A METAHEURISTIC APPROACH FOR GENERALIZED CELL FORMATION WITH RATIO-LEVEL DATA

Dr. S.S.Mahapatra[1] **R.Sudhakara Pandian[2]**

Department of Mechanical Engineering, National Institute of Technology, Rourkela-769008. INDIA
Phone: (0661) 2462512 Fax: (0661) 2462999

1ssm@nitrkl.ac.in *[2]sudhakarme@rediffmail.com*

Abstract: *Cellular layout is a well-known solution method to the layout problems in modern batch type production industries. In Cellular manufacturing system (CMS), machines and parts are grouped together based on manufacturing similarities. Generally, zero-one machine-part incidence matrix (MPIM) obtained from the route sheet information is used to form machine cells. The major demerit with such approaches is that real life production factors cannot be accounted for. In an attempt to address generalized cell formation problem, this study considers operational time of the parts processed on each machine. An efficient algorithm based on Genetic Algorithm (GA) is suggested to form machine cells with the combined objective of minimizing total cell load variation and exceptional elements when workload matrix is presented as input. A new grouping efficiency measure called Modified Grouping Efficiency (MGE) is proposed to evaluate the performance of GA. The proposed algorithm is tested with varied sizes of problems from open literature.*

Key words: *Cell formation, Genetic Algorithm, Grouping Efficiency, ART1.*

1.0 INTRODUCTION

Cellular manufacturing is a manufacturing philosophy where the similar parts are grouped together based on design and/or manufacturing attributes. The basic problem in cellular manufacturing is to group the machines into machine cells and the parts into part families. Major approaches in cell formation problems can be categorized as visual inspection, parts classification and coding and production flow analysis. However, the production flow analysis is quite popular method in industries because existing production data can be utilized to form machine cells. The major advantages of cellular manufacturing system can be listed as reduced material handling, manufacturing lead time, work-in-process, setup time, and increased flexibility and utilization of resources. The past research work reveals that the cell formation problems are addressed with zero-one incidence matrices in most cases (figure 1a). In order to develop a generalized model for cell formation problem, it is essential to consider the production data like lot size of the products, machine capacity, operation time and sequence. In this paper, an attempt has been made to solve the problem with operation time of the parts**.** Meta-heuristics like Genetic Algorithm, Simulated Annealing, Tabu search and Ants Colony Systems seem to be prominent algorithms for cell formation problems using MPIM. Therefore, it is highly desirable to test the performance of meta-heuristics using real valued matrices. To this end, an algorithm based on most commonly used meta-heuristic known as Genetic Algorithm is proposed. The model has been tested using wide variety of problems from literature and found to be consistent in producing good results.

2.0 LITERATURE REVIEW

Burbidge [1] views group technology (GT) as a change from an organization of people mainly on process, to an organization based on completed products, components and major completed tasks. From 1960 onwards there are six basic approaches cited in literature. They are listed as similarity coefficient methods, graph theory, mathematical programming, meta-heuristics, fuzzy set theory and neural networks frequently used to solve cell formation problems. Similarity coefficient methods (SCM), rank order clustering (ROC) [2] and graph theory [3] methods were initially used to form machine cells followed by MODROC [4] the extended basic rank order clustering method and ZODIAC[5] for concurrent formation of part families and machine cells. Heuristics based MACE [6]. GRAFICS [7] are some well known methods found in literature. Meta-heuristics based Simulated Annealing algorithm applied to cell formation was developed by F.F.Boctor [8]. Venugopal et al. [9] adopted genetic algorithm model to multi-objective cell formation problems. Jayakrishnannair et al.'s CASE [10] considers sequence of operations that a part undergoes through a number of machines. Fernando et al. [11] introduced a new Hybrid Genetic Algorithm by combining local search algorithm with genetic algorithm. Wu et al. [12] proposed two methods based on Tabu search for small problems. Moon [13] introduced back propagation neural network model for (GT). The adaptive resonance theory (ART1), proposed by Carpenter et al. [14], is an example of unsupervised learning. Kaparthi et al. [15] made an attempt to apply adaptive resonance theory (ART1) to cell formation problems. Kumar et al. [16] proposed grouping efficacy as a performance measure for the block diagonal forms of binary matrices.

