



Asian Journal of Scientific Research

ISSN 1992-1454

science
alert
<http://www.scialert.net>

ANSI*net*
an open access publisher
<http://ansinet.com>

A Memetic Algorithm Approach for Minimizing Exceptional Elements in Cell Formation

¹R. Sivaprakasam and ²V. Selladurai

¹Department of Mechanical Engineering, Karpagam College of Engineering,
Coimbatore-641 032, Tamilnadu, India

²Department of Mechanical Engineering, Coimbatore Institute of Technology,
Coimbatore-641 014, Tamilnadu, India

Abstract: Cellular Manufacturing System (CMS) is an application of Group Technology (GT) in which similar parts and machines are grouped into part families and machine cells. In this study, a metaheuristic called Memetic Algorithm (MA) is introduced to solve the machine cell formation problem. This study is conducted to minimize the intercellular movement of parts known as exceptional elements. MA is incorporated using Genetic Algorithm (GA) and Tabu Search (TS) Algorithm. In the MA approach, local optimization (TS) is applied to each newly generated offspring at the end of genetic algorithm. The MA is tested on a number of problems of various sizes and its performance is evaluated. The results obtained by MA are highly comparable with an objective obtained by Metaheuristics GA, TS and there is a considerable reduction in computational effort.

Key words: Cellular manufacturing system, cell formation, memetic algorithm, genetic algorithm, tabu search

INTRODUCTION

Cell formation, one of the most important stages in CMS, is to group parts with similar geometry, function, material and process into part families and the corresponding machines into machine cells. Design of cellular manufacturing systems (CMSs) is a complex, multi-criteria and multi-step process. In the design of CMSs, design objective (s) must be specified. Minimizing inter-cell moves, distances, costs and the number of exceptional parts (parts that need more than one cell for processing) are common design objectives. An exceptional part can be also called an exceptional element or a bottleneck part. An example is given in order to introduce some of the terminology to be used in this study.

An example from Kusiak (1992) of 5 part types and 4 machine types was used in order to form cells. A machine-part matrix is one way to represent the processing requirements of part types on machine types as shown in Table 1. A 1 entry on row i and column j indicates that part type j has one or more operations on machine type i. For instance, part type 1 has operations on machine types 1 and 3. Manufacturing cells are formed with the objective of minimizing inter-cell moves. Two cells (clusters) are formed as shown in Table 1. Cell 1 consists of machine type 2 and 4 and produces part type 5 and 2. Cell 2 consists of machine type 1 and 3 and produces part type 3, 1 and 4. Part type 3 needs to be processed on machine type 1 and 3 in cell 2, however, it also needs to be processed on machine type 2 which is assigned in cell 1. Therefore, an inter-cell move is required: the symbol * represents an inter-cell move of part type 3. Part type 3 is an exceptional part, so these two cells (clusters) are called partially separable.

Corresponding Author: R. Sivaprakasam, Department of Mechanical Engineering, Karpagam College of Engineering,
Coimbatore-641 032, Tamilnadu, India

Table 1: Illustration of machine-part matrix before and after cell formation

Initial machine part matrix before cell formation						Final machine part matrix after cell formation					
Part type	1	2	3	4	5	Part type	5	2	3	1	4
Machine type						Machine type					
1	1		1	1		2	1	1	*		
2		1	1		1	4	1	1			
3	1		1			1			1	1	1
4		1			1	3			1	1	0

Analogous to an exceptional part, a bottleneck machine is one that processes parts belonging to more than one cell. Two possible approaches to eliminate exceptional parts are by considering alternative process plans for parts or additional machines. In Table 1, 0 represents a void in cell 2. A void indicates that a machine assigned to a cell is not required for the processing of a particular part in the cell. In this example, machine type 3 is not necessary for part type 4. The presence of voids leads to inefficient large cells, which in turn could lead to additional intra-cell material handling costs and complex control requirements. Any cell configuration should satisfy operational goals (constraints) such as desired machine utilization, production volume, number of manufacturing cells, cell sizes, etc.

