



MANUFACTURING CELL DESIGN WITH REDUCTION IN SETUP TIME THROUGH GENETIC ALGORITHM

M. MURUGAN ¹, V. SELLADURAI ².

1. SENIOR LECTURER, DEPARTMENT OF MECHANICAL ENGINEERING, SRI RAMAKRISHNA ENGINEERING COLLEGE, COIMBATORE, TAMILNADU, INDIA
2. PROFESSOR, DEPARTMENT OF MECHANICAL ENGINEERING, COIMBATORE INSTITUTE OF TECHNOLOGY, COIMBATORE, TAMILNADU, INDIA

ABSTRACT

Cellular manufacturing emerged as a production strategy capable of solving the problems of complexity and long manufacturing lead times in batch production. The fundamental problem in cellular manufacturing is the formation of product families and machine cells. This paper presents a new approach for obtaining simultaneous arrangement of part families and machine cells for cellular manufacturing systems. The main feature of the proposed method is, the relevant production data such as process sequences and setup times are taken in to account. It has the ability to select the best solution among the solutions of compactness, group technology efficiency and reducing setup time efficiency for each part before attempting to cluster the machines and parts. The formation of part family and machine cell has been treated as a maximization problem according to a defined performance measure ' β '. A genetic algorithm has been developed for solving the cell formation problem considering the reduction in setup time. The validation has been done based on a real time manufacturing data. This algorithm is written in the 'C' language on Intel Pentium / PIII compatible system.

Keywords: *Part family, Machine cell, Setup time, cell formation, genetic algorithm and Performance measure.*

M.Murugan, Faculty of Mechanical Engineering, Sri Ramakrishna Engineering College, Coimbatore- 641 022, Tamilnadu, India. Phone: 91-422-2460088,extn. 364, Fax: 91-422-2461089, **E-mail:** murugan_srec@yahoo.co.in



1. INTRODUCTION

Recent competitive economic situations demand quicker supply of newer products with more innovative functionality to satisfy quickly changing customer requirements. In striving to remain competitive, the concept of Cellular Manufacturing has been extensively employed to the manufacturing systems. Cellular manufacturing emerged as a production strategy capable of solving the problem of complexity and long manufacturing lead times in batch production systems in the beginning of the 1960s. Burbidge (1979) defined group technology (GT) as an approach to the optimization of work in which the organizational production units are relatively independent groups, each responsible for the production of a given family of products. The fundamental problem in cellular manufacturing is the formation of product families and machine cells.

Group technology is a principle, which decomposes a global system into several subsystems, which are easier to manage than the entire system. Applied to manufacturing, this principle is the base for the design of production cells. According to Wemmerlov and Hyer, the main improvements that can be expected from cellular manufacturing are reductions in throughput time, in material handling, in setup time and improvement of part quality.

In essence, the basic information required to solve a CM problem is the Machine-Part Incidence Matrix, which consists of values of 0s and 1s, where 1 in an entry denotes that the corresponding coordinate of a part that requires the service of that machine, or otherwise. All CM problems are resolved by manipulating the incidence matrix in a manner such that the grouping of all similar objects is possible. The manipulation of a machine-part incidence matrix is based on, (a) the direct approach and (b) the indirect approach. The direct approach to a CM problem includes those methods in which the grouping of similarity objects entails rearrangement of rows and columns of the original incidence matrix. In such an approach, machine cells and part families are formed simultaneously. The Rank Order Clustering (ROC) algorithm of King (1980) and the Cluster Identification Algorithm (CIA) of Kusiak and Chow (1987) are

typical examples of such an approach. While the indirect approach involves the transformation of the original incidence matrix into a different form of information before data analysis is carried out. Data transformation can be done in two ways. The first way is by transforming a Machine Part Incidence Matrix (MPIM) into a part-based matrix in which the final result is in the form of part families. The second way is by transforming the original Machine Part Incidence Matrix into a machine-based matrix, the result of which is based on machine cells. Indirect approaches have been studied by Chow and Hawaleshka (1992), King and Nakomachai (1982), Wei and Kern (1989) and so on. Most of these methods of cell formation are based on machine-part incidence matrix alone. Other factors such as operations sequence, reduction in setup time and production volumes, if incorporated, can enhance the quality of the solutions. Nair and Narendran (1998) presented the algorithm, which clusters machines and parts on the basis of sequence data. Hiroshi Ohta & Masateru Nakamura (2002) developed the algorithm for cell formation with reduction in set up times through iterative method.

To the best of our knowledge, no algorithm to optimally solve the product- families and machine-cells problem has yet been proposed in the literature. This paper proposes a new cell formation with reduction in setup times between machines in the same cell through genetic algorithm approach. The objective of this paper is to present a procedure for obtaining manufacturing cells considering the factor sequence data with reduction in setup time. The approach combines a heuristic with a genetic algorithm. The heuristic proposed by Hiroshi Ohta et al. is responsible for selecting the initial solutions and the genetic algorithm is responsible for generating sets of machines cells with the objective of constructing sets of machine/product groups and improving the performance measure.

2. LITERATURE SURVEY

The fundamental problem in cellular manufacturing is the formation of product families



and machine cells. The objective of this product-machine grouping problem is to form perfect (i.e. disjoint) groups in which products do not have to move from one cell to the other for processing. The most common algorithms for GT found in the literature can be classified into the following four method categories: array-based, clustering, mathematical programming-based, and graph theoretic.

Array-based clustering methods perform a series of column and row permutations to form product and machine cells simultaneously. King (1980) and later King and Nakornchai (1982) developed the earliest array-based methods. King and Nakornchai (1982), Chandrasekharan and Rajagopalan (1987), Khator and Irani (1987), and Kusiak and Chow (1987) proposed other algorithms.

