognized as a complex problem in the literature. Thus, in order to reduce difficulties in the cell formation problem, a large number of methods, heuristics and non-heuristics have been developed.

The operation requirements of parts on machines can be obtained from routing cards. A matrix called the machine-parts incidence matrix, which is $m \times n$ matrix with 0 or 1 elements, commonly represents this information. In an incidence matrix, the 1 in a column indicates the column part number requires a row machine number for an operation; and the empty 0 value indicates that the column part number does not require a machine row number for an operation. All the 1s in a row and column can be rearranged in such a way that they form individual diagonal matrices. It should be noted that the row or a column alone is called an exceptional element, and is required for particular operations as well as for grouped cell operations.

There have been several methods proposed to solve this cell formation problem: array manipulation, hierarchical clustering, non-hierarchical clustering, mathematical programming, graph theory, heuristics, similarity co-efficient, knowledge-based algorithms, search algorithms, etc. These methods are found to produce good solutions for well structured matrices where part families and machine cells exists naturally. But they fail to produce good solutions for poorly structured matrices where the block diagonalization will end up with many exceptional elements. For such circumstances, fractional cell formation is employed where a fraction of the machines are grouped into cells and rest of them are grouped in the reminder cell, which will function like a pure job shop. In fractional cell formation, the parts in a part family flow inside its associated GT machine cell, and possibly, if required, through another GT cell and/or reminder cell. The movement between a GT cell and reminder cell is not considered as an exceptional element; but the movement between GT cells will be considered as exceptional elements. The incom-

P. Venkumar
Department of Mechanical Engineering,
Arulmigu Kalasalingam College of Engineering,
Anand Nagar, Krishnankoil – 626 190, Tamil Nadu, India
E-mail: pvenkumar@yahoo.com
Tel.: +91-45-63263365
Fax: +91-45-63289322

A. Noorul Haq (✉)
Department of Production Engineering,
National Institute Of Technology,
Tiruchirappalli – 620 015, Tamil Nadu, India
E-mail: anhaq@nitt.edu

ing parts from different GT cells into the reminder cell will be routed/scheduled inside the reminder cell as if it is a pure job shop.

During the past decade, there has been a considerable amount of research work in fractional cell formation. The problem was originally identified by Murthy and Srinivasan [1]. They used simulated annealing (SA) and heuristics algorithms (HA) for fractional cell formation. In other research, Srinivasan and Zimmers [2] used a neighborhood search algorithm for fractional cell formation.

This paper uses the neural network paradigm called the adaptive resonance theory1 (ART1) for cell formation. Basically, the ART1 network classifies a set of binary vectors into groups based on their similarity. The ART1 recognizes patterns and clusters of binary vectors with a recognized pattern based on a devised comparison mechanism.

In this paper, an ART1 network is employed to solve the fractional cell formation problem. The architecture of the ART1 is based on the idea of adaptive resonant feedback between two layers of nodes, as developed by Grossberg [3]. The ART1 Model described in Carpenter and Grossberg [4] was designed to cluster binary input patterns.

The ART network is an unsupervised vector classifier that accepts input vectors and subsequently classifies them according to the stored pattern they most resemble. It also provides a mechanism for the adaptive expansion of the output layer of neurons until an adequate size is reached based on the number of classes inherent in the observation. The ART network can adaptively create a new neuron corresponding to an input pattern if it is determined to be sufficiently different from existing clusters. This determination, called the vigilance test, is incorporated into the adaptive backward network. Thus, the ART architecture allows the user to control the degree of similarity of patterns placed in the cluster.

In this paper, a few modifications are made to the standard ART1 architecture to make it suitable for this fractional cell formation problem. Dagli and Huggahalli [5] and Chen and Park [6] also modified the ART1 in their works to improve its performance in GT cell formation. But their modifications are not suitable for fractional cell formation. Our proposed modifications make the ART1 suitable for GT cell formation as well as fractional cell formation. The method is validated with the test cases studied in the literature [1], and comparisons are presented.

2 Modified ART1

This modified ART1 has two phases. The first phase is similar to the standard ART1 neural networks, while the second phase is totally new as compared to the standard ART1.

