Int J Adv Manuf Technol (2005) 26: 909–916
DOI 10.1007/s00170-003-2048-5

ORIGINAL ARTICLE

P. Venkumar · A. Noorul Haq

Manufacturing cell formation using modified ART1 networks

Received: 15 August 2003 / Accepted: 17 November 2003 / Published online: 8 December 2004

Abstract The primary objective of group technology (GT) is to enhance the productivity in the batch manufacturing environment. The GT cell formation problem is solved using modified binary adaptive resonance theory networks known as ART1. The input to the modified ART1 is a machine-part incidence matrix comprised of the binary digits “0” and “1”. And the outputs are the list of part families and the corresponding part list, machine cells and their corresponding list of machines, and the number of exceptional elements. This method is applied to the known benchmarked problems found in the literature and it is found to outperform 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 · Cell formation · Group technology

1 Introduction

In the batch shop production environment, the cost of manufacturing is inversely proportional to the batch size and the batch size determines the productivity. In the real time environment the batch size of the components is often small leading to frequent changeovers, larger machine idleness and less productivity. To alleviate this problem group technology (GT) can be implemented to accommodate small batches without loosing much of the production run time. In GT the part families are characterized based on their similarity and its processing sequence. Similar type of parts are clustered into part families, and the associated machines, which in this family of parts are required to be processed, are grouped into machine cells. This facilitates the processing of batches of similar components within the same machine cell without much time loss in changeover/machine setup time. The primary concern in GT is to identify the machine cells and part families such that movement of parts from one GT cell to another cell is kept minimum. This is achieved through the following steps.

(1) Construction of machine-part incidence matrix with the help of route cards of all the components. In this matrix the column represents the parts, the row represents machines and the entry will have “1” or “0”, where “1” indicates the part corresponding to the particular column which is to be processed on the machine corresponding to the particular row and “0” indicates otherwise.

(2) Block diagonalizing the machine-part incidence matrix to yield 1s along the diagonal block.

The part families and machine cells can be identified from the diagonal blocks of this matrix containing more 1s and less 0s. If there are any 1s off the diagonal blocks, it indicates the inter-cell movements of the concerned parts known as exceptional elements.

There have been several methods to solve this cell formation problem viz., array manipulation, hierarchical clustering, nonhierarchical clustering, mathematical programming, graph theory, heuristics, etc. These methods are found to produce good solutions for well structured matrices where part families and machine cells exist naturally. However, they fail to produce so, for ill structured matrices in the block diagonalization will end up with many exceptional elements.

The iterative activation and competition model proposed by Moon [1] exhibited a significant advantage over the earlier algorithm, which are based on iterative procedure. The neural network applications proposed by Kaparthi and Suresh [2], Malave and Ramachandran [3] and Dagli and Huggahalli [4] have demonstrated the ability of a neural network in solving cell formation problem.

P. Venkumar
Department of Mechanical Engineering,
Arulmigu Kalasalingam College of Engineering,
Anand Nagar, Krishnankoil – 626 190, Tamil Nadu, India

A.N. Haq (✉)
Department of Production Engineering,
National Institute of Technology,
Tiruchirappalli – 620 015, Tamil Nadu, India
Email: anhaq@nitt.edu

This paper uses the neural network paradigm called ART1 paradigm 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 the binary vectors with the recognized pattern based on the devised comparison mechanism.

In this paper, an ART1 network is employed in solving the cell formation problem. The architecture of ART is based on the idea of adaptive resonant feedback between two layers of nodes as developed by Carpenter and Grossberg [5]. The ART1 Model described in Carpenter and Grossberg [5] was designed to cluster binary input patterns.

The ART network is an unsupervised vector classifier that accepts input vectors that are classified according to the stored pattern they most resemble. It also provides for a mechanism 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 on the standard ART1 architecture to make it suitable for the cell formation problem. Dagli and Huggahalli [4] also modified the ART1 in their work to improve its performance in cell formation. The limitation of the heuristics is identified and removed while formulating this procedure. This modified ART1 is totally different from Dagli and Huggahalli's [4] modified ART1 procedure. The method is validated with the test cases studied in the literature and comparisons are presented.

2 The modified ART1

2.1 Drawbacks of ART1 in cell formation

The basic form of ART1 paradigm provides excellent results when there is a similarity in the input vector. However, this method does not provide satisfactory results when the stored patterns grow sparser. Changing the vigilance parameter of the network can minimize this effect, but the optimization still becomes difficult as the number of input patterns increases. The classification process is also dependent on the order in which the input vectors are applied. In the ART1 network, determining the proper vigilance value can be problematic. For the machine-part matrix problem too high a vigilance value will result in groups that are more similar, at the expense of creating too many groups. Too low a vigilance value will result in everything being placed into just a few groups, essentially performing no true classification.