A comprehensive comparison of three array-based clustering techniques is given in Chu and Tsai (1990). The quality of the solution given by these methods depends on the initial configuration of the zero-one matrix. McAuley (1972) and Carrie (1973) developed the first algorithms using clustering and similarity coefficients. Since then, Mosier and Taube (1985a, b), Seifoddini (1989), Gupta and Seifoddini (1990), Khan et al. (2000), Yamada and Yin (2001), and Dimopoulos and Mort (2001) proposed hierarchical methods. These methods have the disadvantage of not forming product and machine cells simultaneously; so additional methods must be employed to complete the design of the system.

GRAFICS, developed by Srinivasan and Narendran (1991), and ZODIAC, which is a modular version of MacQueen's clustering method, developed by Chandrasekharan and Rajagopalan (1987), are examples of non-hierarchical methods. Miltenburg and Zhang (1991) present a comprehensive comparison of nine clustering methods where non-hierarchical methods outperform both array-based and hierarchical methods.

Mathematical programming methods treat the clustering problem as a mathematical

programming optimization problem. Different objective models have been used. Kusiak (1987) suggested the p -median model for GT, where it minimizes the total sum of distances between each product/machine pair. Shtub (1989) modeled the grouping problem as a generalized assignment problem. Choobineh (1988) formulated an integer-programming problem, which first determines product families and then assigns product families to cells with an objective of minimizing costs. Co and Araar (1988) developed a three-stage procedure to form cells and solved a 3-assignment problem to assign jobs to machines. Gunasingh and Lashkari (1989) formulated an integer-programming problem to group machines and products for cellular manufacturing systems.

Srinivasan et al. (1990) modeled the problem as an assignment problem to obtain product and machine cells. Joines et al. (1996) developed an integer program that is solved using a genetic algorithm. Cheng et al. (1998) formulate the problem as a traveling salesman problem and solve the model using a genetic algorithm. Chen and Heragu (1999) present two stepwise decomposition approaches to solve large-scale industrial problems. Won (2000) presents a two-phase methodology based on an efficient p -median approach. Akturk and Turkcan (2000) propose an integrated algorithm that solves the machine/product-grouping problem by simultaneously considering the within-cell layout problem.

Plaquin and Pierreval (2000) propose an evolutionary algorithm for cell formation taking into account specific constraints. Zhao and Wu (2000) present a genetic algorithm for cell formation with multiple routes and objectives. Caux et al. (2000) address the cell formation problem with multiple process plans and capacity constraints using a simulated annealing approach. Onwubolu and Mutingi (2001) develop a genetic algorithm approach taking into account cell-load variation. Uddin and Shanker (2002) address a generalized grouping problem, where each part has more than one process route. José Fernando Gonçalves and Mauricio G.C. Resende (2004) propose an evolutionary algorithm for obtaining the manufacturing cells. The problem is formulated as an integer-programming problem and a procedure based on a genetic algorithm is suggested as a solution methodology.



Rajagopalan and Batra (1975) were the first to use graph theory to solve the grouping problem. They developed a machine graph with as many vertices as the number of machines. Two vertices were connected by an edge if there were parts requiring processing on both the machines. Cliques obtained from the graph were used to determine machine cells. The limitation of this method is that machine cells and part families are not formed simultaneously. Kumar et al. (1986) solved a graph decomposition problem to determine machine cells and part families for a fixed number of groups and with bounds on cell size. Their algorithm for grouping in a flexible manufacturing system is also applicable in the context of Group Technology.

Vannelli and Kumar (1986) developed graph theoretic models to determine machines to be duplicated so that a perfect block diagonal structure can be obtained. Kumar and Vannelli (1987) developed a similar procedure for determining parts to be subcontracted in order to obtain a perfect block diagonal structure. These methods are found to depend on the initial pivot element choice. Vohra et al. (1990) suggested a network-based approach to solve the grouping problem. They used a modified form of the Gomory-Hu algorithm to decompose the part-machine graph.

Askin et al. (1991) proposed a Hamiltonian-path algorithm for the grouping problem. The algorithm heuristically solves the distance matrix for machines as a TSP and finds a Hamiltonian path that gives the rearranged rows in the block diagonal structure. The disadvantage of this approach is that actual machine groups are not evident from its solution. Lee and Garcia-Diaz (1993) transformed the cell formation problem into a network flow formulation and used a primal-dual algorithm developed by Bertsekas and Tseng (1988) to determine the machine cells. Other graph approaches include the heuristic graph partitioning approach of Askin and Chiu (1990) and the

minimum spanning tree approach of Ng.S., (1993, 1996). Selim et al. (1998) provide a comprehensive mathematical formulation of the cell formation problem and present a methodology-based classification of prior research. .

Srinivasan and Narendran (1991) developed a non-hierarchical clustering algorithm that identified seeds for clustering by solving an assignment problem. The above algorithms are based on the binary data. Based on production data, Gupta and Seifoddini (1990) and Nair and Narendran (1998) developed a similarity coefficient based on part sequence data and developed non-hierarchical clustering algorithm to allow natural clusters to emerge and yield solutions of higher quality. They also developed (1999) a similarity coefficient based on production sequence, volumes, processing times and machine capacities and developed a non-hierarchical clustering algorithm with a twin objectives of minimizing within cell load variation as well as intercellular moves. Hiroshi Ohta and Nakamura (2002) developed an algorithm, which clusters machines and parts on the basis of sequence data and reduction in setup times between machines in the same cell, with an objective of maximizing the performance measure through an iterative method.

This paper proposes a new cell formation with reduction in setup times between machines in the same cell through genetic algorithm approach. The objective of this paper is to present a procedure for obtaining manufacturing cells considering the factors sequence data and reduction in setup time. The approach combines a heuristic with genetic algorithm. The heuristic proposed by Hiroshi Ohta et al. is responsible for selecting the initial solutions and the genetic algorithm is responsible for generating sets of machines cells with the objective of constructing sets of machine/product groups and improving the performance measure.