3.0 GENETIC ALGORITHM

Genetic Algorithm (GA) is computerized search and optimization algorithm based on the mechanics of natural genetics and natural selection. GA is a search technique for global optimization in a search space. GAs are different from traditional optimization and search techniques in the following ways.

- GA works with a coding of parameters, not with parameter themselves.
- GA searches from population of points, not a single point.
- GA uses information of fitness function not derivatives or other auxiliary knowledge.
- GA uses probabilistic rules rather than deterministic rules.

In Genetic Algorithm, a candidate solution represented by sequence of genes called chromosome. A chromosome's potential is called its fitness value, which is evaluated by the objective function. A set of selected chromosomes is called population and the population is subjected to generations (number of iterations).

4.0 OBJECTIVE FUNCTION

The objective of this study is to minimize both total cell load variation and the exceptional elements bringing them to a common scale. The cell load variation is calculated as the difference between the workload on the machine and the average load on the cell [9]. The exceptional elements are identified from the off diagonal non-zero values. The visit of the parts to the machines has been denoted in terms of their workload known as ratio level data (as shown in figure 1b) on the machines for the computation of cell load variation. The first component of Eq. (1) expanded in Eq. (2) represents the ratio between square root of the total cell load variation and the total workload of the matrix. The second component shown in Eq. (3) indicates the ratio of number of exceptional elements and the total elements of the matrix. In view of equal importance for both the objectives, the values of the weights are assigned to 0.5 to each.

Minimize

$$Z = q_1 Z_1 + q_2 Z_2 \tag{1}$$

$$Z_1 = \sqrt{\sum_{i=1}^{m} \sum_{k=1}^{c} x_{ik} \sum_{j=1}^{p} (W_{ij} - M_{kj})^2} \, / \, W_T \tag{2}$$

$$Z_2 = \frac{1}{2} \sum_{k=1}^{c} \sum_{j=1}^{p} \sum_{i=1}^{m} |X_{ik} - Y_{jk}| a_{ij} / E_M \tag{3}$$

$$M_{kj} = \frac{\sum_{i=1}^{m} X_{ik} \cdot W_{ij}}{\sum_{i=1}^{m} X_{ik}} \tag{4}$$

where $\sum_{i=1}^{m} X_{ik} \cdot W_{ij}$ is the total cell load of cell k that induced by part j and $\sum_{i=1}^{m} X_{ik}$ is the total number of machines in cell k. For a predefined number of cell k, the Z value is calculated using the equation (1).

m – number of machines.($i = 1,2,3\ldots,m$)
p – number of parts.($j = 1,2,3\ldots,p$)
c – number of cells.($k = 1,2,3\ldots,c$)
q_1,q_2 – weights ($0<q_1<1$),($0<q_2<1$), $q_1+q_2=1$.

$[x_{ik}]$ – machine cell (m x k) membership matrix where
$x_{ik} = 1$ if i^{th} machine is in cell k,
$= 0$ otherwise.

$[Y_{jk}]$ – part cell membership matrix where
$Y_{ik} = 1$ if j^{th} part is in cell k,
$= 0$ otherwise.
$[W_{ij}]$ – machine part (m x p) matrix in terms of workload on machine i induced by part j.
$[a_{ij}]$ – machine part (m x p) binary matrix
$[M_{kj}]$ – cell part (k x p) matrix of average cell load.
W_T – Total workload of the matrix.
E_M – Total non-zero elements of the matrix

5.0 IMPLEMENTATION OF PROPOSED ALGORITHM TO GENERALIZED MACHINE CELL FORMATION

5.1 Representation

Representation forms a key role in the development of GA. A problem can be solved once it is represented in the form of solution string. In the problem, each gene represents cell number and its position gives machine number.

