Monograph:
Morad, N. and Zalzala, A.M.S. (1995) The Formation of Manufacturing Cells Using Genetic Algorithms. Research Report. ACSE Research Report 575 . Department of Automatic Control and Systems Engineering

Department of
AUTOMATIC
CONTROL
and SYSTEMS
ENGINEERING
UNIVERSITY of SHEFFIELD

THE FORMATION OF MANUFACTURING CELLS USING GENETIC ALGORITHMS

N. MORAD and A.M.S. ZALZALA

Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom

Research Report #575
May 1995

Tel : +44 (0)114 2825250
Fax : +44 (0)114 2731729
EMail : rrg@sheffield.ac.uk
RRG **Robotics Research Group**

THE FORMATION OF MANUFACTURING CELLS USING GENETIC ALGORITHMS

N. Morad and A.M.S. Zalzala

Department of Automatic Control and Systems Engineering
University of Sheffield, Mappin Street
Sheffield S1 3JD
United Kingdom

ABSTRACT

This paper proposes a genetic-based algorithm to handle the multi-criteria optimisation problem associated in the formation of cells in Group Technology (GT). GT or Cellular Manufacturing (CM) is a concept where a manufacturing system is decomposed into subsystems or cells. This is done by grouping a variety of parts with similar shape, dimension or process route. This manufacturing concept allows small batch-type production to gain economic advantages similar to those in mass production and still retain the flexibility of job-shop production.

In this report, a genetic-based algorithm is developed to solve the cell-formation problem. Genetic Algorithm (GA) is an optimisation technique that imitates the survival-of-the-fittest concept. The advantages of applying the GA approach in this problem include producing more than one acceptable solution and using several objective functions.

To overcome the problem of multi-criteria optimisation associated in the formation of cells, the criteria are prioritised and modelled as multi-objective functions in the algorithm. Consequently, the algorithm is able to find a compromise between goals. Three different objective functions are used: minimising the inter-cell movement, minimising the variation of workload, and maximising the similarity of machines within cells.

1. INTRODUCTION

Group technology is a manufacturing concept which was conceived in Russia in 1959 by Mitrofanov. The idea behind GT is to decompose a manufacturing system into subsystems or manufacturing cells, by grouping a variety of part having similarities of shape, dimension or process route[7]. This concept is also known as Cellular Manufacturing (CM), where it allows small batch-type production to gain economic advantages similar to those in mass production and still retain the flexibility of job-shop production. This idea can also be applied in the FMS where the tool-changing and set-up time can be minimised by grouping similar parts to be processed together.

There are many advantages of Group Technology. Among them are creating mass-production effect, possibility of flow-shop pattern, reduction of set-up time and cost, GT layout simplifies material flow and handling, and saving production cost.

One of the most important step in GT is cellular formation, where parts with similar design features and processing requirements are grouped together into families and form associated machines into cells. The ideal situation would be when all parts in a family are fully processed within a single cell, which is difficult to accomplish in the real world.

Various heuristics and analytical methods have been proposed to solve the cell-formation problem. Generally, these methods can be grouped into two: the classification and coding system and the production flow analysis.

In the classification and coding system, grouping is done based on the features of the parts to be processed. This method is quite subjective, depending on the designer to decide on the appropriate code for each part. Hence this method is not as popular as the Production Flow Analysis [PFA] developed by Burbidge [2].

The main objective of the PFA is to find part-family based on the routing information. Most methods to solve the part-family problem is based on PFA, where many criteria have been chosen as objectives of grouping the parts. The criteria include minimising the Work-in-Process (WIP), minimising the inter cell movement and minimising the material handling.

Most approaches use the routing information obtained in the part-incidence matrix, A[i,j], as shown in Fig. 1. These methods [4,5,11] basically manipulate the part-incidence matrix by re-arranging the matrix using certain heuristics until a block-diagonal pattern is formed. The effectiveness of these methods is normally judged by the number of exceptional elements found, which are the number of 1's appearing outside the diagonal block denoting that the particular part has to go to different cells for processing.

Part m/c	1	2	3	4
1	0	0	1	0
2	1	1	0	0
3	1	0	0	0
4	0	0	1	1
5	0	0	1	0

Figure 1: Part-Machine Incidence Matrix

Other methods developed include the graph theoretic [14], integer programming [12], similarity-coefficient based method [13] which yield increased complexity as the number of parts and machines increase.

The main drawback of the above-mentioned methods is that they only use the part-incidence matrix without taking into consideration of other parameters such as processing times and machine capacity which are also important in the formation of cells [9].

Venugopal [15] first developed a genetic-based method for the part-family problem. However, in his method he uses different objective function to evaluate different populations. How the solution would be when the same population is used and the objective functions integrated are not investigated thoroughly. We developed the idea further using an integration of multi-objective functions which can be prioritised. A single population is used in the formulation.