Notations

n	number of parts
m	number of machines
b_{ji}	the operations sequence of part j on machine i
r_j	maximum number of operations for part j
p_{jil}	setup time for part j from machine i to machine l
d_{jil}	reducing rate of setup time p_{jil} if machines i and l are in the same cell
S_{il}	pairwise similarity between machines i and l
a, b	weighting factors for similarity (where $a + b = 1, 0 \leq a, b \leq 1$)
IC_{il}	pairwise similarity between machines i and l on the movement of parts
ST_{il}	pairwise similarity between machines i and l on setup times
x_{jil}	number of import-trips to machine i and export-trips from machine i for part j that needs machine i and l
y_{ji}	number of trips on machine i for part j
λ, μ, θ	weighting factors for performance measure (where $\lambda + \mu + \theta = 1, 0 \leq \lambda, \mu, \theta \leq 1$)
CPN	compactness
GTE	group technology efficiency
STE	reducing efficiency of setup times
c	number of machine-cells
TO_k	total number of operations in the k -th cell
NO_k	total number of non-operations (voids) in the k -th cell
H	the maximum number of inter-cell travels possible
G	number of inter-cell travels in the obtained cell
U	total reduced times in the obtained cell
T	maximum reduced setup times possible
z_{jk}	0, if the k -th and $(k + 1)$ -th operations for part j are done in the same cell, and 1, otherwise
v_{jkil}	1, if the k -th and $(k + 1)$ -th operations for part j are done on machines i and l in the same cell, respectively, and 0, otherwise
SA_{ei}	average similarity between machine which belongs machine-cluster e and machine i which has the same maximum similarity to centroids or singleton i
ϵ_e	number of machines which belong to machine-cluster e
W_e	universal set of machine number which belongs to machine-cluster e
PC_{qj}	travel-count of part j to the q -th cell
CS_{ij}	1, if part j visits machine i , and 0, otherwise
MC_{iq}	1, if machine i is in the q -th cell, and 0, otherwise

3. Similarity measure

Considering the number of trips for the parts and the set up time between machines i

and l , and the similarity measure introduced by Hiroshi Ohta & Nakamura is well taken in to this analysis.

$$S_{il} = aIC_{il} + bST_{il} \quad (1)$$



Where

Sil: Pairwise similarity between machines i and l

ICil: Pairwise similarity between machines i and l on the movement of part

STil: Pairwise similarity between machines i and l on setup times

a, b : Weighting factors for similarity (where a + b = 1, 0 <= a, b <= 1)

$$IC_{il} = \frac{\sum_{j=1}^n (x_{jil} + x_{jli})}{\sum_{j=1}^n (y_{ji} + y_{jl})} \tag{2}$$

$$y_{ji} = \begin{cases} 0 & \text{if } b_{ji} = 0 \\ 1 & \text{if } b_{ji} = 1 \text{ or } r_j, \\ 2 & \text{otherwise,} \end{cases} \quad x_{jil} = \begin{cases} 0 & \text{if } b_{ji} = 0 \text{ or } b_{jl} = 0 \\ 1 & \text{if } b_{ji} = 1 \text{ or } r_j \text{ and } b_{jl} \neq 0 \\ 2 & \text{otherwise and } b_{jl} \neq 0, \end{cases}$$

Where,

Xjil: Number of import-trips to machine i and export-trips from machine i for part j that needs machine i and l

Yji: Number of trips on machine i for part j

bji: The operations sequence of part j on machine I ,bji=0 means that part j does not need machine i

$$ST_{il} = \sum_{j=1}^n (d_{jil}P_{jil} + d_{jli}P_{jli}) / \sum_{j=1}^n \left(\sum_{f=1}^m d_{jfi}P_{jfi} + \sum_{h=1}^m d_{jhl}P_{jhl} \right) \quad (i \neq l, i \neq f, l \neq h) \tag{3}$$

djil : reducing rate of setup time Pjil if machine i and j are in the same cell

Pjil : setup time for part j from machine i to machine l

3.1 Performance measure

The performance of the obtained cells is evaluated by the Compactness (members of the same group are highly similar), the group

technology efficiency, and the reducing efficiency of the setup times. The performance measure introduced by Hiroshi Ohta & Nakamura is



$$\beta = \lambda CPN + \mu GTE + \theta STE \quad \text{_____} \quad (4)$$

β : Performance measure

CPN: Compactness \diamond members of the same group are highly similar

GTE: Group technology efficiency

STE: Efficiency of setup times

λ, μ, θ : weighting factors for performance measure

Where,

$$CPN = \sum_{k=1}^c TO_k / \sum_{k=1}^c (TO_k + NO_k), \quad \text{_____} \quad (5)$$

TO_k : total number of operations in the k-th cell

NO_k : total number of non-operations in the k-th cell

C : number of machine cell

$$GTE = (H - G)/H \quad \text{_____} \quad (6)$$

$$H = \sum_{j=1}^n (r_j - 1),$$

$$G = \sum_{j=1}^n \sum_{k=1}^{(r_j - 1)} z_{jk}$$

Where,

H: the maximum number of inter-cell travels possible

G: Number of inter-cell travels in the obtained cell

Z_{jk}: 0, if the k-th and (k+1)-th operations for part j are done in the same cell, and 1, otherwise

$$STE = U/T \quad \text{_____} \quad (7)$$



$$U = \sum_{j=1}^n \sum_{k=1}^{(r_j-1)} \sum_{i=1}^m \sum_{l=1}^m d_{jil} v_{jkil} p_{jil} \quad (i \neq l),$$

$$T = \sum_{j=1}^n \sum_{i=1}^m \sum_{l=1}^m d_{jil} p_{jil} \quad (i \neq l).$$

Where,

U : Total reduced times in the obtained cell

T : Maximum reduced setup times possible

d_{jil} : reducing rate of setup time p_{jil} if machine i and j are in the same cell

v_{jkil} : 1, if the k -th and $(k+1)$ -th operations for part j are done on machines j and i in the same cell, respectively, and 0, otherwise

p_{jil} : setup time for part j from machine i to machine l

Eqn. (2) gives the similarity coefficient as the ratio of the sum of the moves common to the machines i and l and sum of the total number of moves to and from machines i and l . Eqn. (3) gives the similarity coefficient as the ratio of the sum of the reduced setup times when machines i and l are in the same cell and sum of total reduced setup times possible between machines i and l . Eqn.(4) gives the performance measure β Eqn. (5) gives the compactness that is defined as the ratio of the number of operations within it to the maximum number of operations possible in it. Eqn. (6) gives the Group Technology Efficiency (GTE) which is defined as the ratio of the difference between the maximum number of intercell travels possible and the number of intercell travels in obtained cells. Eqn. (7) gives the reducing efficiency of the setup times which is defined as the ratio of the sum of reduced setup times in the obtained cells to the sum of the reduced setup times possible.