In the last three decades, several research papers and practical reports have been published in the field of CM, seeking effective methods for designing CMSs. Reviews of existing CM literature can be found in Joines *et al.* (1996) and Selim *et al.* (1998). According to those reviews, the existing CM design methods in the CMSs can be classified into the following categories: part coding analysis, cluster techniques, similarity co-efficient, graph partitioning, mathematical programming, heuristic search and AI-based approaches.

Each design approach considers different numbers of design objectives and constraints to different extent, depending upon the scope and interest of each. For instance, clustering analysis approaches consider only one objective, the minimization of inter-cell moves. In the design process of clustering techniques, only part operations and the machines for processing those operations are considered. Other product data (such as operational sequences and processing times) and production requirements (such as production rate) are not incorporated into the design process. Thus, solutions obtained may be valid in limited situations. However, they are simple to implement and solutions can be obtained in reasonable amounts of time. Minimizing inter-cell flows of parts is fundamental to achieving many of the benefits associated with CM (Cao and Caen, 2005). The cell formation problem is complicated by the existence of exceptional parts and/or exceptional machines. Both exceptional parts and exceptional machines cause intercellular movement of parts. Ideally a part-cluster is processed in a single machine cell for its entire operations. In practice, however, it is a very rare case. Extensive work has been done by many researchers to provide new techniques for solving this problem.

The recent shift toward heuristic solutions to the Machine-Part Cell Formation (MPCF) problem is due, in part to the fact that the problem is NP-complete. With heuristic approaches showing promise in this area, present study focuses on the application of genetic algorithm, tabu search and memetic algorithm to the MPCF problem. An objective function provides the basis for evaluating the machine groupings arrived at by a searching method such as GA, TS. This work uses the metaheuristics of GA, TS and memetic algorithm for cell formation with an objective of minimizing the exceptional elements. The proposed memetic algorithm approach combines genetic algorithm with tabu search heuristic. The objective of the heuristic is to construct a set of machine/product groups and improve it, if possible. The proposed Memetic Algorithm is validated with the test cases studied in the literature and comparisons are presented.

METAHEURISTIC ALGORITHMS

The major drawback of mathematical programming approaches is computational timing required for large problems. So heuristic approaches have been used as alternatives to obtain reasonably good

solutions. Nowadays metaheuristics are widely used to solve important practical combinatorial optimization problems. A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. Examples of metaheuristics include Simulated Annealing (SA), tabu search (TS), genetic algorithm and Ant Colony Optimization (ACO).

GENETIC ALGORITHM

Genetic Algorithms are search and optimization procedures that are motivated by the principle of natural genetics and natural selection. GA is a metaheuristic for solving combinatorial optimization problems (Goldberg, 1989). GA is a heuristic solution technique that works by encoding a population of solutions to a given problem and manipulating these solutions through the use of operators such as crossover and mutation in an attempt to evolve superior solutions. The new population is further evaluated and tested for termination. If the termination criterion is met, GA process stops otherwise, the above cycle is followed until the termination criterion is met.

GENETIC OPERATORS

Reproduction

Reproduction is usually the first operator applied on population. Reproduction selects good strings in a population and forms a mating pool. There exist a number of reproduction operators in GA literature and rank selection method is used for reproduction. The individuals in the population are ranked according to fitness and the expected value of each individual depends on its rank rather than on its absolute fitness.

Reproduction selects good strings in a population and forms a mating pool. The linear ranking method proposed by Baker (1987) is as follows: Each individual in the population is ranked in increasing order of fitness from 1 to N. The expected value of each individual i in the population at time t is given by:

$$\text{Expected value}(i,t) = (\min) + (\max - \min) \times \frac{\text{rank}(i,t) - 1}{N - 1} \quad (1)$$

Where:

N = Sample size.

Min = 0.4.

Max = 1.6.

After calculating the expected value of each rank, reproduction is performed using Monte Carlo simulation by employing random numbers.