The rest of the paper is arranged as follows: Section 2 will discuss the principles of GAs and the genetic-based algorithm for the cell-formation. Formulation of the fitness functions are included in Section 3. Section 4 discusses the results obtained and comparisons with other methods. Finally, Section 5 concludes the paper. In addition, Appendix A shows the matrices used in the simulation of Section 4.

2. A GENETIC-BASED APPROACH

2.1 Brief Overview of GAs

A Genetic Algorithms is an adaptive search technique which imitates the process of biological evolution. This idea was initiated by Holland [10] and later developed by Goldberg [8] by introducing the Simple Genetic Algorithm (SGA).

The underlying basis of GAs is represented by a population of chromosomes or strings, where the chromosomes are actually a representation of the system that needs to be optimised. Each chromosome is decoded using an objective value to obtain the respective fitness values. In addition, the chromosomes undergo three different kinds of processes, namely, reproduction, crossover and mutation.

Reproduction is a process in which the individual strings are duplicated based on the fitness value of the chromosome. The higher the fitness value, the higher the probability of the chromosome to be included into the next generation. *Crossover* is a process in which strings in a population are mated at random. A pair of parent strings will undergo crossing over at the crossover point where the bits will be swapped, thus, the resulting crossover will produce new strings which will be part of the new generation. *Mutation* is the occasional random alteration of the value of the string position, and is needed to overcome the probability that some useful genetic material might be lost in the crossover and recombination process.

After undergoing the above processes, the population is again decoded to get fitness values and the processes continue for a number of generations until the fitness values converge. The final population would contain the best chromosome representing the optimal solution to the problem.

2.2 A Method for Forming Cells

In this formulation, integers are used for the chromosome representation. The length of the chromosome (LIND) is the total number of machines (*m*) and the position of the gene represent the machine number. The gene represent the cell number. For example, chromosome {3 2 1 3 3 2 1 2} represents a system with eight machines and three cells, with the machines going into the cells as follows:

Cell 1: machine 3 and 7,
Cell 2: machine 2,6 and 8, and
Cell 3: machine 1,4 and 5.

The initial population is generated at random. Although the total number of cells (*k*) required is determined before the run, this number can be increased or decreased to determine the optimum number of cells for each problem. This can be achieved by running the program for a few times to check whether the fitness value decreases when *k* is changed. Since this formulation is to minimise the fitness function, the lower the fitness value obtained, the better the solution is. Fig. 2 shows an example of how the fitness function decreases when *k* decreases.

This population is then decoded based on the objection functions used, where the latter are explained in Section 3. Ranking is then applied with a selective pressure of 2.0 to prevent premature convergence as suggested by Baker [1]. An elitist strategy is used whereby a certain percentage of the population is retained for the next generation. This allow the GA to converge faster. 10% of the best population is generated into the next population in this problem.

A single point crossover with probability of 0.7 and a mutation probability of 0.05 are used in this problem. Selection is based on stochastic universal sampling. The programming is done using MATLAB with the GA Toolbox developed by Chipperfield et.al. [6].

3. THE OBJECTIVE FUNCTIONS

Three objective functions are applied to evaluate the populations.

(A) Minimise the intercellular movement of parts (function F1): By minimising the intercellular movements of parts, the WIP and the material handling can also be reduced. The equation used is adopted from Venugopal [15]:

$$F_1 = \sum_{j=1}^{n} N_j \left[\sum_{l=1}^{k} y_{jl} - 1 \right] \tag{1}$$

Notations:

j = part number

n = total number of parts

i = machine number

m = total number of machines

l = cell number

k = total cell number

N_j = production requirement of each part

$y_{jl} = 1$ if $\sum_{i=1}^{m} e_{ji} x_{il} > 0$

$= 0$ otherwise

$x_{il} = 1$ if ith machine i is in cell l

$= 0$ otherwise

E $= [e_{ji}]$ is an n x m matrix where $e_{ji} = 1$ if $t_{ij} > 0$

$= 0$ otherwise

t_{ij} = processing time matrix

(B) Minimise the variation of workload within cells (function F2): This criterion ensures that excessive WIP will not build up within cells. The capacity of machines, the processing times as well as the requirements of parts are considered in the algorithm developed.

We formulated the second fitness function (F2) as follows:

$$F2 = \sum_{l=1}^{k} (CL_l - AV_l)^2 \tag{2}$$

where:

CL = celload of each cell

$$CL_l = \sum_{i=1}^{m} (x_{il} \times WL_i)$$

WL = total workload of each machine

$$WL_i = \frac{t_{ij} \times N_j}{c_i}$$

AV = average workload for each cell,

$$AV_l = \frac{CL_l}{\sum_{l=1}^{m} x_{il}}$$

(C) Maximise the similarity of machines (function F3): The similarity coefficient used is adopted from Rajagopalan and Batra [14]. This criterion is to ensure that machines in each cells are as similar as possible to each other.