4. Cell Formation

$$SA_{ei} = \frac{\sum_{l \in W_e} S_{il}}{\epsilon_e}.$$

Based on the similarity measure given by Eqn. (1), the machine-pair is chosen for which the similarity measure is minimum as the seeds for clustering. The machines chosen as the seeds for clustering are called *centroids*. Once a set of centroids is determined, machine-cells can be formed around these seeds by allotting other machines to the seed with which it has the maximum similarity. If a machine has the maximum similarity to the centroids, Nair and Narendran (1998) suggest that ties are broken arbitrarily (by randomly choosing a subset of machines from the contenders). However, in such a case, Hiroshi Ohta & Nakamura proposes the following modified procedure.

- The allotting machine is deferred and other machines are assigned to the centroids.
- The average similarity between the deferred machine and each cluster is computed by using Eqn. (8).

(8)



Where,

SA_{ei}: average similarity between machine which belongs machine-cluster e and machine i which has the same maximum similarity to centroids i

We: universal set of machine number which belongs to machine-cluster e

E_e: number of machines which belong to machine-cluster e

- The deferred machine is allotted to the cluster with the maximum average similarity. If any centroid fails to attract at least one other machine to its cluster (such centroids are called singletons), the following procedure is adopted.
- The average similarity between the singleton and each cluster is computed by using Eqn. (8).
- The singleton is assigned to the cluster with which it has the maximum of above values.

$$PC_{qj} = \sum_{i=1}^m CS_{ij}MC_{iq},$$

where $q = 1, \dots, c, j = 1, \dots, n.$

----- (9)

Where,

PC_{qj}: travel-account of part j to the q-th cell

CS_{ij}: 1, if part j visits machine i, and 0, otherwise

MC_{iq}: 1, if machine i is in the q-th cell, and 0, otherwise

4.2 Performance evaluation of cell formation

The performance of the cells obtained based on the centroids which have the same similarity is evaluated by using Eq (4). The cell, which has the maximum performance measure, is chosen as the best solution. Machine cells are formed based on the following algorithm presented by Hiroshi Ohta & Nakamura. The procedure is explained with an illustrative example shown in table -1 and table-2.

Algorithm

- Algorithm Step 0: read the data
- Step 1: set TAL (threshold affinity level)=0,

4.1 Part-families formation

Each machine-cell, thus obtained, provides a seed to cluster parts. The procedure for clustering parts is as follows:

- The travel-count of each part to each obtained cell is computed by using Eqn. 9
- The part is assigned to the cell with which it has the maximum travel-count. Break ties in favour of the cell, which has the smallest number of ‘1’s

- Step 2: compute the similarity measures and put the min similarity into TAL
- Step 3: Identify the centroids
- Step 4: Cluster machines to form cells
- Step 5: Assign parts to machine-cells to form part-families
- Step 6: compute performance measure β
- Step 7: if the β is higher than existing β , exchange the β
- Step 8: if all machine are chosen, go to step 10
- Step 9: increase TAL, go to Step 5
- Step 10: print the best solution
- Step 11: Stop.

4.2.1 Illustrative Example



Consider the data of 8 machines and 10 parts problem shown in table-1 showing the initial part incidence matrix, which gives the operation sequence of parts on the machines. Table-2 shows the setup times of each part from operation kth to operation (k+1) th. Assume that all reducing rates of setup time $D_{ji=0.1}$ i.e. the set up times for part j

from machine i to machine l can be reduced 10% if machines i and l are in the same cell. By using Eqn. (1), the pair wise similarity measures were obtained for the data of table 1 & 2 and centroids that have the smallest similarity $S_{il=0.00}$, are shown in table-3 and the results obtained are shown in table-4.

Problem Data

Part	Machine							
	1	2	3	4	5	6	7	8
1	4				1	2		3
2			1					
3	3					2		1
4		1		2				
5		3		1	2			
6			2				1	3
7	3		4			1		2
8			2				1	
9	2				3	4		1
10		1		3	2			

Table-1 - 8 machines –10 parts incidence matrix

Setup Time Data

Part	Operation k → k + 1		
	1 → 2	2 → 3	3 → 4
1	5	2	4
2	4		
3	2	3	
4	5		
5	1	3	
6	1	5	
7	4	3	3
8	1		
9	2	5	2
10	2	2	

Table-2 - Setup times in the 8 machines x 10 parts problem



Pair-wise similarity between machines i and l

Machine l	Machine i						
	1	2	3	4	5	6	7
2	0.000						
3	0.238	0.000					
4	0.000	0.822	0.000				
5	0.356	0.439	0.000	0.373			
6	0.572	0.000	0.100	0.000	0.437		
7	0.000	0.000	0.617	0.000	0.000	0.000	
8	0.657	0.000	0.429	0.000	0.215	0.662	0.111

Table-4 - Pair-wise similarity between machines i and l

Machine cells and Part families by Hiroshi Ohta algorithm

Machine cells	Part family	CPN	STE	GTE	EFFICIENCY
(13678,245)	(136789,24510)	0.667	0.796	0.850	0.744986
(124568,37)	(123457910,68)	0.538	0.864	0.900	0.710332
(23457,168)	(2456810,1379)	0.619	0.661	0.750	0.662278
(2345,1678)	(24510,136789)	0.625	0.627	0.650	0.631780
(12456,378)	(123457910,68)	0.543	0.661	0.650	0.599493
(134678,25)	(136789,24510)	0.545	0.593	0.650	0.583532
(123678,45)	(136789,24510)	0.545	0.559	0.650	0.575058
(12368,457)	(13679,245810)	0.550	0.525	0.550	0.543856

Table-4 – Result of Hiroshi Ohta algorithm



The results of machine cells and part families based on the 1 set of centroids and the performance measure β obtained by using Eqn. (4). as shown in table 4, in this example 4 types of cell formed in the first iteration, but type 1 has the

performance measure $\beta=74.5\%$. Therefore, centroid (1, 2) was chosen as the best centroids in the first iteration, because the other sets of centroids in type 1 lead the same cell formation and the result is shown in table-5.