2.2 The operation of modified ART1

The modified ART1 architecture consists of two stages. The operation of the first stage is almost similar to ART1. The Fig. 1

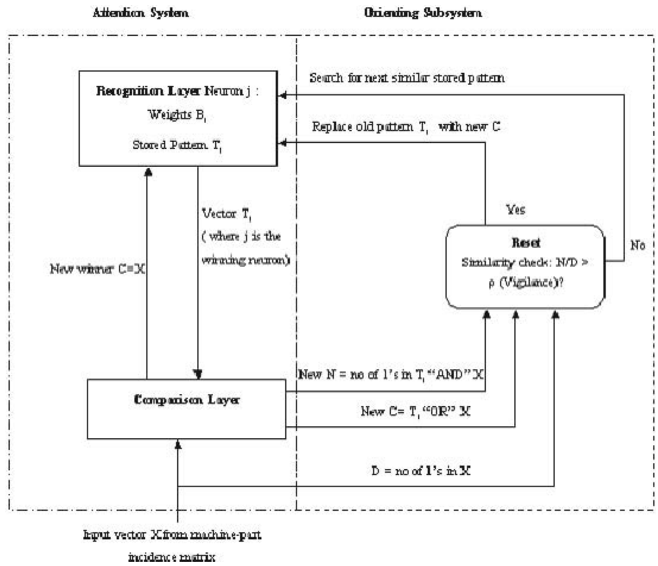

Fig. 1. The modified ART1 architecture

shows the architecture of the first stage of the modified ART1 network. In this stage, from the machine-part incidence matrix row-wise inputs are applied to the network.

The first stage consists of two layers systems, one being the attention system and the other, the orienting subsystem. The attention system consists of two layers of neurons (comparision layer and recognition layer) with feed-forward and feedbackward characteristics. The classification decision is indicated by a single neuron in the recognition layers that fires. This system determines whether the input pattern matches one of the prototypes stored. If a match occurs, resonance is established. The system also consists of an additional module labeled reset. The orienting subsystem is responsible for sensing a mismatch between the bottom-up and top-down patterns on the recognition.

The recognition layer response to an input vector is compared to the original input vector through the term vigilance. When the vigilance falls below a threshold, a new category must be created and the input vector must be stored into that category. This is a previously unallocated neuron within the recognition layer and is allocated to a new cluster category associated with the new input pattern.

Each recognition layer neuron, j, has a real-valued weight vector B_j associated with it. For each neuron in the recognition layer a dot product is formed between its associated weight B_j and the output vector C. The output C is the logical "OR" between the input vector X and the binary valued weight vector T_j. For the new cluster the input X is identical to the output C. The neuron with the largest dot product has weights that best match the input vector. It wins the competition and fires, inhibiting all other outputs from this layer.

In the original ART1, the logical "AND" operator is applied between the input vector X and T_j. In the modified ART1, due to the logical "OR" operator if the number of 1s in the output C is more when compared to the T_j then the C is given as the input to the network. If any other stored neuron wins then the clusters are merged. If the same pattern or new cluster wins then the same stored patterns are maintained.

During the comparison phase, a determination must be made as to whether an input pattern is sufficiently similar to the winning stored prototype to be assimilated by that prototype. A test for this termed vigilance is performed during this phase.

The similarity ratio, S, is the number of 1s in the logical "AND" between the vector X and the vector T_j to the number of 1s in the X vector. If the unallocated neuron (i.e., new cluster) wins then there is no similarity checking. However, for all other stored pattern there will be a criterion by which to accept or reject a cluster according to this metric. The test for vigilance can be represented as follows.

$$S > \rho \quad \text{vigilance test passed} \tag{1}$$

$$S \leqslant \rho \quad \text{vigilance test failed} \tag{2}$$

where ρ is the vigilance parameter

If the vigilance test is passed, there is no substantial difference between the input vector and the winning prototype. Thus the required action is simply to store the input vector into the winning neuron cluster. In this case, there is no reset signal. Therefore, when the search phase is entered, the weights for this input vector are adjusted.

If S is below a preset threshold, the vigilance level, then the pattern is not sufficiently similar to the winning neuron cluster and the firing neuron should be inhibited. The inhibition is done by the reset block, which resets the currently firing neuron.