2.1 First phase of the modified ART1

The architecture of the first phase of the modified ART1 has two main Layers: the input layer, also called comparison layer; and the output layer, also called recognition layer. Every input (bottom) neuron is connected to every output (top) layer neuron. There are bottom-up weights (B_{ij}) associated with the arcs from input neurons to output neurons and top-down weights (Tj) associated with the output neurons to the input neurons. The bottom-up weights are used for cluster competition and top-down weights are used for cluster verification. The first phase of the modified ART1 architecture is shown in Fig. 1.

In the first phase, the first cluster originates from ART1 using the first row input vector received from the machine-part incidence matrix. It then creates another cluster if the dissimilarity of the next input vector exceeds a certain threshold. Otherwise, this input is grouped with the corresponding cluster. The first phase ART1 procedure consists of three processes: first, a cluster search process in which the network computes a matching score to reflect the degree of similarity of the present row-wise input vector (X) to the existing stored neurons. The matching score for neuron j, denoted by net_j, is defined by:

$$net_j = \sum_{i=1}^{M} b_{ij} c_i \tag{1}$$

where

b_{ij} is a bottom-up weight vector;
c_i is output of the logical AND operator applied between input vector (X) and t_{ij}; and
t_{ij} is a top-down weight vector.

The initial t_{ij} and b_{ij} weights are calculated using the following equations:

$$bij = (L)/(L-1+M) \quad \text{for all } i \text{ and } j \tag{2}$$

$$t_{ji} = 1 \quad \text{for all } i \text{ and } j \tag{3}$$

where L is a constant > 1 (typically $L = 2$), and M is the number of input neurons.

The largest net_j, say net_J, implies that the most like group and the associated group J is the candidate of the group.

The next process of the first phase is cluster verification. Even though J is the "most like" group, it does not guarantee that the input vector (X) will pass the vigilance test. To pass the vigilance test requires $S > \varrho$, where S is the similarity ratio. The similarity ratio S is the ratio of number of 1s in the c_i to the number of 1s in the input vector. If the input X passes the test, it is included as a member of group J. Otherwise, the process returns to the cluster search process and tries the next largest net_j. The vigilance parameter ϱ, $0 \le \varrho \le 1$ determines the degree of the required similarity between the current input and a neuron already stored.

The above two processes are similar to the standard ART1 except the last cluster learning process. If the similarity between the input vector X and the group J is good enough, then the vector X is accepted as a member of group J. The learning process updates the bottom-up weight vector (b_{ij}) and the top-down vector (t_{ji}). For the new group, the top-down weight vector is identical to the input vector (X). But for the already stored neuron the logical