Final Result of the 8 machines x 10 parts problem

Part	machine							
	2	4	5	3	7	6	1	8
2	2	1						
4	1	2						
5	3	1	2					
10	1	3	2					
6				2	1			3
8				2	1			
1			1			2	4	3
3						2	3	1
7				4		1	3	2
9			3			4	2	1

Table-5- Final result of Hiroshi Ohta Algorithm.

5. GENETIC ALGORITHM APPROACH

Genetic algorithms (GA) were introduced by Holland (1975) and have been applied in a number of fields, e.g. mathematics, engineering, biology, and social science (Goldberg, 1989). GAs are search algorithms based on the mechanics of natural selection and natural genetics. They combine the concept of survival of the fittest with structured, yet randomized, information exchange to form robust search algorithms.

The concept of genetic algorithms is based on the evolution process that occurs in natural biology. An initial population of possible solutions (referred to as *individuals* or *chromosomes*) is generated. Some individuals are selected to be parents to produce offspring via a *crossover operator*. All the individuals are then evaluated and selected based on Darwin's concept

of survival of the fittest. The process of reproduction, evaluation, and selection is repeated until a termination criterion is reached. In addition, a *mutation operator* with certain probability is applied to the individuals to change their genetic makeup. The objective of this mutation process is to increase the diversity of the population and ensure an extensive search. Each *iteration* (also referred to as *generation* or *family of solutions*) is made up of chromosomes. Each chromosome is in turn made up of individual *genes*. These genes are encodings of the design variables that are used to evaluate the function being optimized. In each iteration of the search process, the system has a fixed population of chromosomes that represent the current solutions to the problem. Figure - 1 represents a pseudo-code for a standard genetic algorithm.



Genetic algorithm

```

{
    Generate initial population  $P_t$ 
    Evaluate population  $P_t$ 
    while stopping criteria not satisfied
repeat
    {
        Select elements from  $P_t$  to put
into  $P_{t+1}$ 
        Crossover elements of  $P_t$  and put
into  $P_{t+1}$ 
        Mutate elements of  $P_t$  and put
into  $P_{t+1}$ 
        Evaluate new population  $P_{t+1}$ 
         $P_t = P_{t+1}$ 
    }
}

```

Figure -1 A standard genetic algorithm.

The GA calls a subroutine to compute the fitness value (the quality) for each chromosome in the population. This fitness value is the only feedback to the GA. As mentioned earlier, the fitness function used is performance measure. The other important aspects of genetic algorithms: chromosomal representation and decoding, parent selection, crossover, and mutation will be discussed next.

5.1 Proposed Genetic Algorithm

The GA approach for machine grouping problem can be summarized as follows:

Step 1. Initialization

- Initialize population P_0 randomly or select from the known values
- Compute fitness function f_i for all chromosomes in the population P_0 .

Step 2. Parent selection

- Randomly select two chromosomes C_k and $C_{k'}$ from the current population P_t .

Step 3. Crossover (one-point)

- Generate an integer random cutting point x from the range $[1, 8]$.
- Obtain two offspring chromosomes by exchanging the genes of chromosomes C_k and $C_{k'}$.

Step 4. End of generation

- If the size of offspring is less than L , go to step 2; otherwise, go to the next step.

Step 5. Mutation

- Generate an integer random number y from the range $[1, 8]$.
- Select y genes randomly and change their values to random integer numbers from the range $[1, 8]$.

Step 6. Fitness evaluation

- Compute fitness function f_i for all offspring chromosomes.
- For the next generation population P_{t+1} , select the best chromosomes among the offspring and the current population.

Step 7. Termination

- If the termination condition is not satisfied, set $t \rightarrow t+1$ and go to step 2; otherwise go to next step

Step 8. Stop

- Calculate the performance measure.

In the above algorithm, the user can set the termination condition. Either a limited number of generations or a maximum run time could be used as the termination rule. The GA algorithm has been coded in “C++” language.

5.1.1 Representation

In this work, number of cells considered is only two and coding is done using alphabets **a** & **b**, and every gene represents a cell number, and the



position of a gene in the chromosome represents a machine number. The length of a chromosome represents the number of machines considered in the problem. For example

abab

baaa

- Length of the chromosome is 8, which indicates the number of machines in the problem is 8.
- The genes in the chromosome are **a** and **b**. The alphabet **a** indicates the position of the machine number and it belongs to the cell number 1. The alphabet **b** indicates the position of the machine number and it belongs to the cell number 2.
- The position of an alphabet indicates which machine is present in which cell, i. e. machine 2, 4 and 5 is in cell 2 and machine 1, 3,6,7,8 is in cell 1.

