Genetic cell formation using ratio level data in cellular manufacturing systems

S. S. Mahapatra · R. Sudhakara Pandian

Abstract Manufacturing cell formation is a useful strategy in batch type production industries for enhancing productivity and flexibility. The basic idea rests on grouping the parts into part families and the machines into machine cells. Most of the literature used zero-one incidence matrix representing the part visiting a particular machine as one and zero otherwise. The output is generated in the form of block diagonal structure where each block represents a machine cell and a part family. In such models real life production factors such as operation time and sequence of operations are not accounted for. In this paper, the operational time of the parts required for processing in the machines is considered. It is attempted to develop an algorithm using genetic algorithm (GA) with a combined objective of minimizing the total cell load variation and the exceptional elements. The results are compared with the solutions obtained from K-means clustering and C-linkage clustering algorithms.

Keywords Exceptional elements · Genetic algorithm · Grouping efficiency

1 Introduction

Cellular manufacturing system (CMS) extensively uses the concept of group technology (GT) for design of efficient layout in batch type industries so that smooth flow of materials can be attained. GT is a manufacturing philosophy in which the similar parts are grouped together based on their design and manufacturing attributes. In CMS, machines are clustered into cells and parts are grouped into the same number of part families in such a way that all the parts in a family can be completely processed by a particular cell. Ranson [1] defines group technology in a total sense as the logical arrangement and sequence of all facets of company operation in order to bring the benefits of mass production into a high variety, mixed quantity production environment. Three major approaches viz. visual inspection, parts classification and coding and production flow analysis are found in literature for obtaining the GT layout. A cellular manufacturing system is primarily concerned with production flow analysis where machines and parts are grouped together based on manufacturing similarities.

In the past, researchers have proposed a good number of algorithms using evolutionary techniques, graph theory, heuristics for cell formation problems. Various methods like visual inspection, part classification and coding and production flow analysis can be used for solving the cell formation problem. Meta-heuristics-simulated annealing algorithm [2, 3], artificial neural network models [4] are few methods found in literature for CMS. Since cell formation problems are known as combinatorial optimization problems, it is difficult to obtain generalized solutions. Genetic algorithm (GA) and simulated annealing seem to be prominent algorithms to be tested for solution quality when applied to cell formation problems. Cellular manufacturing concept was introduced in order to split the manufacturing system into subsystems to gain some benefits. The benefits of implementing cellular manufacturing are, reduced set-up-time, reduced work-in-process

S. S. Mahapatra (✉) · R. S. Pandian
Department of Mechanical Engineering,
National Institute of Technology,
Rourkela, Orissa, India 769008
e-mail: mahapatrass2003@yahoo.com

R. S. Pandian
e-mail: sudhakarme@rediffmail.com

(WIP) and less inventory, greater manufacturing flexibility, less floor space required around the machines, simplified scheduling and simplification of management control. Nowadays multi-objective models are considered taking all such benefits into account, for example, ACCORD by Jayakrishnan et al. [5], hybrid GA by Jose Fernando et al. [6], and the memetic algorithm by Muruganandham et al. [7].

The objective of these algorithms is to optimize either one or combination of performance measures like intercell and intracell moves, grouping efficiency, exceptional elements and material handling time. Uniform workload distribution among the cells is an important aspect of CMS because non-uniform workload distribution among the cells may give rise to increase in work-in-process inventories and lead time. However, this vital aspect is not adequately addressed in the literature. Venugopal et al. [8] have proposed a methodology based on genetic algorithm that considers cell load variation as the objective of the algorithm. Since their aim is to minimize a multi-objective function, which is a combination of cell load variation and number of intercell moves, the rationalization of two objectives is highly desirable to bring both the objectives into same scale.

Cell load variation and number of exceptional elements are heavily dependent on number of cells. Increasing number of cells leads to increase in number of exceptional elements and reduce cell load variation whereas decreasing number of cells causes to lower the number of exceptional elements and increase cell load variation. Because of the conflicting nature of two objectives, a combined objective function is considered in this study with different weights assigned to each of the normalized objectives. The efficiency of the layout will also be affected unless a systematic procedure for part allocation is adopted. Therefore, structured method of part allocation into cells is followed in this study.