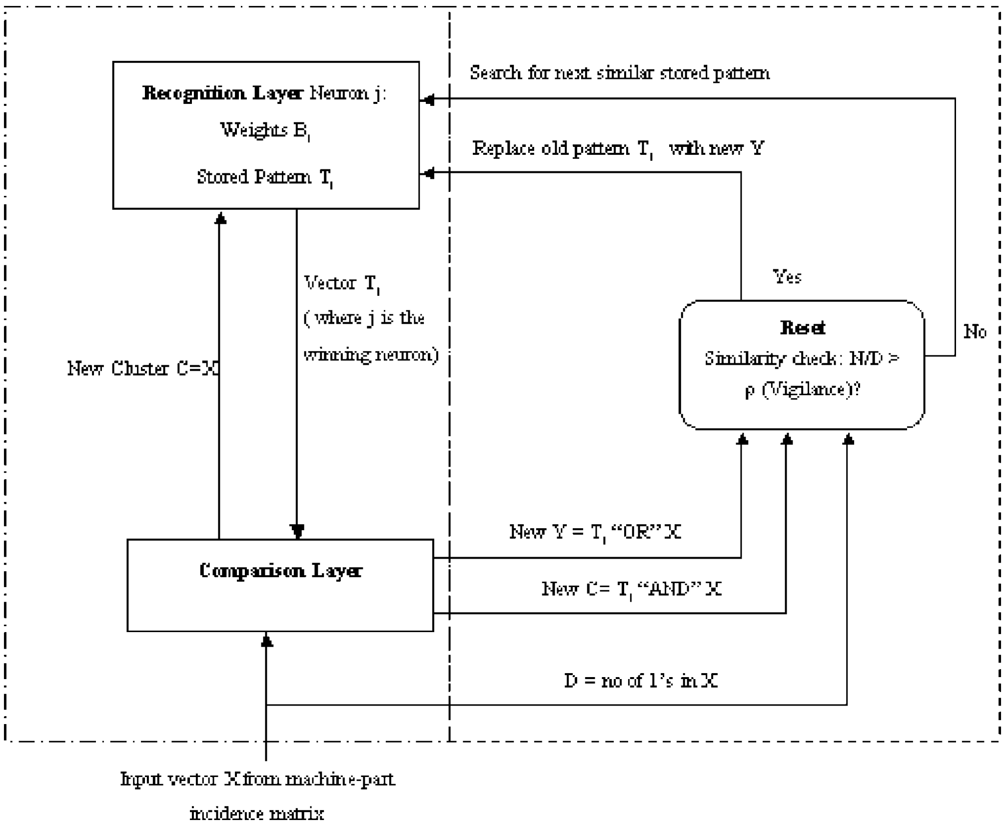

Fig. 1. The first phase of the modified ART1 architecture

OR is applied between input vector (X) and the top-down weight vector (t_j). The learning bottom-up weights are:

$$b_i j = (Ly_i) \Big/ \left(L - 1 + \sum_k y_k \right) \tag{4}$$

where y_i is the logical OR operator applied between X and t_J. The weights will be updated only for the weights associated with group J.

In the standard ART1, the learning of top-down weights vector (t_j) is the logical AND and is applied between input vector (X) and top-down weights. The major disadvantage of the standard ART1 is that for cell formation, the group is degraded during this operation. The stored neuron is degraded by the increase of 0 bits in the vector. The degraded group may lead to an improper classification in the future iterations.

The 1s in a stored neuron and 0s in an input vector X is 0 due to thelogical AND operation between the stored neuron and input vector. Once the value is set to 0, it can never be set to 1 because the logical AND for 0 for any value is always equal to 0. This problem can be overcome by replacing the logical AND operation by the logical OR operation during the learning process.

Due to the logical OR operation in the learning process, if the number of 1s increases in top-down weight vector (t_J) when compared to the previous t_J vector, then the new t_J is given as input to the network. If any other stored neuron wins, then the entire group is merged with the winner group and the weights are updated. But if the same stored neuron or new neuron wins, then the same stored patterns are maintained. The final output of the first phase of the modified ART1 is grouped machines.

2.2 Second phase of the modified ART1

In this phase, there are four processes: the cluster search process; the cluster tuning process; the fractional identification process; and the constraints verification process. In the cluster search process, the top-down weights (t_{ji}) are fixed based on the previous grouped rows, and the bottom-up weights are calculated by the following equation:

$$b_{ij} = (Lt_{ji})/(L - 1 + M) \,. \tag{5}$$

The column-wise inputs are applied to the network, which computes a matching score to reflect the degree of similarity of the present column-wise input vector (X) to the existing cluster. The matching score for node j, denoted by $fnet_j$, is defined as:

$$fnet_j = \sum_{i=1}^{M} b_{ij} x_i \,. \tag{6}$$

The largest $fnet_j$, say $fnet_J$, implies that the most like group and the associated cluster J is the candidate cluster. After column-wise input, the column groups are identified. The output of this process is the part groups.

In the second cluster tuning process the weights t_{ij} and b_{ij} are fixed based on the previous part groups, and row-wise inputs are applied to the network. The final output are grouped machines. The next step in this process is tuning. If the number of previous row groups is equal to the present number row groups, and also if the total number of part groups is equal to the total number of machine groups, then this process ends. Otherwise, the first two processes of the second phase of the modified ART1 continue until the tuning condition satisfies.

Once the tuning condition is satisfied, then the next process is fractional identification. In this process, row-wise inputs are again applied to the network. The weights are fixed based on the final part groups. If any row gives value in the following exceptional element (EE) equation, then that row (i.e. machine) is assigned to the reminder cell.

$$EE = \left(\sum fnet_j - \max(fnet_j)\right) / b_{ij} \tag{7}$$

where $fnet_j$ is calculated using Eq. 6, and b_{ij} is the bottom-up weight fixed value (i.e. $b_{ij} = L/(L-1+M)$).