Initial Population

Sample Number	Initial Population
1	ababbaaa
2	aabaaaba
3	baaaabab
4	baaaabbb
5	aabaaabb
6	abaabaaa
7	aaabbaaa
8	aaabbaba

Table-6- Initial Population

5.1.3 Fitness Function

The fitness function for each string in the population is computed and the objective is to find the maximum fitness function value. Initial population and its objective function values are shown in table-7.

$$F(t) = \frac{1}{(1 + f(t))}$$

where,

f(t) = Objective function value of a string ie objective function value. Eqn. (1) is taken as f(t)

F(t) = Fitness function

Initial Population and their objective function value



Sample Number	Initial Population	Objective Function Value
1	ababbaaa	0.744986
2	aabaaaba	0.710332
3	baaaabab	0.662278
4	baaaabbb	0.631780
5	aabaaabb	0.599493
6	abaabaaa	0.583532
7	aaabbaaa	0.575058
8	aaabbaba	0.543856

Table-7-Initial population and their Objective function values

5.1.4 Selection and Reproduction

Reproduction selects good strings in a population and forms a mating pool. That is why the reproduction operator is sometimes called the selection operator. Reproduction is responsible for the exploration of the current population by making many duplicates of good strings. Cross over and mutations are responsible for exploring a set of good strings for better strings. The success of GA depends on a balance between the two. If too many copies of the good strings are allocated in the mating pool, then the diversity of the mating pool reduces, which in turn reduces the extent of the search that can be accomplished using crossover and mutation operators.

Here the Universal sampling method is adopted for selecting the good strings and the population after reproduction is shown in table-8. The probability of selecting each string is calculated by

$$SI = \frac{F(t)}{F_{total}}$$

Where,

F (t) = Fitness value of the ith string

F_{total} = Total fitness value of all strings

The Total fitness of the population is calculated as

$$F_{total} = \sum_{i=1}^n F(t)$$



After Reproduction

Sample no	Population before reprod'n	Objective function value	Raw fitness value	Probability of selection	Cumulative probability	Random No.	Re prod'n Sample no.	Population after reprod'n
1	ababbaaa	74.498596	0.013423	0.104849	0.104849	0.14830	1	ababbaaa
2	aabaaaba	71.033203	0.014078	0.109964	0.214813	0.37190	2	aabaaaba
3	baaaabab	66.227798	0.015099	0.117943	0.332755	0.144100	1	ababbaaa
4	baaaabbb	63.178001	0.015828	0.123636	0.456391	0.936600	7	aaabbaaa
5	aabaaabb	59.949299	0.016681	0.130295	0.586686	0.239500	1	ababbaaa
6	abaabaaa	58.353199	0.017137	0.133859	0.720545	0.591500	4	aaabbaaa
7	aaabbaaa	57.505798	0.017390	0.135831	0.856376	0.934400	7	aaabbaaa
8	aaabbaba	54.385601	0.018387	0.143624	1.000000	0.661400	5	ababbaaa

Table-8- Population after Reproduction

5.1.5 Crossover

The chromosome to be crossed and the crossing points are to be selected randomly. Crossover is carried out with a probability called the crossover probability C_{prob} . Random numbers are generated for each chromosome and compared with the crossover probability values. The For example,

chromosomes within the values of C_{prob} are chosen for crossover. Here, Crossover probability is taken as 0.90. There are several types of crossover operators available and some of them are single point, two points, uniform and arithmetic crossover operators. Single point crossover is adopted in this work. The new population after mutation is shown in table-9.



(Parent 1)	a bbabaaa
(Parent 2)	a abaaaba

(Child 1)	aabaaaba
(Child 2)	ababbaaa

After Crossover

S.No.	Population after reproduction	Objective function value	Random numbers	Crossover Y or N	Cross over Site	Population after crossover
1	ababbaaa	80.497177	0.939700	N	-	ababbaaa
2	aabaaaba	83.595337	0.869000	Y	6	aabaaaaa
3	ababbaaa	80.497177	0.291400	Y	6	ababbaba
4	aaabbaaa	61.420650	0.237300	Y	3	aaabbaaa
5	ababbaaa	80.497177	0.722000	Y	3	ababbaaa
6	aaabbaaa	61.420650	0.524900	Y	3	aaabbaaa
7	aaabbaaa	61.420650	0.501800	Y	3	aaabbaaa
8	ababbaaa	80.497177	0.515100	N	-	ababbaaa

Table-9- Population after Crossover

5.1.6 Mutation

The crossover may cause the occurrence of an empty cell. To overcome this problem, mutation is carried out with mutation probability as 0.20. Random integers X and Y are generated for each solution string. The strings with random number within the mutation probability are chosen for mutation. Mutation operation is carried out within the solution string. Here, the shift mutation is used for each selected string, two mutation sites

are randomly selected and the genes corresponding to the mutation sites are interchanged. The same procedure is repeated for the rest of the selected solution strings. The new population after mutation is shown in table-10.

For example,

Take random numbers X=2, Y=5 then

Before mutation aabbbaab

After mutation abbbbaab

After Mutation



S.No.	Population before mutation	Random numbers	Mutate Y OR N	Mutation Site	Population after mutation
1	ababbaaa	0.128600	Y	4 6	abaabbaa
2	aabaaaaa	0.865400	N	0 0	aabaaaaa
3	ababbaba	0.176600	Y	5 6	abababba
4	aaabbaaa	0.315000	Y	4 2	abaabaaa
5	ababbaaa	0.680100	Y	6 5	abababaa
6	aaabbaaa	0.384100	Y	3 5	aabaaaaa
7	aaabbaaa	0.443800	Y	6 4	aaaabbaa
8	ababbaaa	0.188900	Y	5 2	ababbaaa

Table-10 - Population after Mutation

5.4 Termination of GA

Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found. The

application of conventional termination criteria becomes problematic. A common practice is to terminate the GA after a prespecified number of generations and then test the quality of the best members of the population against the problem. The solutions obtained after 100 generations are shown in table-11.

Machine cells	Part family	CPN	STE	GTE	Efficiency
(135678,24)	(136789,24510)	0.636	0.864	0.900	85.583977
(124568,37)	(12345910,678)	0.437	0.864	0.900	83.595337
(13678,245)	(136789,24510)	0.666	0.796	0.850	80.497177

Table-11- Results of Genetic Algorithm Approach

6.Results And Discussions

The optimum solution for the problem under study has been obtained using Genetic

Algorithm approach with crossover probability as 0.9 and mutation probability as 0.2. Table-12 shows the result of the objective function values and the relevant part families / machine cells.