The search phase is then entered and if no reset signal has been generated, the match is considered adequate and the classification is complete. Otherwise this process repeats, neuron by neuron, until one of two events occurs:

(1) A stored pattern is found that matches X above the level of the vigilance parameter, that is, $S > \rho$. The weight vectors, T_j and B_j of the firing neuron are adjusted or
(2) All stored patterns have been tried, then a previously unallocated neuron is associated with the pattern, and T_j and B_j are set to match the pattern.

In the second stage, there is only the comparison and recognition layer. There is no reset. The output of the first stage modified ART1 is the grouped rows. In this second stage the T_j and B_j weights are fixed based on the first stage-grouped rows.

Next the column-wise inputs are applied to the network. For each neuron in the recognition layer a dot product is formed between its associated weight B_j and the input vector X. The neuron with the largest dot product wins the competition and fires, inhibiting all other outputs from this layer, but the weights are not adjusted. This process continues for all the columns. The final output is the grouped columns.

Once again row-wise inputs are applied to the second stage network. The weights are fixed based on the grouped columns. If the final output from this stage´s number of row groups are identical to the previous number of row groups then the process ends. Otherwise, the second stage process continues until the previous number of row groups and current number of row groups are identical.

The next step of the modified ART1 is to find out the number of exceptional elements. Once again the row-wise inputs are applied to the second stage of modified ART1. The previous row-wise input weights are taken. If any one of the input rows has a dot product value only in the winner neuron then there are no exceptional elements. However, if there is any row which has a dot product value more than one neuron then there will be an exceptional element.

The final output of the modified ART1 is the GT cells, corresponding part and machine groups and exceptional elements.

2.3 Modified ART1 algorithm

The modified ART1 algorithm is as follows:

Step 1. Before starting the network training process, all the weight vectors bottom-up weight (B_j) and top-down weight (T_j) as well as the vigilance parameter (ρ) must be set to initial values. The ρ value range is between 0 and 1. According to Carpenter and Grossberg [5], the B_j should be

$$b_{ij} = L/(L-1+m) \quad \text{for all } i \text{ and } j, \tag{3}$$

where $m =$ the number components in the input vector
$L =$ a constant > 1, (typically $L = 2$)
The weights of the T_j are all initialized to 1, so

$$t_{ji} = 1 \quad \text{for all } i \text{ and } j. \tag{4}$$

Step 2. The row-wise inputs are applied to the network; recognition is performed as a dot product for each neuron in the recognition layer and is expressed as follows:

$$net_j = \sum_{i=1}^{m} b_{ij} c_i. \tag{5}$$

The maximum of net_j is the winner neuron,
where $b_{ij} =$ the bottom-up weight in the neuron i in the comparison layer to neuron j in the recognition layer.
$c_i =$ the output vector of the comparison layer neuron; for the new cluster it is identical to input X_i, otherwise the logical "OR" vector applied between input vector X_i and the stored vector T_j.
If the new cluster wins then go to step 4 otherwise go to step 3. For the new cluster all the T_j elements are 1.

Step 3. For the winner, compare the vector T_j to the input vector X. If their similarity S is below the vigilance threshold, then select the next maximum net_j value as a winner neuron. This step continues until the condition 7 is satisfied.

$$S = K/D \tag{6}$$

$$S > \rho, \tag{7}$$

where
$D =$ the number of 1s in the X vector and

K = the number of 1s in the resulting vector. The resulting vector is the logical "AND" vector applied between X and T_j.

Step 4. In this step, the network enters a training cycle that modifies the weights in t_{ji} and b_{ij}, the weight vectors associated with the winning recognition layer neuron. The weights in the vector t_{ji} that are associated with the new stored pattern are adjusted so that they equal the corresponding binary values in the vector C

$$t_{ji} = c_i \quad \text{for all } i. \tag{8}$$

If the number of 1s increases in c_i when the logical "OR" vector is applied between X and T_j, then c_i is given as an input to the network. If any other stored neuron wins and also passes the vigilance test then the entire group is merged with the stored neuron and the t_{ji} updated. Suppose all other stored patterns are found to mismatch the input, but a previously unallocated neuron or same stored neuron wins, then the same stored patterns are maintained.

Step 4a. The b_{ij} are updated to the normalized values of the vector C. According to Carpenter and Grossberg [5] the b_{ij} should be

$$b_{ij} = (Lc_i)/\left(L - 1 + \sum_k c_k\right). \tag{9}$$

Step 5. Repeat Step 2 to Step 4 for all the rows.