1 2 3 4 5 ⟶ position of the machine number
2 1 2 1 2 ⟶ cell number (gene)

This represents that the machine number 1 and 3 are in cell number two, the machine number 2 and 4 are in cell number one and the machine number 5 in cell number two. After representation, and initial solution can be generated using a set of algorithms and the generated solution is subjected to iterations or generations.

5.2 Reproduction

A fitness function value is computed for each string in the population and the objective is to find a string with the maximum fitness function value. Since objective is minimization, it is required to map it inversely and then maximize the resultant. Goldberg [17] suggested a mapping function given as,

F(t)-fitness function of t^{th} string $[F(t)= Z_{max} - Z(t)]$
Z_{max}-max [Z (t)] of all strings (t).

The advantage is that the worst strings get a fitness function value of zero and there is no chance of the worst strings getting reproduced into the next generation.

5.3 Crossover and Mutation

The crossover operator is carried out with a probability known as crossover probability. Crossover is nothing but exchange of a portion of strings at a point called crossover site (S). The two strings, which take part in the crossover operation, are also selected at random. Here swapping crossover is performed i.e., crossover site is selected and the genes of one string between the sites are swapped with genes of another string. Sometimes there is a chance of getting the string without required cells. To overcome this draw back, the procedure is modified such that in the resultant if all the cells are present in the offspring string then it will be considered for further steps otherwise the crossover site is changed from current site and the steps are repeated. If no strings get the required number of cells then the crossover operation is cancelled for the chosen parents and the algorithm steps are repeated.

Parent_1 1 2 | 2 1 2 Offspring_1 1 2 | 1 1 2
Parent_2 1 2 | 1 1 2 Off spring_2 1 2 | 2 1 2
S S

Mutation is also done randomly for each gene and it depends upon another parameter called mutation probability. Here one gene is selected at random and the mutation operation is performed. The mutation operation may consist of any of three operators given below.

Shifting: A gene or a machine going out from one cell and residing in another cell.

Inversion: A gene or a machine comes out from one cell and goes to another cell, while a machine from latter cell comes to the former cell.

Creating: It is a process in which a gene or machine goes from one cell and creates a new cell and resides there. In this problem inversion mutation is selected. Thus the genes are mutually interchanged.

2 1 1 2 2 → 2 1 2 1 2

5.4 Eliminating singleton machine cell

If single machine found in any cell, the following operations are carried out to merge the single machine cells with other cells. The average workload of each part in the cell and the Euclidean distance between the cells are calculated. The minimum Euclidean distance between cells is found out. Cells with single machine are merged to the cells which has minimum Euclidean distance.

5.5 Part Assignment

The following procedure given by [18] is used to assign parts into the machine cells. A machine cell which processes the part for a larger number of operations than any other machine cell is found out and the corresponding part is assigned into that cell. Ties are broken by choosing the machine cell which has the largest percentage of machines visited by the part. In the case of tie again the machine cell with the smallest identification number is selected. Thus all the parts are assigned to all the cells which form part families using membership index given in equation (5).

$$P_{kj} = \frac{f_{kj}}{f_k} \cdot \frac{f_{kj}}{f_j} \cdot \frac{T_{kj}}{T_j} \qquad (5)$$

P_{kj} - Membership index of part j belongs to cell k.
f_{kj} - Number of machines in cell k required by part j.
f_k -Total number of machines in cell k.
f_j -Total number of machines required by part j.
T_{kj} -Processing time of part j in cell k.
T_j-Total processing time required by part j.