In addition, real life data like operational time, sequence of operations and lot size of the parts are not reflected when zero-one incidence matrix is considered. The process of clustering machines into cells and parts into part families without using such information may lead to inferior manufacturing plans [5]. Hence, the need arises to use non-binary data for obtaining groups or clusters of machines and parts [5, 8]. These data can be either ordinal-level or ratio-level. For the ratio-level data, it is common to make use of the workload information and obtain a modified incidence matrix. The total processing time of a part is computed as product of the production quantity of the part and its unit processing time. This workload (or ratio) is the value that replaces '1' in the incidence matrix. The resultant workload values can take any value in the ratio scale, and they constitute the ratio level data [9]. Similarly the sequence of operations can be captured in the incidence matrix by replacing the 1s in it by the appropriate sequence number. These can take values on an ordinal scale and hence called ordinal-level data. However, suitable measure to assess the goodness of cell formation is hardly found when real valued input pattern using ratio level data and sequence data is considered. In this study, an attempt has been made to address the following three issues:

- Minimization of the combined objective of cell load variation and exceptional elements in a normalized scale so that smooth flow of parts, reduction of work-in-process inventories and lead time can be achieved.
- Uniform allocation of parts following a patterned procedure, which also aids in achieving distribution of loads uniformly to machine cells and reduce the exceptional elements.
- A new measure, modified grouping efficiency (MGE), is proposed to find out the grouping performance with ratio level data.

As genetic algorithm (GA) has been tested successfully in cell formation problems [8, 10], the methodology uses GA in a wide range of problem sizes.

2 Genetic algorithm: an overview

Genetic algorithm (GA) is a 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. As the name suggests, they employ the concepts of natural selection and genetics using past information for directing the search with expected improved performance to achieve fairly consistent and reliable results. The traditional methods of optimization and search do not work well over a broad spectrum of problem domain. GA attempts to mimic the biological evolution process for discovering good solutions. They are based on a direct analogy to Darwinian natural selection and mutations in biological reproduction and belong to a category of heuristics known as randomized heuristics that employ randomized choice operators in their search strategy and do not depend on complete priori knowledge of the features of domain. These operators have been conceived through abstractions of natural genetic mechanisms such as crossover and mutation and have been cast into algorithmic forms. Holland [11] envisaged the concept of these algorithms in the mid-1960s and it has been applied in diverse areas such as music generation, genetic synthesis, strategy planning and also to address business problems, such as traveling salesman problem, production planning and scheduling problem, facility location problem and cell design problems. GA is different from traditional optimization and search techniques in the following ways. It works with a coding of parameters; not with parameter themselves. GA searches from population of points; not from a single point. It uses probabilistic rules rather than deterministic rules.

3 Objective model

Several objectives like intercell and intracell moves, grouping efficiency, and exceptional elements are associated with machine grouping problem as found in literature. But all these objectives hardly reflect smooth flow of materials and causes high work-in-process inventories. In order to achieve smooth flow of materials, less work-in-process inventories and increased productivity, cell load variation must be considered. The objective of this study is to minimize both the total cell load variation and the exceptional elements [2] bringing them to a common scale. The visit of the parts to the machines has been denoted in terms of their workload on the machines for the computation of cell load variation. The cell load variation is calculated as the difference between the workload on the machine and the average load on the cell [8]. The exceptional elements are found out by counting the number of non-zero values in off diagonal blocks. Thus the objective model of the cell formation problem is formulated as shown in Eq. (1). The first component 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 0.5 to each and combined to obtain a minimization function as shown in Eq. (1).

Minimize

$$Z = q_1 Z_1 + q_2 Z_2 \quad (1)$$

$$Z_1 = \frac{\sqrt{\sum_{i=1}^{m}\sum_{k=1}^{c} X_{ik} \sum_{j=1}^{p} \left(W_{ij} - m_{kj}\right)^2}}{T_w} \quad (2)$$

$$Z_2 = \frac{\frac{1}{2}\sum_{k=1}^{c}\sum_{j=1}^{p}\sum_{i=1}^{m} \left|X_{ik} - Y_{jk}\right| a_{ij}}{T_o} \quad (3)$$

i machine index (i = 1,2,3,....,m)
j part index (j = 1,2,3,....,p)
k cell index (k = 1,2,3,....,c)
q_1,q_2 weighting factors $0 \le q_1 \le 1$, $0 \le q_2 \le 1$, $q_1 + q_2 = 1$
T_w total workload of the matrix
T_o total elements of the matrix
$[X_{ik}]$ machine cell membership matrix where $X_{ik} = 1$ if i^{th} m/c is in cell k= 0 otherwise.
$[W_{ij}]$ machine component incidence matrix in terms of workload (ratio level data) on machine i induced by part j
$[Y_{jk}]$ part cell membership matrix where $Y_{jk} = 1$ if j^{th} part in cell k= 0 otherwise.
$[a_{ij}]$ machine part binary incidence matrix
$[m_{kj}]$ cell part matrix of average cell load.

$$\text{where} \quad m_{kj} = \frac{\sum_{i=1}^{m} X_{ik} \cdot W_{ik}}{\sum_{i=1}^{m} X_{ik}} \quad (4)$$

4 Methodology

Genetic algorithm (GA) is adopted to find out the machine clusters to form cells. The process of assigning parts in to the cells is discussed in Sect. 6. In GA a candidate solution represented by sequence of genes called chromosome. A chromosome potential is called its fitness function, 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). In each generation crossover and mutation operators (explained in Sect. 4.4) are performed to get new population.