Step 6. Based on the grouped rows, the T_j weights are fixed and then the B_j weights are fixed by using Eq. 10

$$b_{ij} = \left(Lt_{ji}\right)/(L - 1 + m). \tag{10}$$

Step 7. The Column-wise inputs are applied to the network and the $fnet_j$ is calculated by using Eq. 11. The maximum $fnet_j$ is the winner neuron.

$$fnet_j = \sum_{i=1}^{m} b_{ij} x_i, \tag{11}$$

where x_i = input vector.

Step 8. Repeat Step 7 for all the columns.

Step 9. Based on the grouped columns the T_j weights are fixed and the B_j weights are fixed by using Eq. 10.

Step 10. The row-wise inputs are applied to the network and the $fnet_j$ is calculated by using Eq. 11. The maximum $fnet_j$ is the winner neuron.

Step 11. Repeat Step 10 for all the rows.

Step 12. If the current number of row groups are identical to previous number of row groups then go to Step 13, otherwise go to Step 6.

Step 13. The previous row-wise input weights are taken.

Step 14. Once again row-wise inputs are applied to the network and the exceptional elements (EE) are calculated by using the following equation:

$$EE = \sum_{n=1}^{N} \left(\sum fnet_j - \max\left(fnet_j\right)\right)/b_{ij}. \tag{12}$$

Where N = the total number of rows (i.e., machines) and b_{ij} = the fixed bottom-up weight value (i.e., $b_{ij} = L/L - 1 + m$).

3 Numerical illustration

The proposed modified ART1 network was tested on a matrix used by Boctor's [6] example problem. Table 1 shows the incidence matrix and the size of the matrix is 7×11, where eleven parts use seven machine types. Therefore, eleven neurons are required for the row-wise input.

The weights are initialized by using Eqs. 3 and 4. The L value 2 is selected and the ρ value is taken as 0.1.

$$b_{ij} = 2/(2 - 1 + 11) = 0.16 \quad \text{for all } i \text{ and } j\,,$$
$$t_{ji} = 1 \quad \text{for all } i \text{ and } j\,.$$

The row-wise inputs are applied to the network. The input row number, winner neuron, x_i, t_{ji} and c_i are shown in Table 2. In Table 2, the last column is checking whether the number of 1s increases in c_i when compared to the t_{ji}. If there is an increase then c_i is given as the input to the network and the winner is identified. In this example the same stored neuron is the winner. So there is no change in the stored pattern.

The training weights t_{ji} and b_{ij} are calculated by using Eqs. 8 and 9. The weights for each row and corresponding winner are shown in Table 3.

The row groups are identified and the grouped neurons are "0", "1" and "2". For the "0" neuron the assigned rows are 1, 5 and 6. For the "1" neuron the assigned rows are 2 and 3 and for the "2" neuron the assigned rows are 4 and 7.

Based on the grouped parts the t_{ji} weights are fixed and the b_{ij} weights are calculated by using Eq. 10. The weights are shown in Table 4 and column-wise inputs are applied to the network.

The column groups are identified and the total number of groups is 3. For the neuron "0" the grouped columns are 1, 3, 4, 7 and 11. For the neuron "1" the grouped columns are 2, 6 and 9. The grouped columns for the neuron "2" are 5, 8 and 10.

Once again, the t_{ji} weights are fixed based on the grouped columns and the b_{ij} are calculated by using Eq. 10. The weights

Table 1. Machine-part incidence matrix

		Parts										
		1	2	3	4	5	6	7	8	9	10	11
Machines	1	1	0	1	0	0	0	1	0	0	0	1
	2	1	1	0	0	0	1	0	0	0	0	0
	3	0	1	0	0	0	1	0	0	1	0	0
	4	0	0	0	1	1	0	0	0	0	1	0
	5	0	0	1	0	0	0	1	0	0	0	0
	6	0	0	1	1	0	0	0	0	0	0	1
	7	0	0	0	0	1	0	0	1	0	1	0