5.6 Proposed GA

Notations used:

gen - Number of generations.
P_s - Population size.
P_a - Probability of acceptance of the solution.
P_m - Mutation probability.
P_c - Cross over probability.
P_t - Probability of selection of t^{th} string
r - Random number between 0 and 1

I. Initialization

Step 1: Set the values of Ps, gen, Pc, Pm.
Step 2: Read the workload given in terms of processing time w_{ij} of part j on machine i.
Step 3: Create an initial population of size Ps and call it old population (P_{old}).
Step 4: Do part assignment as given in section VI
Step 5: Calculate the objective function using equation (Z).
Step 6: Sort string in the increasing order of objective function value.
Step 7: Set gen = 0.

II. Reproduction

Step 1: Compute Fitness value for P_{old}.
Step 2: Compute P_t of each string.
Step 3: Find the cumulative probability of each string.
Step 4: Generate ‘r’ and select the string from P_{old} according to r and reproduce it in P_{new}.
Step 5: Repeat step 4 for P_s time.
Step 6: End.

III. Crossover

Step 1: Generate ‘r’ if ($r<P_c$) go to step 2 else go to step 4.
Step 2: Select two strings randomly and swap genes between them by selecting crossover site (S)
Step 3: Repeat step 2 for Ps/2 times.
Step 4: End.

IV. Mutation

Step 1: Generate ‘r’.
Step 2 : If ($r<P_m$) go to step 3 else go to step 1.
Step 3: Select two machines randomly in t^{th} string and interchange its positions.
Step 4: Repeat step 1 for all genes in P_{new}.
Step 5: End.

V. Eliminating machine cell with single machine

Step 1: Check for single machine cells. If single machine found in any cell, perform the following operations to merge the single machine cells with other cells.
Step 2: Calculate the average workload of each part in the cell.
Step 3: Calculate the Euclidean distance between the cells.
Step 4: Find out the minimum distance between cells.
Step 5: Merge the cell to the cell, which has minimum Euclidean distance.

VI. Part Assignment

Step 1: Find a machine cell which processes the part for a larger number of operations than any other machine cell and assign the part in that machine cell.
Step 2: If tie occurs, choose the machine cell which has the largest percentage of machines visited by the part and assign in that cell.
Step 3: If again tie occurs, select the machine cell with the smallest identification number and assign the part in that machine cell.
Step 4: End.

VII. Main Algorithm

Step 0: Define the number of cells
Step 1: Initialize the values, do part assignment as in section VI and evaluate the objective function as given in section I.
Step 2: Do Reproduction as given in section II.
Step 3: Do Crossover as given in section III.
Step 4: Do Mutation as given in section IV.
Step 5: Eliminate single machine cells as given in section V, if singleton exists.
Step 6: Do Part Assignment as given in sectionIV after eliminating singletons.
Step 7: For the first half of the P_{new}, accept the string ‘t’ if good else reject.
Step 8: For the next P_s /2, generate ‘r’ if ($r<P_a$) accept else reject.
Step 9: If (counter < gen) go to step 3 else step10.
Step 10: Stop.

6.0 WORKING OF THE PROPOSED ALGORITHM

The example problem of size 6 x 8 , as shown in figure 1, is considered for illustrating the working of the proposed algorithm. A randomly generated real valued workload matrix of size 6 x 8 is presented to the algorithm. First the values of Genetic Algorithm parameters and the number cells are initialized. The algorithm generates a solution after 250 generations with population size of 10. Thus the resultant matrix of 2 cells is as shown in (figure 1c).

7.0 MEASURE OF PERFORMANCE

To evaluate the cells grouped, the following performance measures have been generally used in the literature [16] (i) grouping efficiency and (ii) grouping efficacy.

7.1 Groping Efficiency

Grouping efficiency (η) was defined as the weighted average of two functions η 1and η 2.

$$\eta = (q_a \times \eta_1) + (1 - q_a)\eta_2 \tag{6}$$

$$\eta_1 = \frac{\text{Number of ones in the diagonal blocks}}{\text{Total number of elements in the diagonal blocks}} \tag{7}$$

$$\eta_2 = \frac{\text{Number of zeroes in the off-diagonal blocks}}{\text{Total number of elements in the off diagonal blocks}} \tag{8}$$

q is a weighting factor (0<q<1).This value lies in the range of zero to one. Higher the value of η indicates better clustering.