4.1 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 [8].

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

4.2 Notations used

t String
m Number of machines

p Number of parts
c Number of cells
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
$Z_n(t)$ Objective function of t^{th} string in new population (P_{new})
$Z_o(t)$ Objective function of t^{th} string in old population (P_{old})
Z_{max} Max $\{Z(t)\}$
$F(t)$ Fitness function of t^{th} string $[Z_{max}-Z(t)]$
F^1 $\sum_{t=1}^{P_s} F(t)$
P_t Probability of t^{th} string
A $F(t)/F^1$
S Crossover site
r_a Random number between 0 and 1

4.3 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 [12] 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

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

4.4 Crossover and mutation

The crossover operator is carried out with a probability known as crossover probability. Crossover is 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 partial mapped crossover given by Michalewicz [13] is performed i.e., crossover site is selected and the genes of one string between the sites are swapped with genes of another string

1	2	2	1	2		1	2	1	1	2
1	2	1	1	2		1	2	2	1	2
S		Parent				S		Offspring		

Mutation [13] 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 It is a gene or a machine going out from one cell and residing in another cell.
Inversion In this, 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 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.

$$\begin{array}{c} \quad\downarrow\downarrow \\ 2\ 1\ 1\ 2\ 2 \end{array} \rightarrow 2\ 1\ 2\ 1\ 2$$

4.5 Part assignment

The following procedure given by Zolfaghari et al. [14] 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 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.

4.6 Proposed Genetic Algorithm

I. Initialization

Step 1 Set the values of P_s, gen, P_c, P_m.
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 Calculate the objective function using Eq. (1).
Step 5 Sort string in the increasing order of objective function value.
Step 6 Set gen = 0.

II. Reproduction

Step 1 Compute $F(t)$ for P_{old}.
Step 2 Compute P_t of each string.
Step 3 Find the cumulative of P_t.
Step 4 Generate 'r_a' 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 t1 and t2 and swap genes between them by selecting crossover site S randomly.
Step 3 Repeat step 2 for $P_s/2$ times.
Step 4 End.

IV. Mutation

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

V. Eliminating single machine cells.

Step 1 Check for single machine cells. If a single machine is found in any cell, Go to step 2
Step 2 Determine average workload of each cell.
Step 3 Calculate the Euclidean distance between the cells.
Step 4 Merge a cell containing single machine with another in such a way that Euclidean distance between them is minimum.

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 tie occurs again, 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 c = k. (k=2,3,....,m)
Step 1 Generate chromosomes randomly equal to the population size.
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 Do elimination of single machine cells as given in section V.
Step 6 Do part assignment as given in section VI.
Step 7 Evaluate the objective function as given in section I.
Step 8 Increment counter.
Step 9 If counter < gen, go to step 2 else step 10.
Step 10 Store the objective value in Z. Go to step 0. k = k+1.
Step 11 Print the best value of Z.
Step 12 Stop.

4.7 Convergence

Problem 1 (Table 1) of size 5×8 is taken as an example to illustrate the convergence curve during iterations. For the first iteration the objective value (×100) is to be 51.77. It gets reduced when the number of generation increases. At the 10th generation it reached the value of 30.35, a reduction of 41.37%. The final optimal solution is obtained during the 25th generation having a Z value (×100) of 5.58, a reduction of 81.67%. Based on the exhaustive experimentation for all the problems reported in Table 1, it is observed that the Z value is reduced when the number of generations increased till it reaches the global best value at some iteration and afterward the Z value continues to be constant even though the number of generations is increased. Since the methodology gives the same pattern of convergence for all the tested problems which proves the convergence property. The convergence curve is shown in Fig. 1.