Table 2. First stage output of the modified ART1

Row no.	Winner	Row-wise input (x_i)	Top-down weights (t_{ji})	Output (C_i)	Checking (If $\sum C_i > \sum t_{ji}$)
1	0	1 0 1 0 0 0 0 1 0 0 0 1	New winner	1 0 1 0 0 0 0 1 0 0 0 1	
2	1	1 1 0 0 0 1 0 0 0 0 0 0	New winner	1 1 0 0 0 1 0 0 0 0 0 0	
3	1	0 1 0 0 0 1 0 0 1 0 0	1 1 0 0 0 1 0 0 0 0 0	1 1 0 0 0 1 0 0 1 0 0	Yes. Then C_i is input to the network
	1	1 1 0 0 0 1 0 0 1 0 0	Same winner. So the same stored pattern is maintained		
4	2	0 0 0 1 1 0 0 0 0 1 0	New winner	0 0 0 1 1 0 0 0 0 1 0	
5	0	0 0 1 0 0 0 1 0 0 0 0	1 0 1 0 0 0 1 0 0 0 1	1 0 1 0 0 0 1 0 0 0 1	
6	0	0 0 1 1 0 0 0 0 0 0 1	1 0 1 0 0 0 1 0 0 0 1	1 0 1 1 0 0 1 0 0 0 1	Yes. Then C_i is input to the network
	0	1 0 1 1 0 0 1 0 0 0 1	Same winner. So the stored pattern is maintained		
7	2	0 0 0 0 1 0 0 1 0 1 0	0 0 0 1 1 0 0 0 0 1 0	0 0 0 1 1 0 0 1 0 1 0	Yes. Then C_i is input to the network
	2	0 0 0 1 1 0 0 1 0 1 0	Same winner. So the same stored pattern is maintained		

Table 3. First stage weights of the modified ART1

Row no.	Winner	Top-down weights (t_{ji})	Bottom-up weights (b_{ij})
1	0	1 0 1 0 0 0 1 0 0 0 1	0.40 0.00 0.40 0.00 0.00 0.00 0.40 0.00 0.00 0.00 0.40
2	1	1 1 0 0 0 1 0 0 0 0 0	0.50 0.50 0.00 0.00 0.00 0.50 0.00 0.00 0.00 0.00 0.00
3	1	1 1 0 0 0 1 0 0 1 0 0	0.40 0.40 0.00 0.00 0.00 0.40 0.00 0.00 0.40 0.00 0.00
4	2	0 0 0 1 1 0 0 0 0 1 0	0.00 0.00 0.00 0.50 0.50 0.00 0.00 0.00 0.00 0.50 0.00
5	0	1 0 1 0 0 0 1 0 0 0 1	0.40 0.00 0.40 0.00 0.00 0.00 0.40 0.00 0.00 0.00 0.40
6	0	1 0 1 1 0 0 1 0 0 0 1	0.33 0.00 0.33 0.33 0.00 0.00 0.33 0.00 0.00 0.00 0.33
7	2	0 0 0 1 1 0 0 1 0 1 0	0.00 0.00 0.00 0.40 0.40 0.00 0.00 0.40 0.00 0.40 0.00

Table 4. Second stage fixed weights based on the grouped columns

Neuron no.	Top-down weights (t_{ji})	Bottom-up weights (b_{ij})
0	1 0 0 0 1 1 0	0.25 0.00 0.00 0.00 0.25 0.25 0.00
1	0 1 1 0 0 0 0	0.00 0.25 0.25 0.00 0.00 0.00 0.00
2	0 0 0 1 0 0 1	0.00 0.00 0.00 0.25 0.00 0.00 0.25

are shown in Table 5. The row-wise inputs are applied to the network and the total number of groups is 3.

The numbers of grouped rows are identical when compared to the previous grouped rows. Once again the row-wise inputs are applied to the network. The previous fixed weights are taken and the exceptional elements are calculated by using the Eq. 12. The total number of exceptional elements is 2.

The total number of row groups and column groups is 3. Therefore the total number of GT cell is 3. For the GT cell number "1" the first row groups and first column groups are assigned. Similarly for the second and third GT cells, the corresponding row and column groups are assigned. The number of exceptional elements is identical to Boctor's [6] solution, but the groups are different.

4 Experimentation on data from literature

The proposed algorithm has been coded in C++ and executed in a Pentium III, 700 MHz system. For all the tested problems the ρ value is taken as 0.1. The first set of problems was solved based on problems solved by Boctor [6]. The problem solved by Boctor [6], involves an incidence matrix for ten problems with 16 rows (machines) and 30 columns (parts). By using the simulated annealing algorithm (SA) the data was tested by Boctor [6]. The same data were given as input to the modified ART1 network in the same manner. The solution given by the modified ART1 does not meet the Boctor's [6] constraints. The constraints are the number of cells and maximum machines per-