Crossover

In the crossover, new strings are created by exchanging information among strings of the mating pool. In single point crossover, one crossover point is selected; binary string from beginning of chromosome to the crossover point is copied from one parent and the rest is copied from the second parent.

Mutation

Mutation is also done randomly for each gene and it depends upon another parameter called mutation probability. Here one gene is selected at random and the mutation operation is performed.

Tabu Search

Tabu search is a meta-strategy for guiding known heuristics to overcome local optimality. Tabu Search (TS) has its antecedents in methods designed to cross boundaries of feasibility or local optimality standard treated as barriers and to systematically impose and release constraints to permit exploration of otherwise forbidden regions.

The philosophy of Tabu Search is to derive and exploit a collection of principles of intelligent problem solving. A fundamental element underlying tabu search is the use of flexible memory. From the standpoint of tabu search, flexible memory embodies the dual processes of creating and exploiting structures for taking advantage of history (hence combining the activities of acquiring and profiting from information). TS methods operate under the assumption that a neighborhood can be constructed to identify adjacent solutions that can be reached from any current solution (Baykasoglu and Gindy, 2000; Tavakkoli and Arunlzhaz, 2005).

TS Operators

- **Move attribute:** The pair of sites of the string being swapped.
- **History record:** List of sites of the string classified under Tabu restrictions with record of on which iterations each of them classified as tabu.
- **Tabu classification/Restriction:** The sites of string swapped in the previous iterations will not be considered for swapping.
- **Tabu tenure:** The tabu restriction of a site is lifted after a consecutive three iterations.
- **Aspiration criterion:** Tabu restrictions are lifted for the solutions under tabu classification, with the value of the objective, 10% or more, less than that of the current solution.
- **Choice criterion:** The solution with the minimum Objective value among the neighboring solutions of the current solution.
- **Termination criteria:** Reaching a predefined minimum value of objective or 50 numbers of iterations whichever occurs first.

Memetic Algorithm

GA are not well suited for fine tuning structures, which are close to optimal solutions. MAs are EAs that apply a separate local search process to refine individuals (i.e., improve their fitness by tabu search algorithm). Under different context and situations, MAs are also known as hybrid EAs, genetic local searchers (Muruganandam *et al.*, 2005). The operation of MA begins with a population of random strings representing design and decision variables. Thereafter each string is evaluated to find the objective value. The population is then operated by 4 main operators reproduction, crossover, mutation and TS.

Pseudo-code for a simple MA

```
Begin
  Initialize population;
  Evaluate population;
  Repeat
    Reproduction;
    Crossover;
    Mutation;
    Tabu search;
  Until (termination criteria);
End.
```

Problem Formulation

In this study, an objective of minimization of exceptional elements is considered to evaluate the goodness of the cell formation.

Minimization of exceptional elements is considered as the objective function (Z)

$$Z = \sum_{i=1}^m \sum_{j=1}^n a_{ij} \left(\sum_{k=1}^G |x_{ik} - y_{jk}| \right) / 2 \quad (2)$$

Where:

- k = Cell index
- G = No. of manufacturing cell
- M = No. of machines
- n = Total number of parts
- x_{ik} = Binary value indicating if machine i is assigned to cell k
- y_{jk} = Binary value indicating if part j is assigned to cell k
- a_{ij} = Element of machine part incident matrix

Decision Variables

- X_{ik} = 1 if machine type i is assigned to cell k.
= 0, otherwise
- Y_{jk} = 1, if part/component j is assigned to cell k
= 0, otherwise

Representation

Representation forms a key role in the development of Genetic Algorithm, Tabu Search and Memetic Algorithm. A problem can be solved once it is represented in the form of solution string. In the problem, each gene represents whether the machine or part is in that cell or not.

Model Illustration

For 16×30 size problem (Bector, 1991), each cell or the string is coded and decoded as shown in Table 2.

After representation, an initial solution can be generated using random numbers and the generated solution is subjected to iterations or generations.