7.2 Grouping Efficacy

The drawback of grouping efficiency is its weak discriminating power. which means the ability to distinguish good quality grouping from bad quality grouping. For example, a bad solution with many 1s (ones) in the off-diagonal blocks often shows efficiency figures around 75%. When the matrix size increases, the effect of 1s in the off-diagonal blocks becomes smaller, and in some cases, the effect of inter-cell moves is not reflected in grouping efficiency. To overcome the shortcoming of grouping efficiency, there is another measure called grouping efficacy. It is defined as follows:

$$\tau = \frac{(1-\psi)}{(1-\phi)} \tag{9}$$

$$\Psi = \frac{\text{Number of exceptional elements}}{\text{Total number of operations}} \tag{10}$$

$$\varphi = \frac{\text{Number of voids in the diagonal blocks}}{\text{Total number of operations}} \tag{11}$$

Unlike grouping efficiency, grouping efficacy is not affected by the size of the matrix. However, both measures - grouping efficiency and grouping efficacy treat all operations equally and suitable only for the zero-one incidence matrix. These measures cannot be adopted for generalized cell formation problem where information regarding operational times is of importance. Therefore, generalized grouping efficacy introduced by Zolfaghari et al. [18] can be conveniently used to measure the performance considering operational times of the parts. But in contrast to grouping efficiency and grouping efficacy measures, generalized grouping efficacy ignores the effect of voids inside cells, which predominantly affects the goodness of the block diagonal structure. Hence, a new measure for grouping efficiency termed as modified grouping efficiency (MGE) has been proposed in this work to find out the performance of the cell formation method that deal with workload matrix with due consideration of voids inside the cells.

$$MGE = \frac{T_{pti}}{T_{pto} + \sum_{k=1}^{c} T_{ptk} + \sum_{k=1}^{c} T_{ptk} \cdot \frac{N_{vk}}{N_{ek}}} \tag{12}$$

T_{pti} - Total processing time inside the cells.
T_{pto} - Total processing time outside the cells.
T_{ptk} - Total processing time of cell k.
N_{vk} - Number of voids in cell k.
N_{ek} - Total number of elements in cell k.

Modified Grouping Efficiency has been calculated for the twenty four problems taken from literature and the comparison is made with GA and ART1 algorithm. The implications are discussed in the section 10.

8.0 RESULTS AND DISCUSSION

In this study, an efficient algorithm based on Genetic Algorithm has been proposed for cell formation problem considering operational time of the parts instead of conventional zero-one incidence matrix with the objective of minimizing total cell load variation. Number of problems with varied sizes form literature, as given in table 1, is considered for testing the proposed algorithm. The real valued matrix is produced by assigning random numbers in the range of 0.5 to 1 as uniformly distributed values by replacing the ones in the incidence matrix and zeros to remain in its same positions. The crossover probability and mutation probability have been fixed to 0.5 and 0.1 respectively. To tune the algorithm these values can be varied depending on size of problem. The number of generations is varied from problem to problem in the range of 250 to 1000. Similarly, the population size is varied in the range of 10 to 40 depending on the size of problem. A new method, Modified Grouping Efficiency, has been proposed to measure the performance of the grouping with real values. This method gives comfortable results. The results are compared with the results obtained from ART1 algorithm (table 1). The size of machine-part incidence matrices considered in this paper ranges from 5×7 to 30×50. The number of cells is varied from 2 to 6. The exceptional elements are comparatively reduced from other heuristics found in the literature. The algorithm is coded in C^{++} and tested on the Pentium IV machine.