In the final constraints verification process, if the constraint of the maximum number of machines permitted in the reminder (m_r) is not satisfied, then the highest value of EE rows are allotted to the reminder cell. The next constraint is that each GT cell machine group has at least two machines. If this condition is not satisfied then that cell is merged with other cells. The selection of the merging cell is based on the cell that gives a minimum EE value. The final constraint is the number of GT cells. If the number of GT cells is greater than the number of GT cells allowed, then the cells are merged so that it will satisfy the condition, as well as giving minimum of total EE value. Suppose the number of GT cells is less than the number of GT cells allowed, then the entire modified ART1 is stopped and the vigilance (ϱ) value is increased and once again the network is started from the first phase.

And the final outputs are the list of part families and the corresponding part list, GT cells and their corresponding list of machines, as well as the reminder cells with their corresponding list of machineries and the number of exceptional elements.

3 Experimentation on data from literature

The proposed modified ART1 algorithm is tested on Murthy and Srinivasan's [1] experimentation problems. In that paper, they used a simulated annealing algorithm (SA) and a heuristic algorithm (HA) for fractional cell formation. Using the modified ART1 algorithm, 24 of Murthy and Srinivasan's [1] experimentation problems were solved. The results are compared with SA and HA proposed by Murthy and Srinivasan [1]. In that paper, their objective was to minimize of exceptional elements among GT cells, and their constraints are as follows:

i. Assign each machine to only one cell, which is either GT cell or the reminder cell.
ii. Assign every part to only one GT cell.
iii. Ensure at most (m_r) machines are in the reminder cell.
iv. Ensure that a GT cell has at least two machines.
v. Ensure that the total number of cells includes reminder cell (P).

Table 1. Comparison of results for the Murthy and Srinivasan [1] test problems

Pro No.	Size	SA			HA			Modified ARTI				Source
		P	m_r	EE	P	m_r	EE	P	m_r	EE	Results compare best with	
1	8 × 20	3	4	1	3	4	7	3	4	1	HA	[7]
2	8 × 20	3	4	6	3	4	6	3	4	1	SA & HA	[7]
3	16 × 30	4	6	10	3	8	7	3	7	0	HA	[8]
4	16 × 30	4	6	0	3	8	1	4	5	0	SA	[8]
5	16 × 30	4	6	0	3	6	0	4	4	0	SA	[8]
6	16 × 30	4	6	5	3	8	4	3	8	0	HA	[8]
7	16 × 30	4	6	1	3	8	0	4	6	0	SA	[8]
8	16 × 30	4	6	0	3	6	1	4	3	0	SA	[8]
9	16 × 30	4	6	0	3	8	1	4	5	0	SA	[8]
10	16 × 30	4	6	1	3	8	0	4	6	1	Equal to SA	[8]
11	16 × 30	4	6	1	3	8	2	4	4	0	SA	[8]
12	16 × 30	4	6	1	4	7	2	4	6	1	HA	[8]
13	14 × 24	4	5	0	4	5	1	4	0	0	SA & HA	[9]
14	24 × 18	4	12	5	4	12	1	4	12	4	SA & HA	[10]
15	23 × 20	4	10	3	4	10	5	3	10	5	Constraint (P) is not satisfied	[11]
16	16 × 43	4	6	0	5	6	2	5	4	0	HA	[12]
17	27 × 27	4	15	12	4	15	15	3	13	0	Constraint (P) is not satisfied	[13]
18	24 × 40	6	10	10	4	12	6	4	12	3	HA	[14]
19	24 × 40	6	10	12	4	12	8	4	12	7	HA	[14]
20	24 × 40	6	10	15	4	12	12	4	12	9	HA	[14]
21	30 × 41	6	10	2	5	12	2	5	8	0	HA	[15]
22	28 × 46	6	10	19	5	12	8	5	11	10	HA	[10]
23	36 × 90	6	12	13	6	12	10	5	12	18	Constraint (P) is not satisfied	[12]
24	40 × 100	6	15	4	7	15	14	6	15	2	SA	[16]