The similarity between machines is expressed as follows:

$$S_{ij} = \frac{X_{ij}}{X_{ii} + X_{jj} - X_{ij}} \tag{3}$$

where
Sij = similarity coefficient of machines i and j
Xij = number of components visiting machines i and j
Xii = total number of components visiting machine i
Xjj = total number of components visiting machine j

To minimise the dissimilarity of machines within cells, the following computation is done:

$$F3 = \sum_{l=1}^{k} (S_l - AVS_l)^2 \tag{4}$$

where
S_l = total similarity within cells
$S_l = \sum_{i=1}^{m} x_{il} \times S_i$
Si = similarity between machines
AVS = average similarity for each cell
$AVS_l = \frac{S_l}{\sum_{i=1}^{m} x_{il}}$

The advantages of using the genetic-based algorithm in this problem is that we can find a compromise solutions by giving different weightage to the fitness functions. The formulation of the total fitness function is done as follows:

$$F = \alpha F1 + \beta F2 + \gamma F3 \tag{5}$$

where α, β, γ are weightage for Fl, F2 and F3, respectively. By varying the values of α, β and γ, the decision maker can decide which function should be given a higher priority.

Genetic-based algorithm also produces a set of solutions, not a single solution. In addition, other parameters that are also important in the formation of manufacturing cells, e.g. processing times and machine capacity are included in the algorithm. Hence, the evaluation is based on a more comprehensive representation of the system.

In the formulation, a constraint is imposed to limit the maximum and minimum number of machines in a cell. This is done by imposing a certain penalty when the constraint is violated. The upper and lower limit of machines in the cells in the problem are considered to be eight and three, respectively, but can be changed to suit the particular problem.

4. SIMULATION RESULTS

The algorithm is tested using three part-incidence matrices which appear to be widely used in the literature for comparison of methods. The description of the matrices used is listed in Table 1. These matrices are shown in Appendix A. Since the algorithm developed required data for the processing times, the machine capacity and the total number of each parts required, hypothetical data are used for these parameters.

Matrix	**Description (machines x parts)**	**Source**
1	15 x 10	Chan and Milner[4]
2	14 x 24	King and Nakornchai[11]
3	16 x 43	Burbidge[3]

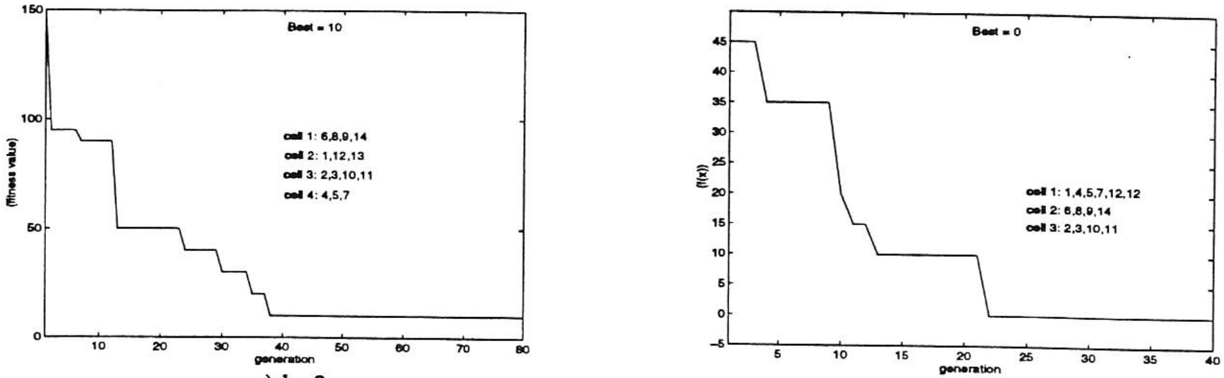
Table 1: Description of Matrices used

Using data with equal processing time (*t*), machine capacity (*C*) and part requirement (*N*) and working with Matrix 1 and Matrix 2 considering only F1, results obtained are exactly the same as obtained by [4][11]. Fig. 2 shows the results obtained using Matrix 2 and F1 using different values of *k* (i.e. number of cells). Fig. 2b gives a lower fitness value compared to Fig 2a. Since the objective is to minimise F1, the results obtained in Fig. 2b give a better solution.

We tested Matrix 1 initially using equal values for *N*, *t* and *C*. Then the values of *C* are changed to variable values since in the real-world, machine capacity are seldom the same. Fig. 3 shows results obtained using F1 and F2. Result shows that when *C* is changed, the final results of part-family obtained also changes. Consequently, this shows that the capacity of machine (*C*) do affect the formation of machine cell.