9.0 CONCLUSION

The proposed algorithm is tested with varied sizes of problems from open literature and the solutions are compared with the solutions from ART1 algorithm. In most of the problems, it is observed that the solutions of the proposed model either outperform existing methods or remain the same. However, the proposed algorithm gives less number of exceptional elements and greater modified grouping efficiency (MGE). The work can be further extended in future incorporating production data like machine capacity, production volume and product sequence with varied product type, layout considerations and material handling systems enhancing it to more generalized manufacturing environment.

10.0 REFERENCES

1 **Burbidge.J.L.,** “A Manual Method of Production Flow Analysis”, The Production Engineer, 1977, Vol.56, pp (34-38).

2 **King,** “Machine Component grouping inproduction flow analysis: An approach using a rank order clustering algorithm”, International Journal of Production Research, 1980, Vol 18, (2) pp (213-232).

3 **Rajagopalan, Batra.** “Design of cellular manufacturing systems: a graph theoretic approach”, International Journal of Production Research, 1975, Vol 13, (6), pp (567-579).

4 **Chandrasekaran.MP, Rajagopalan.R.,** “MODROC: an extension of rank order clustering for group technology”, International Journal of Production Research, 1986, Vol 24, (5) pp (1221-1233).

5 **Chandrasekaran.MP, Rajagopalan.R** “ZODIAC: An algorithm for concurrent formation of part families and machine cells”, International Journal of Production Research, 1987, Vol 25, (6) pp (835-850).

6 **Waghodekar, P.H, Sahu, S.**, “Machine-component cell formation in group technology: MACE”, International Journal of Production Research, 1984, Vol 22, pp (937–948).

7 **Srinivasan,G, Narendran., T.T.** “A non hierarchical clustering algorithm for group technology GRAFICS”, International Journal of Production Research,1991, Vol 29, (3) pp (463-478).

8 **Boctor,F.F.,** “A linear formulation of the machine-part cell formation problem”, International Journal of Production Research, 1991, Vol 29 pp (343-346)

9 **Venugopal,V, Narendran.,** T.T. “A genetic algorithm approach to the machine component grouping problem with multiple objectives”, Computers and Industrial Engineering, 1992, Vol 22 pp (469-480).

10 **Jayakrishnannair.G., Narendran,** T.T. “CASE: A clustering algorithm for cell formation with sequence data”, International Journal of Production Research, 1998, Vol 36, pp (157-179).

11 **Jose Fernando, Maurico.,** “A Hybrid Genetic Algorithm in cell formation problems”, AT&T Labs Research Technical Report TD-5FE6RN, AT&T Labs Research, Florham Park, NJ 07932 USA, 2002.

12 **Wu, Low, Wu,** “A tabu search approach to the cell formation problem”, International Journal of Advanced Manufacturing Technology, 2004, Vol 23, pp (916-924).

13 **Kao.Y, Moon.B,** “A Unified Group Technology Implementation Using the Back propagation Learning Rule of Neural Networks”, Computers and Industrial Engineering, 1991, Vol 20, pp (425-437).

14 **Carpenter, Grossberg.S.,** “Adaptive Resonance Theory, A Handbook of Brain Theory and

Neural Networks”, Second Edition MIT Press, Massachusetts, 2002.

15 .**Kaparthi.S, Suresh.N.C,** “Machine Component cell formation in GT: A NN approach”, International Journal of Production Research, 1992, Vol 30, pp(1353-1367).

16 **Kumar.K.R, Chandrasekaran.M.P** “Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology”. International Journal of Production Research, 1990, Vol 28, (2), pp (233-243).

17 **Goldberg.D.E,** “Genetic Algorithms in Search, Optimization and Machine Learning”,Addison-Wesley, Reading, MA. 1989.

18 **Zolfaghari.S Liang.M,** “A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes”, Computers and Industrial engineering, 2003, Vol 45, pp (713-731).