Result of genetic algorithm approach



Machine cells	Part family	CPN	STE	GTE	Performance measure (β %)
(135678,24)	(136789,24510)	0.636	0.864	0.900	85.583977
(124568,37)	(12345910,678)	0.437	0.864	0.900	83.595337
(13678,245)	(136789,24510)	0.666	0.796	0.850	80.497177
(1234568,7)	(1234567910,8)	0.436	0.966	0.900	88.671280

Table-12 Results obtained through genetic algorithm approach

Machine Cells

Cell 1: m1, m3, m5, m6, m7, m8.

Cell 2: m2, m4.

Compactness of cells: **63.6%**

Setup time efficiency: **86.4%**

Group technology efficiency: **90%**

Performance measure (β): **85.58%**

Part families

Family 1: p1, p3, p6, p7, p8, p9.

Family 2: p2, p4, p5, p10.

6.1 Comparison of proposed method with Hiroshi Ohta approach

Consider the data used by Harhalakis, Nagi and Proth (1990) in table-1 showing the initial machine part incidence matrix (zeros not printed). Numbers in table-1 shows the operation sequence of part on machine. Table-2 shows the set up times. Let $a=b = 0.5$, $\lambda=0.50$, $\mu= 0.25$, $\theta=0.25$ and $d_{ji}=0.1$, by using the algorithm given in section 5.1, the best solution obtained is shown in

table-11, the performance measure β of this solution is $\beta=0.855$ (**85.5%**). The performance measures of the cell formations obtained by Harhalakis et al. (1990) and Nair and Narendran (1998) and Hiroshi Ohta and Nakamura (2002) and the proposed approach are $\beta= 0.6966$ (69.66%), 0.7411 (74.11%), 0.7449 (74.49%) and 0.855 (**85.5%**) respectively. Comparison of the proposed approach with the Hiroshi Ohta et al. approach is shown in table-13.

Comparison of Hiroshi Ohta Algorithm Vs Genetic algorithm

Variable	Hiroshi Ohta Algorithm	Proposed Genetic Algorithm
Machine cells	(1,3,6,7,8), (2,4,5) (1,3,5,6,7,8), (2,4) (1,2,4,5,6,8), (3,7) (3,7,8), (1,2,4,5,6)	(1,3,5,6,7,8), (2,4) (1,2,4,5,6,8), (3,7) (1,3,6,7,8), (2,4,5) (1,2,3,4,5,6,8), (7)
Part family	(1,3,6,7,8,9), (2,4,5,10) (1,3,6,7,8,9), (2,4,5,10) (1,2,3,4,5,7,9,10), (6,8) (1,2,3,4,5,9,10), (6,7,8)	(1,3,6,7,8,9), (2,4,5,10) (1,2,3,4,5,9,10), (6,7,8) (1,3,6,7,8,9), (2,4,5,10) (1,2,3,4,5,6,7,9,10), (8)



CPN	-----	0.636
		0.437
		0.666
		0.436
STE	-----	0.864
		0.864
		0.796
		0.966
GTE	-----	0.900
		0.900
		0.850
		0.900
Performance Measure β (%)	74.50	85.583977
	73.43	83.595337
	71.03	80.497177

Table-13 Comparison of Hiroshi Ohta Algorithm with the Proposed Genetic Algorithm

7. CONCLUSION

When solving a cellular manufacturing problem by means of the machines assignment method, the common problem encountered is how to group the machines and the parts. Most of the existing methods of grouping are indifferent to the processing sequence. Considering the processing sequence, the set up time is an important factor. In this paper, a genetic algorithm approach for grouping machines and part families by considering the sequence data with reduction in set up time between machines in the same cell is presented.

The performance measures of the cell formations obtained by Harhalakis et al. (1990) and Nair and Narendran (1998) and Hiroshi Ohta and Nakamura (2002) and the proposed approach are $\beta = 0.6966$ (69.66%), 0.7411 (74.11%), 0.7449 (74.49%) and 0.855 (**85.5%**) respectively. The results obtained through the Genetic Algorithm approach in the illustrative example were very much satisfactory. Considering the other production factors such as part volume, alternate routings for parts and processing time etc can extend this current research work. Studies are in progress in this direction using metaheuristics such as Tabu Search and Neural Networks.

References

1. Akturk, M.S., and Turkcan, A., 2000. Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research*, 38(1) 2327-2347.
2. Askin, R.G., and Chiu K. S., 1990. A graph partitioning procedure for machine assignment and cell formation in-group technology. *International Journal of Production Research*, 28(8) 1555-1572.
3. Askin, R.G., Cresswell, S. H., Goldberg J. B. and Vakharia A. J., 1991. Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research*, 29, 1081- 1100.
4. Burbidge, J. L., 1979. Group Technology in Engineering Industry (London: mechanical engineering Publications).
5. Carrie, S., 1973. Numerical Taxonomy applied to Group Technology and Plant Layout *International Journal of Production Research*. 11, 399-416.
6. Caux, C., Bruniaux, R., and Pierrelval, H: 2000. Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics*, 64 (1-3) 279- 284.
7. Chandrasekharan, M.P. and Rajagopalan, R., 1989. GROUPABILITY: Analysis of the properties of binary data matrices for group technology. *International Journal of Production Research*, 27(6) 1035-1052.
8. Chandrasekharan, M.P. and Rajagopalan, R., 1987. ZODIAC - An algorithm for