5 Performance measures

There are two popular measures that are grouping efficiency and grouping efficacy used to check the performance of block diagonal structure generated by a cell formation technique. Grouping efficiency (η) as defined by Chandrasekharan et al. [15] is a weighted average of two functions η_1 and η_2.

$$\eta = (r \times \eta_1) + (1 - r)\eta_2 \tag{5}$$

$$\eta_1 = \frac{\text{Number of ones in the diagonal blocks}}{\text{Total number of elements in the diagonal blocks}} \tag{6}$$

$$\eta_2 = \frac{\text{Number of zeroes in the off} - \text{diagonal blocks}}{\text{Total number of elements in the off} - \text{diagonal blocks}} \tag{7}$$

r is a weighting factor that lies between zero to one ($0<r<1$) and its value is decided depending on the size of the matrix. Grouping efficiency considers two functions-packing density inside the cells (η_1) and intercell moves (η_2). Weighting factor is used to achieve a trade off between two functions depending on desirability of the decision maker. A higher value of η is supposed to indicate better clustering.

Table 1 Performance of the proposed GA

Problem	Size of sample Problems from sources [2, 8, 15]	Number of cells	K-means		C-link		Proposed GA	
			Z (×100)	MGE %	Z (×100)	MGE %	Z (×100)	MGE %
1	5×8	2	5.80	100.00	5.80	100.00	5.80	100.00
2	7×11	2	15.52	63.42	15.52	63.42	15.52	63.42
3	8×20	2	18.38	59.74	18.38	59.74	18.38	59.74
4	8×20	2	7.86	72.11	7.86	72.11	7.86	72.11
5	9×9	2	11.82	73.25	11.82	73.25	11.82	73.25
6	10×15	2	4.42	72.19	4.42	72.19	4.42	72.19
7	8×14	3	1.65	100.00	1.65	100.00	1.65	100.00
8	9×10	3	1.72	100.00	1.72	100.00	1.72	100.00
9	12×31	3	17.94	53.61	17.12	55.06	17.12	55.06
10	16×30	3	14.25	57.75	12.16	59.84	12.16	59.84
11	16×30	3	6.33	68.55	6.33	68.55	6.33	68.55
12	16×30	3	7.60	67.89	7.60	67.89	7.60	67.89
13	16×30	3	16.50	53.38	16.03	52.71	15.98	54.69
14	16×30	3	7.15	70.05	7.63	68.99	7.15	70.05
15	16×30	3	5.51	69.73	5.51	69.73	5.86	70.91
16	16×30	3	6.31	71.50	6.31	71.50	6.31	71.50
17	16×30	3	9.91	65.18	14.25	59.71	9.91	65.18
18	16×30	3	7.76	71.78	7.76	69.49	7.76	71.78
19	16×30	3	6.71	68.87	9.75	65.70	7.82	67.26
20	16×43	3	17.09	53.47	18.99	44.62	17.09	53.47
21	10×20	4	3.74	96.40	3.74	96.40	3.74	96.40
22	11×16	4	3.81	98.11	3.81	98.11	3.81	98.11
23	16×43	4	19.45	53.41	19.45	53.41	19.45	53.41
24	24×40	4	21.11	46.37	16.82	48.67	17.93	49.15
25	24×40	5	9.20	67.29	6.97	67.55	9.54	67.62

Z - Combined objective, MGE - modified grouping efficiency.

The first drawback of grouping efficiency is its low discriminating capability i.e., the ability to distinguish a good quality solution from a bad quality solution. A solution with many 1's (ones) in the off-diagonal blocks shows higher efficiency (range from 70% to 80%), which intuitively must show lower efficiency. An example illustration is given as shown in Fig. 2. Matrix 1 is ill structured compared to matrix 2. The efficiency value calculated using Eq. (5), produces around 72% efficiency for both the matrices. Secondly, emphasis on number of zeros in the off-diagonal blocks rather than number of 1s in Eq. (7) invariably leads to calculate a higher efficiency. This phenomenon is more closely observed when number of exceptional elements decreases with increase in size of the matrix. Therefore, it can be ascertained that grouping efficiency is highly sensitive to the size of the matrix. To