Table 1. Performance of the proposed algorithm

S.N	Problem from Literature [11]	Problem Size	No of cells	No of EE*	No of EE†	MGE* %	MGE† %
1	King and Nakornchai (1982)	(5 x 7)	2	2	2	77.25	77.25
2	Waghodekar and sahu (1984)	(5 x 7)	2	2	2	78.34	78.34
3	Seiffodini (1989)	(5 x 18)	2	7	7	81.87	81.87
4	Kusiak (1992)	(6 x 8)	2	2	2	79.85	79.85
5	Kusiak (1987)	(7 x 11)	2	3	3	61.77	61.77
6	Boctor (1991)	(7 x 11)	2	1	1	65.48	65.48
7	Seiffodini and wolfe (1986)	(8 x 12)	2	4	6	69.70	62.11
8	Chandrasekaran et al. (1986)a	(8 x 20)	2	25	28	61.30	60.00
9	Chandrasekaran et al. (1986)b	(8 x 20)	3	9	9	83.40	83.40
10	Mosier et al. (1985)	(10 x 10)	3	0	0	77.14	77.14
11	Chan et al. (1982)	(10 x 15)	3	0	0	93.28	93.28
12	Askin et al. (1987)	(14 x 23)	2	2	0	60.59	62.42
13	Stanfel (1985)	(14 x 24)	4	7	3	68.13	73.19
14	Mccornick et al. (1972)	(24 x 16)	4	30	29	51.39	52.02
15	Srinivasan et al. (1990)	(16 x 30)	3	15	20	64.81	64.64
16	Mosier et al. (1985)	(20 x 20)	2	22	29	51.10	50.72
17	Carrie (1973)	(20 x 35)	3	1	1	71.15	71.15
18	Boe et al. (1991)	(20 x 35)	4	28	32	61.70	61.70
19	Kumar et al. (1986)	(23 x 20)	3	39	42	48.14	51.92
20	Chandrasekaran et al. (1989)a	(24 x 40)	6	0	0	90.28	84.58
21	Chandrasekaran et al. (1989)b	(24 x 40)	5	9	9	73.89	73.89
22	Kumar et al. (1987)	(30 x 41)	3	17	15	53.98	56.14
23	Stanfel (1985)a	(30 x 50)	6	26	22	55.51	60.23
24	Stanfel (1985)b	(30 x 50)	3	17	25	53.19	52.35

*- ART1 Algorithm. †- Proposed Genetic Algorithm.
EE - Exceptional Elements. MGE - Modified Grouping Efficiency.

Fig. 1a. Machine – Part incidence matrix of size 6x8

i/j	1	2	3	4	5	6	7	8
1	0	1	0	1	0	0	1	0
2	1	1	1	0	1	1	1	1
3	0	0	1	0	0	1	0	1
4	0	0	0	1	0	0	1	0
5	1	0	1	0	1	1	0	1
6	0	0	0	1	0	0	1	0

Fig. 1b. Real valued workload matrix

i/j	1	2	3	4	5	6	7	8
1	0	0.53	0	0.99	0	0	0.83	0
2	0.91	0.82	0.83	0	0.91	0.92	0.86	0.97
3	0	0	0.79	0	0	0.56	0	0.88
4	0	0	0	0.53	0	0	0.51	0
5	0.98	0	0.83	0	0.71	0.58	0	0.54
6	0	0	0	0.54	0	0	0.74	0

Fig. 1c. Output matrix

i/j	4	7	1	2	3	5	6	8
1	0.989	0.83	0	0.526	0	0	0	0
4	0.528	0.514	0	0	0	0	0	0
6	0.54	0.744	0	0	0	0	0	0
2	0	0.859	0.913	0.823	0.832	0.908	0.916	0.974
3	0	0	0	0	0.787	0	0.561	0.884
5	0	0	0.975	0	0.83	0.708	0.583	0.54

Figure 1 Working of the proposed algorithm