Table 5. Second stage fixed weights based on the grouped rows

Neuron no.	Top-down weights (t_{ji})	Bottom-up weights (b_{ij})
0	1 0 1 1 0 0 1 0 0 0 1	0.16 0.00 0.16 0.16 0.00 0.00 0.16 0.00 0.00 0.00 0.16
1	0 1 0 0 0 1 0 0 1 0 0	0.00 0.16 0.00 0.00 0.00 0.16 0.00 0.00 0.16 0.00 0.00
2	0 0 0 0 1 0 0 1 0 1 0	0.00 0.00 0.00 0.00 0.16 0.00 0.00 0.16 0.00 0.16 0.00

mitted to each cell. After combining one or two outputs of the modified ART1 GT cells the Boctor's [6] constraints are satisfied and the solutions are identical to the optimal solution given by Boctor [6]. The combining operation is also done by using the modified ART1 second stage. The GT cells are combined and the weights are fixed based on the clusters. After combining, the proposed modified ART1 solution is superior in four problems and identical in all other problems when compared to the SA algorithm. Zhao and Wu [7], using the genetic algorithm (GA), tested the same problems. Zhao and Wu [7] also gave the average computational time for GA as 2 min. The modified ART1 is also compared with the genetic algorithm and after combining one or two clusters the results are superior for four problems and identical to all other problems. The computational time is also very low for the modified ART1. The comparative results with computational time are shown in Table 6.

The next tested data set is 36 rows (machines) and 90 columns (parts). This data set adopted from King and Nakornchai [8] was used to further validate the functioning of modified ART1. The modified ART1 performs extremely well, producing results that are very much comparable to the rank order clustering 2 (ROC2) algorithm proposed by King and Nakornchai [8] and modified procedure in ART1 proposed by Dagli and Huggahalli [4]. The ROC2 algorithm produced, a solution that has 18 duplications with six remaining exceptional elements and Dagli and Huggahalli [4] obtained 3 GT cells with 21 duplications and 11 exceptional elements using a modified procedure in ART1. However, this modified ART1 has resulted in 3 GT cells with 12 duplications and 12 exceptional elements, which is shown in Fig. 2. The duplication machines are also identified by using this modified ART1. If any row requires more than one exceptional element then that GT cell will require the particular row (i.e., machine) duplication. That can be identified using step 14. Thus the modi-

Table 6. Comparison of results for Boctor's [6] ten problems

Problem no.	Boctor's [6] Constraints No. of cells	Max. no. of machines	Exceptional elements Optimum	SA	Zhao and Wu [7] GA	*EE* after combining with Boctor's [6] constraints	Modified ART1 Without clustering No. of cells	No. of machines Max.	Min.	*EE*	CPU time (s)
1.	2	8	11	11	11	11	2	8	8	11	0.03
2.	2	12	3	3	3	3	3	7	4	6	0.01
3.	2	12	1	4	4	1	4	5	3	8	0.01
4.	2	9	13	13	13	13	3	7	3	23	0.01
5.	2	12	4	4	4	4	4	5	3	11	0.01
6.	2	12	2	3	3	2	4	5	2	6	0.01
7.	2	12	4	4	4	4	4	7	3	9	0.01
8.	2	11	5	11	11	5	3	7	4	11	0.01
9.	2	11	5	8	8	5	3	8	3	9	0.01
10.	2	12	5	5	5	5	3	7	3	8	0.00