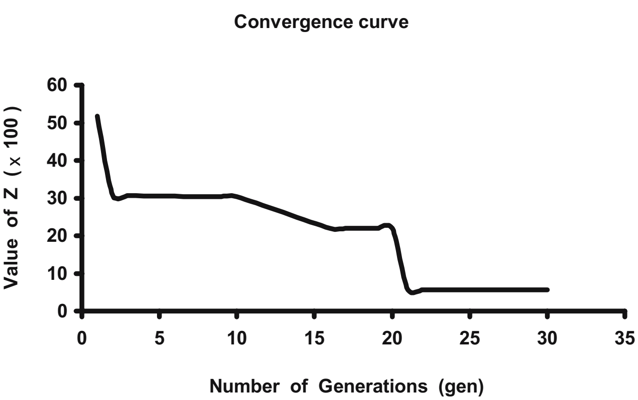


Fig. 1 Convergence curve

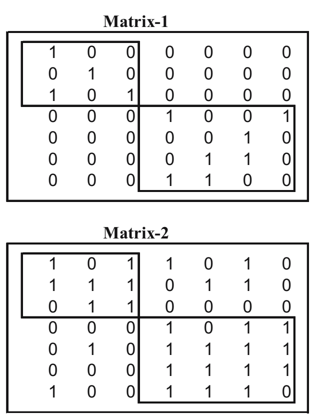


Fig. 2 Grouping efficiency illustration

overcome these shortcomings, grouping efficacy was proposed by Kumar et al. [16] as given in Eq. (8).

$$\tau = \frac{(1-\psi)}{(1-\phi)} \tag{8}$$

$$\psi = \frac{\text{Number of exceptional elements}}{\text{Total number of operations}} \tag{9}$$

$$\phi = \frac{\text{Number of voids in the diagonal blocks}}{\text{Total number of operations}} \tag{10}$$

Unlike grouping efficiency, one sees that grouping efficacy is not affected by the size of the matrix. However, both measures – grouping efficiency and grouping efficacy – treat all operations equally and are 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. [14] 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 deals with workload matrix with due consideration of voids inside the cells.

Modified grouping efficiency Weighting factor to the voids and modified grouping efficiency (MGE) are calculated using the Eqs. (11) and (12), respectively.

$$w_v = N_{vk}/N_{ek} \tag{11}$$

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

w_v weighting factor to the voids.
T_{pto} total processing time outside the cells.
N_{vk} number of voids in cell k.
T_{pti} total processing time inside the cells.
T_{ptk} total processing time of cell k.
N_{ek} total number of elements in cell k.

Unlike grouping efficiency, modified grouping efficiency does not treat all the operations equally. Moreover a weighting factor for voids is considered to reflect the packing density of the cells. It produces 100% efficiency when the cells are perfectly packed without any voids and exceptional elements.

6 Illustration of calculating MGE

Problem 6 (Table 1) is considered for illustration. The binary matrix of size 10×15 is converted into real valued workload matrix replacing the ones by randomly generated numbers in the range of 0.5 to 1 and zeros remain unchanged in the same position, and presented as input to the main algorithm. The algorithm produces output matrix from which block diagonal structure is obtained containing 2 cells as shown in Fig. 3.

Though voids (number of zeros inside the cells) play a role in reducing efficiency, in reality it is not reflected in the system. Hence they are considered by a weighting factor based on the total number of operations in the cell. The number of voids (N_{vk}) in the first cell is 31 and the total number of elements (N_{ek}) is 63. Hence weighting factor to the voids (N_{vk}/N_{ek}) is (31/63) 0.492. The total processing time in first cell (T_{ptk}) is 24.13 and multiplied by 0.4920 to produce the value of $T_{ptk} \times (N_{vk}/N_{ek})$ equal to 11.87. The number of voids in second cell (Nvk) is 3 and the total number of elements (N_{ek}) is 18. Thus, (N_{vk}/N_{ek}) becomes (3/18) 0.1667. The total processing time in second cell (T_{ptk}) is 11.78 and multiplied by 0.1667 to produce the value of $T_{ptk} \times (N_{vk}/N_{ek})$ equal to 1.96. The total processing time inside both the cells one and two (T_{pti}) is 35.91. Since there is no exceptional element (ones outside the cells) T_{pto} becomes zero. The summation of T_{pto}, T_{pti} and $\{T_{ptk} \times (N_{vk}/N_{ek})\}$ is 49.71. Now, the value of MGE is calculated by the ratio of 35.91 and 49.71. Hence the value of MGE obtained using the proposed method is 72.2%. It is observed from the calculation that the exceptional elements and voids are taken into consideration in the calculation of MGE. The increase in either exceptional elements or voids decrease MGE value and vice versa. It also produces 100% efficiency for a perfect block diagonal structure without any exceptional elements and voids (problems 1, 7, 8 in Table 1). Problem 24, MGE produces the value less than 50%. Through exhaustive experimentation twenty five bench mark problems of varied sizes have been tested and compared with standard heuristics K-means and C-linkage clustering methods. The results are encouraging and comparable to the literature.