Parameters Selection

The appropriate values of the GA parameters are arrived at, based on the satisfactory performance of the trials conducted for this application with different ranges of values. The crossover probability varied from 0.4 to 0.9 and it was found that the solution was improving faster for a crossover probability of 0.60. Similarly in the range from 0.001 to 0.020, the mutation probability of 0.015 was found to retain better solutions than worse solutions.

Table 2: Coded and decoded information for 16×30 problem

Cell No.	Machines list		Component list	
	Coded	Decoded	Coded	Decoded
Cell 1	0010001010011100	3,7,9,12,13,14	11011000111110101 1111101011010	1,2,4,5,9,10,11,12,13, 15,17,18,19,20,21, 22,24,26,27,29
Cell 2	1101110101100011	1,2,4,5,6,8,10,11, 15,16	00100111000001010 0000010100101	3,6,7,8,14,16,23,25,28,30

COMPUTATIONAL RESULTS

To demonstrate the performance of the proposed MA, 7 Group Technology (GT) problems of different sizes from the information collected from the literature were tested. The selected matrices range from dimension 4×18 to 24×40 . The matrix sizes and their sources are shown in the Table 3.

The objective function of number of exceptional elements obtained by the proposed MA with that of GA and TS are compared and the algorithm was coded in C++.

From the test results, it is observed that out of 87 trials for the 7 problems with total number of cells as 2 and 3 for different possibilities of minimum number of machines which could be accommodated in either 2 cells or 3 cells, the objective function of MA is minimum or equally good for 72 cases compared to GA and TS. Similarly GA has the minimum or equal number of exceptional elements for 14 cases out of 87. And TS has the minimum or equal number of exceptional elements for 11 cases out of 87 trials (Table 4).

The computational experience has shown that for most of the problems, MA results in solutions with lower objective function value. Even though MA yields higher values in few cases compared to other approaches, it results in better solution and requires very little computational effort.

Table 3: Selected GT problems from the literature

Problem size	Source
4×8	Chang and Lee (2000)
9×9	Al-Ahmari (2002)
12×10	Al-Ahmari (2002)
16×30	Boctor (1991)
20×20	Harhalakis <i>et al.</i> (1990)
23×20	Chang and Lee (2000)
24×40	Venugopal and Narendran (1993)

Table 4: Comparative results of objective function (exceptional elements) for GA, TS and MA with different cell sizes

S. No.	P. No	Size	No. of cells	Min. No. of M/Cs	Z		
					GA	TS	MA
1	1	4×18	2	1	4	11	5
2				2	8	8	5
3			3	1	11	9	8
4	2	9×9	2	1	4	5	3
5				2	6	5	3
6				3	7	8	3
7				4	3	3	3
8			3	1	8	8	5
9				2	7	10	8
10				3	16	10	8
11	3	12×10	2	1	0	5	3
12				2	0	8	0
13				3	3	8	5
14				4	3	8	2
15				5	6	6	5
16				6	8	8	5
17		3		1	9	9	4
18				2	12	8	10
19				3	11	10	5
20				4	11	12	7
21	4	16×30	2	1	16	19	5
22				2	13	17	11
23				3	15	17	11
24				4	26	17	17