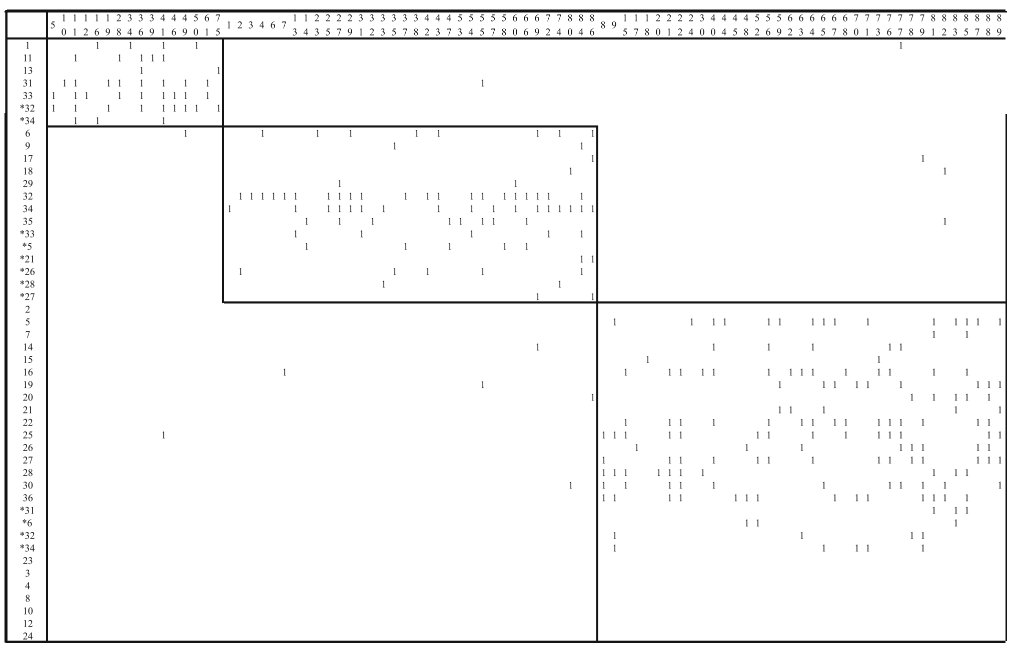

Fig. 2. 36 × 90 matrix–block diagonal form after duplication

Table 7. GT matrices selected from the literature

Problem no.	Size	Source
1.	7×11	Boctor [6]
2.	20×35	Carrie [9]
3.	9×9	Gangaware and Ham [10]
4.	10×15	Chan and Milner [11]
5.	15×10	Chan and Milner [11]
6.	8×20	Chanrasekaran and Rajagopalan [12]
7.	40×100	Chanrasekaran and Rajagopalan [13]
8.	13×25	Leskowsky, Logan and Vannelli [14]
9.	10×15	Kusiak and Lee [15]
10.	24×40	Chandrasekaran and Rajagopalan [16]
11.	24×40	Chandrasekaran and Rajagopalan [16]
12.	16×30	Srinivasan, Narendran and Mahadevan [17]
13.	8×10	Melody Kiyang [18]
14.	10×11	Ravichandran and Chandra Sekhara Rao [19]

fied ART1 shows classification accuracy that is superior to those of previous algorithms. Dagli and Huggahalli [4] also used separate method to find the exceptional elements and duplication machines. However, this modified ART1 gives not only machine and part groups, it also gives the exceptional elements and duplication machines.

In addition to the above tested problems 14 matrices including well-structured and not so well structured matrices have been considered for evaluation. The matrix sizes and their sources are listed in Table 7.

It is found that the proposed modified ART1 algorithm results are identical (without combining) in number of exceptional elements, number of GT cells, part groups and machine groups for the ten problems when compared to all other algorithms. The results of four problems are superior, but they do not meet the source problem constraints (i.e., number of GT cells). The modified ART1 result groups decrease with number of exceptional elements. The modified ART1 results with computational time and literature techniques with results are shown in Table 8.

5 Future work

Several improvements to the modified ART1 are possible. The scope of this paper is restricted to the modified procedure for the ART1 for cell formation with the single objective of minimizing the exceptional elements. Some of the issues like constraints, multi objectives and large size of matrices can be implemented in this modified ART1.

6 Conclusion

The modified ART1 neural network has been successfully implemented for cell formation problems. The results are compared with popular existing algorithms and found that the modified ART1 solution is superior to others. The modified ART1 gives parts and machine clusters and the number of exceptional elements. The computational effort is very low in the modified ART1 when compared to all other algorithms. Also most of the conventional and non-conventional techniques give different results for each and every run. However, the modified ART1 gives the same result for all the runs. 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 and the Department of Production Engineering of the Regional Engineering College, Thiruchirapalli for their support of this research work.

Table 8. Exceptional elements from modified ART1 for selected incidence matrices

Problem no.	Source results: No. of cells	*EE*	Techniques	Proposed by	Modified ART1 results: No. of cells	*EE*	CPU time (s)
1.	3	0	Iri's algorithm	Boctor [6]	3	0	0.00
2.	4	2	Numerical taxonomy	Carrie [9]	4	2	0.01
			Carpenter-Grossberg network	Kaparthi and Suresh [2]			
3.	3	7	Fuzzy clustering approach	Chao-Hsien Chu and Jack Hayya [20]	2	4	0.00
4.	3	6	Cutting plane algorithm	Crama and Oostern [21]	3	6	0.00
5.	3	0	Heuristic algorithm	Aravind and Harold [22]	3	0	0.01
6.	3	9	Cutting plane algorithm	Crama and Oosten [21]	3	9	0.00
			Assignment model	Srinivasan, Narendran and Mahadevan [17]			
			Nonhierarchical clustering	Srinivasan and Narendran [23]			
7.	10	37	Zodiac	Chandrasekaran and Rajagopalan [13]	6	31	0.05
8.	3	9	Cutting plane algorithm	Crama and Oostern [21]	3	9	0.01
9.	3	4	Neural network approach	Kusiak and Lee [15]	3	4	0.00
			LINDO software (Scharge [25])	Kusiak [24]			
10.	7	19	Adaptive GA	Mak, Wong and Wang [26]	5	16	0.01
11.	7	0	ROC	Chandrasekaran and Rajagopalan [16]	7	0	0.00
12.	4	20	Adaptive GA	Mak, Wong and Wang [26]	3	16	0.00
13.	2	5	Kohonen self-organizing map networks	Melody Kiang [18]	2	5	0.00
14.	3	0	Fuzzy approach	Ravichandran and Chandra Sekhara Rao [19]	3	0	0.00