Fig. 3 Illustration of calculating MGE for problem 6 (10×15)

0	1	0	0	0	0	1	0	0	0	1	1	0	0	0
0	0	0	0	1	0	0	0	1	1	0	0	1	0	1
1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
1	0	1	1	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	1	0	0	1	0	1
0	1	0	0	0	1	1	0	0	0	1	1	0	0	0
0	1	0	0	0	1	0	0	0	0	1	1	0	0	0
1	0	1	1	0	1	0	0	0	0	1	0	0	1	0
0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

Machine Part Incidence Matrix of size 10 x 15

0	0.53	0	0	0	0	0.99	0	0	0	0.83	0.91	0	0	0
0	0	0	0	0.82	0	0	0	0.83	0.91	0	0	0.92	0	0.86
0.97	0	0.79	0	0	0.56	0	0	0	0	0	0	0	0.88	0
0.53	0	0.51	0.98	0	0.83	0	0	0	0	0	0	0	0.71	0
0	0	0	0	0	0	0	0.58	0.54	0.54	0	0	0.74	0	0.63
0	0.63	0	0	0	0.53	0.69	0	0	0	0.63	0.68	0	0	
0	0.51	0	0	0	0.61	0	0	0	0	0.94	0.68	0	0	0
0.67	0	0.7	0.84	0	0.79	0	0	0	0	0.99	0	0	0.94	0
0	0	0.84	0.78	0	0.93	0	0	0	0	0	0	0	0.73	0
0	0	0	0	0	0	0	0.98	0.92	0.92	0	0	0.7	0	0.89

Real valued Input Matrix

0	0.53	0	0	0	0.99	0.83	0.91	0	0	0	0	0	0	0
0.97	0	0.79	0	0.56	0	0	0	0.88	0	0	0	0	0	0
0.53	0	0.51	0.98	0.83	0	0	0	0.71	0	0	0	0	0	0
0	0.63	0	0	0.53	0.69	0.63	0.68	0	0	0	0	0	0	0
0	0.51	0	0	0.61	0	0.94	0.68	0	0	0	0	0	0	0
0.67	0	0.7	0.84	0.79	0	0.99	0	0.94	0	0	0	0	0	0
0	0	0.84	0.78	0.93	0	0	0	0.73	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.82	0	0.83	0.91	0.92	0.86
0	0	0	0	0	0	0	0	0	0	0.58	0.54	0.54	0.74	0.63
0	0	0	0	0	0	0	0	0	0	0.98	0.92	0.92	0.7	0.89