Fig. 4 shows results obtained by combining all three fitness function, F1, F2 and F3. Fig. 4b shows different results obtained when the part requirement (*N*) is made variable. Hence, this shows that the part-requirement also affect the formation of machine cells.

Fig. 5 illustrates the processing path taken by part 1 and part 5 in Matrix 1 using the algorithm. Fig 5a is at generation 1 and Fig. 5b at generation 60 and constant data for *N, C* and *t* are used in F1 and F2. At generation 60, the processing path of the parts are limited to within cells.



a) k=3

b) k=4

Fig. 2: Fitness Value (F1) vs. no. of generations (Matrix 2:14 x 24)

5. CONCLUSIONS

In this paper, we have developed a genetic-based algorithm to solve the cell-formation problem in Group Technology. The algorithm uses multi-criteria optimisation, in which criteria are based on the minimisation of inter-cellular movement, the minimisation of variation of workload and the maximisation of the similarity of machines within cells. This algorithm incorporates parameters such as the processing time of each part, machine capacity and parts requirement. Simulation results indicate the superiority of the GA in comparison with other heuristic methods. In addition, the multi-generation procedure provides less optimal alternative solutions which the decision maker may utilise.

REFERENCES

[1]Baker, J.E., 'Adaptive Selection Methods for Genetic Algorithms', *Proceedings of the First International Conference on Genetic Algorithms*, 1985, pp.101-111.

[2]Burbidge, J.L., 'Production Flow Analysis', *Production Engineer*, Vol 42, 1963, pp 742.

[3]Burbidge, J.L., *The Introduction of Group Technology*, Heinemann, London, 1975.

[4]Chan, H.M and Milner, D.A. ,'Direct Clustering Algorithm for group formation in cellular manufacturing', *Journal of Manufacturing Systems*, Vol 1, 1982, pp.65-75

[5]Chandrasekharan, M.P and Rajagopalan, R., 'ZODIAC: an algorithm for concurrent formation of part families and machine cells', *Int. Journal of Production Research*, vol 25,1987, pp 835-842.

✓ [6]Chipperfield, A., Fleming, P. Pohlheim, H., Fonseca, C. Genetic Algorithm TOOLBOX, For use with MATLAB:User Guide, 1994.

[7]Gallagher, C.C and Knight, W.A., *Group Technology*, Butterworths, 1973.

[8]Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[9]Heragu,S.S., 'Group Technology and Cellular Manufacturing', *IEEE Transactions on Systems, Man and Cybernetics*, Vol 24, 1994, pp.203-215.

[10]Holland, J.H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.

[11]King, J.R. and Nakornchai, V., 'Machine-component group formation in group technology: review and extension', *Int. Journal of Production Research*, 20, no 2, 117-133, 1982.

[12]Kusiak, A., 'The Generalized Group Technology Concept', *Int. Journal of Production Research*, Vol 25, No 4, 1987, pp.561-569 .

[13]Mc Auley, J., ' Machine Grouping for Efficient Production', *Production Engineer*, 1972, pp.53-57,.

[14]Rajagopalan, R. and Batra, J.L., 'Design of cellular production systems: a graph theoretic approach', *Int. Journal of Production Research*, vol. 13, no. 6, 1975, pp 567-579.

[15]Venugopal, V. and Narendran, T.T., 'A Genetic Algorithm approach to the machine-component grouping problem with multiple objectives', *Computers and Industrial Engineering*, Vol 22, No.4, 1992, pp 469-480.

Appendix A

Matrices used in the simulation

parts machines	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	1	0	0	0	0
2	1	0	0	0	0	0	1	0	0	1
3	0	1	0	0	1	0	0	1	0	1
4	0	0	0	1	0	1	0	0	1	0
5	0	1	0	0	1	0	0	1	0	0
6	0	0	1	0	0	1	0	0	1	0
7	0	0	0	0	0	0	1	0	0	1
8	0	1	0	0	1	0	0	1	0	0
9	0	0	1	1	0	1	0	0	1	0
10	1	0	0	0	0	0	1	0	0	1
11	1	0	0	0	0	0	1	0	0	1
12	1	0	0	0	0	0	1	0	0	1
13	0	1	0	0	1	0	0	1	0	0
14	0	0	1	1	0	1	0	0	1	0
15	0	1	0	0	1	0	0	1	0	0

a)15 x 10 matrix
Adopted from Chan and Milner [4]

parts machines	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
6	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0
7	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	1	0	0	0	0	0	1	0	0
9	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
12	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0

b)14 x 24 matrix
Adopted from King and Nakornchai [11]

part m/c	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1
6	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1	1
7	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1
9	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0
10	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
16	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0

c)16 x 43 matrix
Adopted from Burbidge[3]