References

1. Moon Y (1990) Interactive activation and competition model for machine-part family formation. Proc International Conference on Neural Networks, Washington DC 2:667–670
2. Kaparthi S, Suresh NC (1992) Machine-component cells formation in-group technology: a neural network approach. Int J Prod Res 30(6):1353–1367
3. Malave CO, Ramachandran S (1991) Neural network-based design of cellular manufacturing systems. J Intell Manuf 2(5):305–314
4. Dagli C, Huggahalli R (1995) Machine-part family formation with the adaptive resonance theory paradigm. Int J Prod Res 33(4):893–913
5. Carpenter GA, Grossberg S (1987) A massively parallel architecture for a self-organizing neural pattern recognition machine. Comput Vision, Graph Image Process 37:54–115
6. Boctor FF (1991) A linear formulation of the machine-part cell formation problem. Int J Prod Res 29(2):343–356
7. Zhao C, Wu Z (2000) A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. Int J Prod Res 38(2):385–395
8. King JR, Nakornchai V (1982) Machine-component group formation in group technology review and extension. Int J Prod Res 20(2):117–133
9. Carrie AS (1973) Numerical taxonomy applied to group technology and plant Layout. Int J Prod Res 24:399–416
10. Gongaware TA, Ham I (1981) Cluster analysis applications for group technology-manufacturing system. Proc 9th North American Manufacturing Research Conference (NAMRC) on Society of Manufacturing Engineers, pp 503–508
11. Chan HM, Milner DA (1982) Direct clustering algorithm for group formation in cellular manufacture. J Manuf Syst 1:65–74
12. Chandrasekaran MP, Rajagopalan R (1986) MODROC-an extension of rank orders clustering for group technology. Int J Prod Res 24:1221–1233
13. Chandrasekaran MP, Rajagopalan R (1987) Zodiac-an algorithm for concurrent formation of part-families and machine cells. Int J Prod Res 25:835–850
14. Leskowsky Z, Logan L, Vannelli A (1987) Group technology decision aids in an expert system for plant layout, in modern production management system. Proc, IFIP TC 5/WG 5-7 Working conferences on Advances in Production Management System, pp 561–585
15. Kusiak, Lee H (1996) Neural computing-based design of components for cellular manufacturing. Int J Prod Res 34(7):1777–1790
16. 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
17. Srinivasan G, Narendran TT, Mahadevan B (1990) An assignment model for the part-families problem in Group Technology. Int J Prod Res 28(1):145–152
18. Kiang MY (2001) Extending the Kohonen self organizing map networks for clustering analysis. Comput Stat Data Anal 38:161–180
19. Ravichandran KS, Chandra Sekhara Rao K (2001) A new approach to fuzzy part-family formation in cellular manufacturing system. Int J Adv Manuf Technol 18:591–597
20. Hsien C, Hayya JC (1991) A fuzzy clustering approach to manufacturing cell formation. Int J Prod Res 29(7):1475–1487
21. Crama C, Oostem M (1996) Models for machine part grouping in cellular Manufacturing. Int J Prod Res 34(6):1693–1713
22. Ballakur A, Steudel HJ (1987) A within-cell utilization based heuristic for designing cellular manufacturing systems. Int J Prod Res 25(5):639–665
23. Srinivasan G, Narendran TT (1991) GRAFICS-a nonhierarchical clustering algorithm for group technology. Int J Prod Res, 20:463–478
24. Kusiak A (1987) The generalized group technology concept. Int J Prod Res 25:561–569
25. Schrage L (1984) Linear, Integer and Quadratic Programming with LINDO. Scientific Press, Palo Alto, CA
26. Mak KL, Wong YS, Wang XX (2000) An adoptive genetic algorithm for manufacturing cell formation. Int J Adv Manuf Technol 16:491–497
27. Lozana S, Onieva L, Larraneta J, Teba J (1993) A neural network approach to part-machine grouping in GT manufacturing. In Takamori T, Tsuchiya K (eds) Proc Robotics, Mectronics and Manufacturing Systems. North-Holland, Amsterdam, pp 619–624