Output Matrix

7 Results and discussion

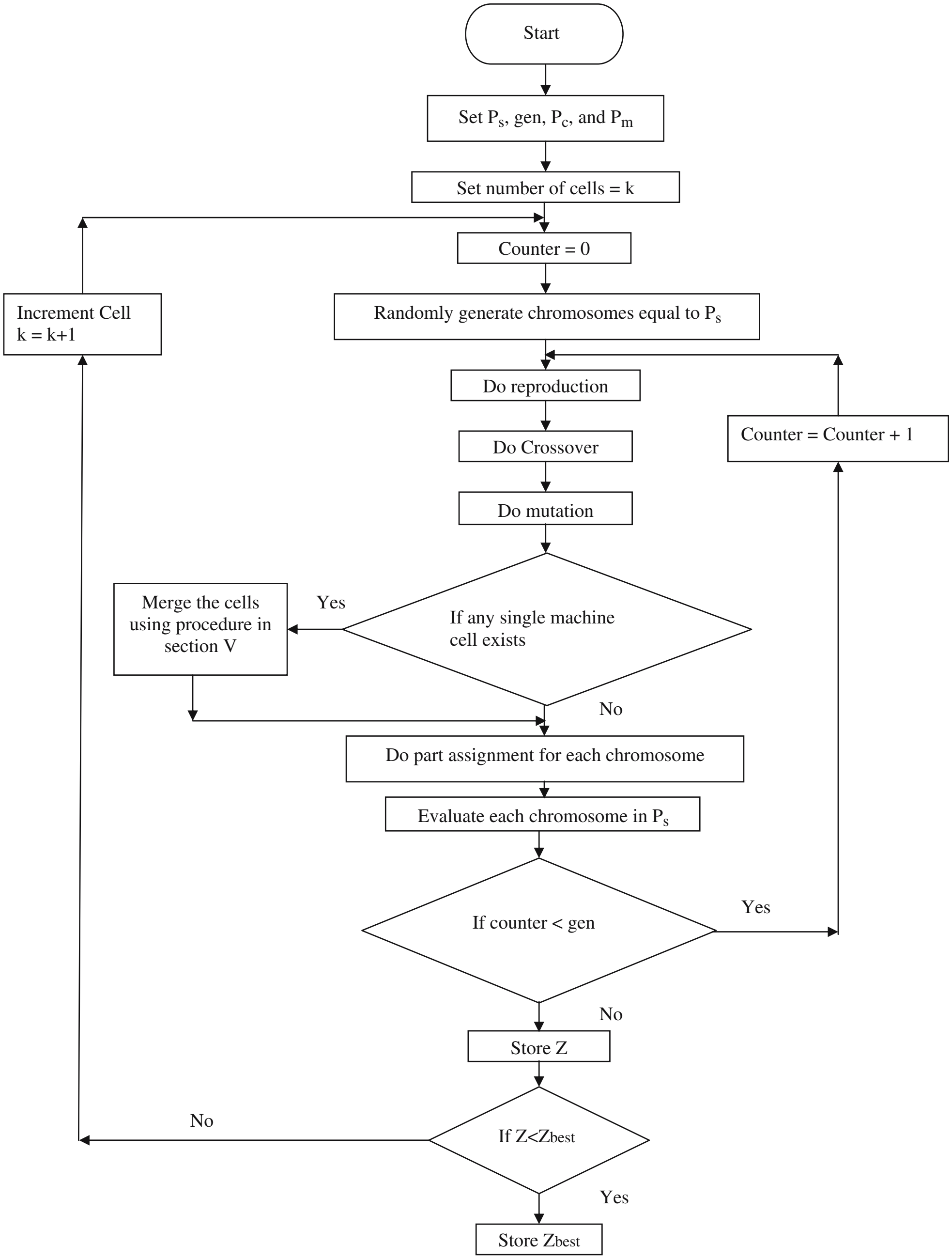
Table 2

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 model developed using GA has been tested with 25 benchmark problems of varied sizes ranging from 5×8 to 24×40 from open literature and the results are compared with K-means clustering and C-link clustering algorithms [3, 17], given in Table 1, confirm that GA is an appropriate solution methodology to such type of optimization problems. The flow chart for the proposed algorithm is shown in Fig. 4. The crossover and mutation probabilities are fixed to be 0.5 and 0.1, respectively. This probability can be varied depending upon the decision maker to tune the algorithm. The chromosome representation used in this study may

Table 2 CPU time for the proposed GA

S.N	Problem size	Population size (nos)	No. of generations	CPU time (sec)
1	10×15	20	103	0.21978
2	12×31	25	248	0.54945
3	16×43	20	494	1.26373
4	24×40	25	853	3.62637
5	30×41	15	593	2.19780

Fig. 4 Flow Chart for cell formation algorithm



result in the formation of an empty cell or violates some constraints. Particularly, crossover may result in the formation of a chromosome like 113331 when predefined number of cells is three. The above chromosome contains an empty cell where cell number 2 is missing. In such cases, the respective chromosomes are rejected. Crossover and mutation steps are repeated with other pairs of chromosomes till a useful chromosome is obtained. The number of cells greatly influences the objective function Z since increase in the number of cells decreases cell load variation and increases exceptional elements. Hence it rests with the decision maker to trade off between the objectives and choose the value of the weighting factors accordingly. In this work the weighting factor is assumed to be 0.5 to provide equal importance to both the objectives. However this can be varied depending on the decision maker. An example with varying weightages is given in Table 3, for problems 5 of size 9×9 (Table 1). The number of generations is varied from problem to problem in the range of 50 to 1000. Similarly, the population size is varied in the range of 10 to 40 depending on the size of problem. The convergence property of GA is given in Fig. 1.

K-means clustering algorithm is used to group the objects based on the attributes or features. K is the number

Table 3 Z value for different weightages for problem 5 (9×9)

S.N	q_1	q_2	Z
0	0.0	1.0	12.50
1	0.1	0.9	12.36
2	0.2	0.8	12.23
3	0.3	0.7	12.09
4	0.4	0.6	11.95
5[a]	0.5	0.5	11.82
6	0.6	0.4	11.68
7	0.7	0.3	11.54
8	0.8	0.2	11.41
9	0.9	0.1	11.27
10	1.0	0.0	11.11

[a] considered in Table 1

of groups to be obtained as a result. K-means clustering algorithm in which centroid coordinate is fixed based on that the distance of each object to the centroid is found and grouping is done. In cell formation problems K-means clustering gives better solutions based on Euclidean distance. In this research work, the solutions obtained by the proposed algorithm are compared with the solutions of K-means clustering algorithm. The complete linkage (C-link) clustering is also a hierarchical cluster analysis. The linkage function specifying the distance between two clusters is computed as the maximal object-to-object distance. In other words, the distance between two clusters is computed as the distance between the two farthest objects in the two clusters. Standard software, SYSTAT, is used for clustering in K-means and C-link algorithms. The number of iterations for K-means and C-link clustering algorithms varies from problem to problem. For each problem this number is varied till the minimum value of Z is obtained. Based on exhaustive experiments it is found that the maximum number of iterations required for large sized problem (24×40) in Table 1 is 30.