- concurrent formation of part families and machine cells. *International Journal of Production Research*, 25(6) 835-850.
9. Chen, J. and Heragu, S.S., 1999. Stepwise decomposition approaches for large scale cell formation problems. *European Journal of Operational Research*, 113(1) 64-79.
 10. Cheng, C.H., Gupta, Y.P., Lee, W.H. and Wong, K.F., 1998. A TSP-based heuristic for forming machine groups and part families. *International Journal of Production Research*, 36(5) 1325-1337.
 11. Choobineh, F., 1988. A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, 26(7) 1161-1172.
 12. Chu, C.H. and Tsai, M., 1990. A comparison of three array-based clustering techniques for manufacturing cell formation. *International Journal of Production Research*, 28(8) 1417-1433.
 13. Co, H.C. and Araar, A., 1988. Configuring cellular manufacturing systems. *International Journal of Production Research*, 26(9) 1511-1522.
 14. Dimopoulos, C., and Mort, N., 2001. A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems. *International Journal of Production Research*, 39(1) 1-19.
 15. Goldberg, D.E., 1989. Genetic Algorithms in Search Optimization, and Machine Learning, Addison-Wesley, Reading, MA.
 16. Gunasingh, K. R., and Lashkari, R. S., 1989a. Machine grouping problem in cellular manufacturing systems-an integer programming approach. *International Journal of Production Research*, 27(9), 1465-1473.
 17. Gupta, T. and Seifoddini, H., 1990. Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system. *International Journal of Production Research*, 28(7) 1247-1269.
 18. Holland, J. H., 1975. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan.
 19. Hiorshi Ohta, Masateru Nakamura., 2002. Cell formation with reduction in set up times. *Computers and Industrial Engineering*, 42, 317-327
 20. Joines, J.A., Culbreth, C.T. and King, R.E., 1996. Manufacturing cell design: An integer programming model employing genetic algorithms. *IIE Transactions*, 28, 69-85.
 21. José Fernando Gonçalves and Mauricio G.C. Resende., 2004. An evolutionary algorithm for manufacturing cell formation. *Computers in Industrial Engineering*, 47, 247-273.
 22. Khan, M., Islam, S. and Sarker, B., 2000. A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems. *International Journal of Production Research*, 38(3) 699-720.
 23. Khator, S.K. and Irani, S.K., 1987. Cell formation in-group technology: A new approach. *Computers in Industrial Engineering*, 12(2) 131-142.
 24. King, J.R. (1980) Machine-component grouping in production flow analysis: An approach using a rank order-clustering algorithm. *International Journal of Production Research*, 18(2) 213-232.
 25. King, J.R. and Nakornchai, V., 1982. Machine-component group formation in-group technology: Review and extension. *International Journal of Production Research*, 20(2) 117-133.
 26. Kumar, K. R., and Vannelli, A., 1987. Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research*, 25(12) 1715-1728.
 27. Kumar, K. R., Kusiak, A., and Vannelli, A., 1986. Grouping of parts and components in flexible manufacturing systems. *European Journal of Operations Research*, 24, 387-397.
 28. Kusiak, A. 1987. The generalized group technology concept. *International Journal of Production Research*, 25(4) 561-569.
 29. Kusiak, A. and Chow, W., 1987. Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, 6(2) 117-124.
 30. Lee, H. and Garcia-Diaz, A., 1993. A network flow approach to solve clustering



- problems in-group technology. *International Journal of Production Research*, 31(3) 603-612.
31. McAuley, J., 1972. Machine grouping for efficient production. *Production Engineer*, 51(2) 53-57.
 32. Mosier, C.T. and Taube, L., 1985a. The facets of group technology and their impact on implementation, *OMEGA*, 13(6) 381-391.
 33. Mosier, C.T. and Taube, L., 1985b. Weighted similarity measure heuristics for the group technology machine clustering problem. *OMEGA*, 13(6) 577-583
 34. Nair G. J & Narendran, T.T., 1988. CASE: A Clustering algorithm for cell formation with sequence data, *International Journal of Production Research*, 36(1), 157-179.
 35. Ng, S., 1993. Worst-case analysis of an algorithm for cellular manufacturing. *European Journal of Operational Research*, 69(3) 384-398.
 36. Ng, S., 1996. On the characterization and measure of machine cells in group technology. *Operations Research*, 44(5) 735-744.
 37. Onwubolu, G.C., Mutingi, M., 2001. A genetic algorithm approach to cellular manufacturing systems. *Computers and Industrial Engineering*, 39(1-2) 125-144.
 38. Plaquin, M., Pierreval, H., 2000. Cell formation using evolutionary algorithms with certain constraints. *International Journal Of Production Economics*, 64 (1-3) 267-278.
 39. Rajagopalan, R. and Batra, J.L., 1975. Design of cellular production systems: a graph-theoretic approach. *International Journal of Production Research*, 13(6) 567-579.
 40. Seifoddini, H., 1989. Single linkage versus average linkage clustering in machine cells formation applications. *Computers and Industrial Engineering*, 16(3) 419-426.
 41. Selim, H.M., Askin, R.G. and Vakharia, A.J., 1998. Cell formation in-group technology: Evaluation and directions for future research. *Computers and Industrial Engineering*, 34(1) 3-20.
 42. Shtub, A., 1989. Modeling group technology cell formation as a generalized assignment problem. *International Journal of Production Research*, 27(5) 775-782.
 43. Srinivasan, G. and Narendran, T.T., 1991. GRAFICS - A nonhierarchical clustering algorithm for group technology. *International Journal of Production Research*, 29(3) 463-478.
 44. Srinivasan, G., Narendran, T. and Mahadevan, B., 1990. An assignment model for the part-families problem in-group technology. *International Journal of Production Research*, 28(1) 145-152.
 45. Uddin, M. K., Shanker K., 2002. Grouping of parts and machines in presence of alternative process routes by genetic algorithm, *International Journal of Production Economics*, 76(3) 219-228
 46. Vannelli, A., and Kumar, K. R., 1986. A method for finding minimal bottleneck cells for grouping part-machine families. *International Journal of Production Research*, 24(2) 387-400.
 47. Vohra, T., Chen, D., Chang, J. and Chen, H., 1990. A network approach to cell formation in cellular manufacturing. *International Journal of Production Research*, 28(11) 2075-2084.
 48. Won, Y., 2000. Two-phase approach to GT cell formation using efficient p-median formulations. *International Journal of Production Research*, 38(7) 1601- 1613.
 49. Yasuda, K., Yin, Y., 2001. A dissimilarity measure for solving the cell formation problem in cellular manufacturing. *Computers and Industrial Engineering*, 39 (1-2) 1-17.
 50. Zhao, C., and Wu, Z., 2000. A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, 38(2) 385-395.