Table 4: Continued

S. No.	P. No	Size	No. of cells	Min. No. of M/Cs	Z		
					GA	TS	MA
25				5	17	15	11
26				6	17	15	21
27				7	31	11	11
28				8	39	11	31
29			3	1	40	39	15
30				2	33	41	26
31				3	45	30	33
32				4	37	35	40
33				5	55	37	36
34	5	20×20	2	1	8	10	6
35				2	9	10	6
36				3	10	10	6
37				4	6	7	6
38				5	12	12	8
39				6	6	12	9
40				7	15	12	6
41				8	15	8	7
42				9	14	12	8
43				10	20	12	8
44			3	1	19	13	10
45				2	22	11	16
46				3	28	23	12
47				4	30	15	17
48				5	26	12	14
49				6	39	19	15
50	6	23×20	2	1	15	25	6
51				2	28	26	7
52				3	12	27	19
53				4	10	23	7
54				5	20	18	9
55				6	29	28	11
56				7	26	24	14
57				8	18	24	15
58				9	21	27	17
59				10	37	27	25
60				11	36	32	24
61			3	1	33	37	21
62				2	23	33	10
63				3	30	48	24
64				4	37	33	21
65				5	43	34	26
66				6	42	35	34
67				7	46	46	44
68	7	24×40	2	1	0	3	0
69				2	1	3	1
70				3	4	7	1
71				4	3	4	2
72				5	1	4	1
73				6	2	15	6
74				7	4	8	1
75				8	1	9	1
76				9	5	2	4
77				10	7	7	1
78				11	13	14	1
79				12	43	6	5
80			3	1	15	16	9
81				2	43	21	21
82				3	27	13	15
83				4	31	6	6
84				5	36	7	13
85				6	59	7	9
86				7	27	17	6
87				8	50	20	11

CONCLUSION

A MA approach for obtaining machine cells and product families has been presented. Computational experience with the algorithm, on a set of Seven GT problems from the literature has shown that the MA heuristic performs better than GA and TS algorithm as far as the objective function of minimizing the exceptional elements are concerned. It is inferred from the results obtained that the proposed MA heuristic is efficient either in the quality of the solutions or in the speed that gets the solutions.

REFERENCES

- Al Ahmari, A.M.A., 2002. Fuzzy analysis approach for part-machine grouping cellular manufacturing systems. *J. Integrated Manufact. Syst.*, pp: 489-497.
- Baker, J.E., 1987. Adaptive Selection Methods for Genetic Algorithms. In: *Proceedings of the 1st International Conference on Genetic Algorithms and Their Application*. Grefenstette, J.J. (Ed.), Lawrence Erlbaum Associates, pp: 14-21.
- Baykasoglu, A. and N.N.Z. Gindy, 2000. Multiple objective capability based approach to form part-machine groups for cellular manufacturing applications. *Int. J. Prod. Res.*, 38: 1133-1161.
- Boctor, F.F., 1991. A linear formulation of the machine part cell formation problem. *Int. J. Prod. Res.*, 29: 343-356.
- Cao, D. and M. Caen, 2005. A robust cell formation approach for varying product demands. *Int. J. Prod. Res.*, 43: 1587-1605.
- Chang, P.T. and E.S. Lee, 2000. A multisolution method for cell formation. Exploring practical alternatives in group technology manufacturing. *Int. J. Comput. Mathematics Applications*, 40: 1285-1296.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search. Optimization and Machine Learning*. Addison-Wesley: Reading, MA.
- Harhalakis, G., R. Nagi and J.M. Andproth, 1990. An efficient heuristic in manufacturing cell formation for group technology application. *Int. J. Prod. Res.*, 28: 185-198.
- Joines, J.A., C.T. Culberth and R.E. King, 1996. Manufacturing cell design: An integer programming model employing genetic algorithms. *IIE Trans.*, 28: 69-85.
- Kusiak A., 1992. The generalized group technology concept. *Int. J. Prod. Res.*, 25: 561-569.
- Muruganandam, A., G. Prabhakaran, P. Asokan and V. Baskaran, 2005. A memetic algorithm approach to the cell formation problem. *Int. J. Adv. Manufact. Technol.*, 25: 988-997.
- Selim, H., R. Askin and A. Vakharia, 1998. Cell formation in group technology: Review, evaluation and directions for future research. *Int. J. Comput. Ind. Eng.*, 34: 3-20.
- Tavakkoli, M.R. and R. Arynlzhad, 2005. Solving a dynamic cell formation problem using metaheuristics. *Int. J. Math. Comput.*, 170: 761-780.
- Venugopal, V. and T. Narendran, 1993. Design of cellular manufacturing systems based on asymptotic forms of a boolean matrix. *Eur. J. Operat. Res.*, 67: 405-417.