The proposed algorithm gives same results as that of K-means and C-link algorithms when the problem size is small, if the problem size increases the GA outperforms other methods. The time taken for the problems varies with the size of the problem, population size and number of generations, which is given in Table 2. A new measure, modified grouping efficiency (MGE), is proposed to assess the performance of the grouping with real values. This measure is capable of judging the goodness of block diagonal structure as it takes care of both voids and exceptional elements into consideration. The results of proposed algorithm are compared with the results obtained from K-means clustering algorithm and C-link algorithm based on MGE (Table 1). Problems have been tested by varying the number of cells from 2 to 5 depending on the total number of machines. The results obtained from GA are found to outperform both K-means and C-link algorithms. The algorithm is coded in C^{++} and run on Pentium IV PC, 2.4 GHz processor. The software can be obtained from the corresponding author on request.

8 Conclusion

The proposed GA is tested with different size problems from open literature and resulting solutions are compared with the solutions obtained from K-means clustering and C-link clustering algorithms. It is found that the proposed GA outperforms existing methods or remains the same as far as modified grouping efficiency is concerned. Since the objective function is a combination of exceptional elements and cell load variation, it depends upon the decision maker to make a trade off between two conflicting objectives by preferring weight for each objective. In some cases, particularly in problems 15 and 25, the value of objective function is more in the case of GA compared to other two algorithms, but a block diagonal structure is always better in the case of GA. If a different weighting factor had been chosen, the values of objective function would have been decreased. The work can be further extended in future incorporating production data like product sequence, machine capacity, lot size of parts, layout considerations and material handling systems enhancing it to a more generalized manufacturing environment.

Acknowledgments The authors would like to thank the organizing secretary of 22nd AIMTDR and the anonymous reviewers for their constructive suggestions that helped to improve the quality of this paper.

References

1. Ranson GM (1972) Group technology: a foundation for better total company operation. McGraw-Hill, London
2. Boctor FF (1991) A linear formulation of the machine-part cell formation problem. Int J Prod Res 29:343–346
3. Venugopal V, Narendran TT (1992) Cell formation in manufacturing systems through simulated annealing. Eur J Oper Res 63:409–422
4. Kao Y, Moon YB (1991) A unified group technology implementation using the back propagation learning rule of neural networks. Comps & Ind Engg 20(4):425–437
5. Jayakrishnan Nair G, Narendran TT (1999) ACCORD: a bicriterion algorithm for cell formation using ordinal and ratio-level data. Int J Prod Res 37(3):539–556
6. Goncalves, JF, Resende, MGC (2002) 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
7. Muruganandam A, Prabhaharan G, Asokan·P, Baskaran V (2005) A memetic algorithm approach to the cell formation problem. Int J Adv Manuf Technol 25:988–997

8. Venugopal V, Narendran TT (1992) A genetic algorithm approach to the machine component grouping problem with multiple objectives. Comps & Ind Engg 22:469–480
9. George AP, Rajendran C, Ghosh S (2003) An analytical-iterative clustering algorithm for cell formation in cellular manufacturing systems with ordinal-level and ratio-level data. Int J Adv Mfg Tech 22:125–133
10. Yasuda K, Hu L, Yin Y (2005) A grouping genetic algorithm for the multi-objective cell formation problem. Int J Prod Res 43 (4):829–853
11. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, Michigan
12. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA, pp 28–150
13. Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin Heidelberg New York, pp 115–263
14. Zolfaghari S, Liang M (2003) A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. Comps & Ind Engg 45:713–731
15. Chandrasekharan MP, Rajagopalan R (1986a) An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. Int J Prod Res 24:451–464
16. Kumar KR, Chandrasekaran MP (1990) Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. Int J Prod Res 28(2):233–243
17. Prabhakaran G, Janakiraman TN, Sachithanandam M (2002) Manufacturing data based combined dissimilarity coefficient for machine cell formation. Int J Adv Manuf Technol 19:889–897