In the modified ART1, we also considered Murthy and Srinivasan's [1] objective function and constraints. The ϱ is taken as 0.1, and the modified ART1 result is found to be superior or equal when compared to the SA and HA and given in Table 1. For problem number 2, 13 and 14, the modified ART1 result is superior when compared to both SA and HA. For the problems 15, 17 and 23, the modified ART1 results are not satisfied with Murthy and Srinivasan's [1] constraints. For all other problems, the modified ART1 results are superior in either SA or HA and the constraints also satisfied. In three problems, the modified ART1 results are equal to the SA and superior to HA, and only one problem result is equal to the SA. Superior means the number of machines in the reminder cell is reduced and/or the total number of exceptional elements are reduced.

4 Future work

Several improvements to the modified ART1 are possible. The scope of this paper is restricted to the modified procedure of the ART1 for the fractional cell formation with the single objective of minimizing the exceptional elements. Some of the issues like applying a number of constraints, multi-objectives, two or more reminder cells and large-sized matrices can be implemented in this modified ART1.

5 Conclusion

The modified ART1 neural network has been successfully implemented for fractional cell formation problems. The results are compared with popular existing algorithms such as the simulated annealing algorithm and heuristic algorithms. It was found that the modified ART1 solution is superior in most of the cases. The modified ART1 gives the outputs as the list of part families and the corresponding part list, GT cells and their corresponding list of machines, the reminder cell with the corresponding list of machinery, and the number of exceptional elements. The computational effort is very low in the modified ART1 when compared to all other algorithms. This modified ART1 is suitable for any size of machine-part incidence matrix.

Acknowledgement The authors wish to thank the management of Arulmigu Kalasalingam College of Engineering, Krishnankovil, India and the Department of Production Engineering of the National Institute of Technology, Thiruchirapalli, India for their support of this research work.

References

1. Murthy CVR, Srinivasan G (1995) Fractional Cell formation in group technology. Int J Prod Res 33(5):1323–1337
2. Srinivasan G, Zimmers EW Jr (1998) Fractional cell formation – issues and approaches. Int J Ind Eng –Theory Appl Practice 5:257–264
3. Grossberg S (1976) Adaptive pattern classification an universal recording: parallel development and coding of neural feature detectors. Biol Cybern 23:187–202
4. GCarpenter GA, Grossberg S (1987) A massively parallel architecture for a self-organizing neural pattern recognition machine. Comput Vision Graph Image Process 37:54–115
5. Dagli C, Huggahalli R (1995) Machine-part family formation with the adaptive resonance theory paradigm. Int J Prod Res 33(4):893–913
6. Chen DS, Chen HC, Park JM (1996) An improved ART neural net for machine cell formation. J Mater Process Technol 61:1–6
7. Chandrasekaran MP, Rajagopalan R (1986) MODROC-an extension of rank orders clustering for group technology. Int J Prod Res 24:1221–1233
8. Fayez Boctor F (1991) A linear formulation of the machine-part cell formation problem. Int J Prod Res 29(2):343–356
9. Askin RG, Subramanian SP (1987) A cost-based heuristic for group technology configuration. Int J Prod Res 25:101–113
10. Carrie AS (1973) Numerical taxonomy applied to group technology and plant layout. Int J Prod Res 24:399–416
11. Kumar KR, Kusiak A, Vannelli A (1986) Grouping of parts and components in FMS. Int J Oper Res 24:387–397
12. King JR, Nakornchai V (1982) Machine-component group formation in group technology review and extension. Int J Prod Res 20(2):117–133
13. McCormick CJ, Sweitzer DW, White TR (1972) Problem decomposition and data reorganization using a clustering technique. Oper Res 20:993–1009
14. Chandrasekaran MP, Rajagopalan R (1989) Group ability: an analysis of the properties of binary data matrices for group technology. Int J Prod Res 27:1035–1052
15. Kumar KR, Vannelli A (1987) Strategic subcontracting for efficient disaggregated manufacturing. Int J Prod Res 25:1715–1728
16. Chandrasekaran MP, Rajagopalan R (1987) Zodiac-an algorithm for concurrent formation of part-families and machine cells. Int J Prod Res 25:835–850