POSSIBILITIES OF GA IN OPTIMIZATION OF MANUFACTURING CELL FORMATION

V. Modrak[1], S. Pandian[2] and P. Knuth[1]
[1]Faculty of Manufacturing Technologies, Technical University of Kosice (TUKE) Bayerova, Presov, Slovakia
[2]Kalasalingam University Anandnagar, Krishnankoil, (Via) Srivilliputtur, Virudunagar Dist., Tamil Nadu, India
E-Mail: vladimir.modrak@tuke.sk

ABSTRACT
Optimization of manufacturing cell formation includes a whole virtual cell formation procedure and has NP-hard complexity. In the last decades, evolutionary algorithms have been widely applied to real-world optimization including to manufacturing cell formation problems. Genetic algorithms (GAs) are possibly the most widespread variant of evolutionary algorithms and are now frequently used for a number of optimization problems in operation management. In this study it is discussed about theoretical and practical aspects of GA to bridge the gap between hypothetical analysis and real time implementation of GA in a given domain.

Keyword: cell formation, genetic algorithm, cellular manufacturing.

INTRODUCTION
In manufacturing firms, characterized by a high variety and low to medium volume product, mix the conventional layout approach is process oriented that is very flexible and insures high equipment utilization rates. An alternative to this traditional layout is the use of Cellular Manufacturing Systems (CMS) that combines the advantages of both product and process oriented systems. Cellular manufacturing (CM) is an application of the group technology (GT) philosophy to designing manufacturing systems. GT approach refers to grouping of parts to be manufactured according to similarities derived using various characteristics and processing them on their requisite machines placed close together in a group called a cell.

In industry, it is not easy to make all parts and machines into stand alone cells, and therefore a separate shop to look after such kind of special operations becomes mandatory. There are firms having some operations outside the cells for different products and spare parts whose changing demand patterns tend to disturb the cell routine works. It has also been pointed out that as the range of parts becomes wider, with parts in different stages of their lifecycles, it is preferable for the manufacturing system to be configured in a hybrid manner [1]. In addition, if there is only one machine or facility of a certain type (such as painting, furnace equipment), used for processing a wide range of parts, it would be advantageous to retain such special machine types in the functional layout. The processing times of these machines are often significantly different, preventing them from working in harmony with other parts in the cells. The cost of duplicating these machines and placing them in multiple cells is often high. As a result, these special machine types (which may also be used by the stable-demand parts processed in the cells) may have to be located in the functional layout. There are number of objectives need to be solved in a real time environment. For example, grouping efficiency of the cells, inter-cell movement, logistics handling and so on. There have been numerous methods available in the literature including modern ones like Simulated Annealing (SA), Genetic Algorithms (GA), Tabu Search (TS), Greedy approaches, Variable-Depth search, Hill climbing procedures, and Ant Colony Optimization (ACO), which can be used to solve such objective functions theoretically. While going for real implementation in shop floor, how many of them do effectively work? In this paper a potential of genetic algorithms for manufacturing cell formation will be considered.

Genetic Algorithm is a computerized search and optimization algorithm based on the mechanics of natural genetics and natural selection. GA is a search technique for global optimization in a search space. As the name suggests, they employ the concepts of natural selection and genetics using past information for directing the search with expected improved performance to achieve fairly consistent and reliable results. The traditional methods of optimization do not work sufficiently over a broad spectrum of problem domain. GA attempts to mimic the biological evolution process for discovering good solutions. They are based on a direct analogy to Darwinian natural selection and mutations in biological reproduction and belong to a category of heuristics known as randomized heuristics that employ randomized choice operators in their search strategy and do not depend on complete a priori knowledge of the features of domain. These operators have been conceived through abstractions of natural genetic mechanisms such as crossover and mutation and have been cast into algorithmic forms. Holland [2] envisaged the concept of these algorithms in the mid-sixties and it has been applied in diverse areas such as music generation, genetic synthesis, fault diagnosis, strategy planning and also to address business problems such as travelling salesman problem, production planning and scheduling problem, facility location problem, transportation problems, telecommunications and network problems, engineering design problems and image processing and cell design problems. GA is different from traditional optimization and search

VOL. 6, NO. 7, JULY 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

techniques in the following ways. It works with a coding of parameters; not with parameter themselves. GA searches from population of points; not from a single point. It uses probabilistic rules rather than deterministic rules. In GA, the solution is represented in terms of specific coding, for which the number of generations or iterations needs to be generated (see Figure-1). The best solution will be searched in a solution space and narrowed down as per the requirement.

- Generate and evaluate the initial population P (t), t = 0.
- Repeat the following steps until stopping condition is satisfied.
  - Selection of chromosomes from the current population
  - Apply the crossover over the parent chromosomes and produce two offsprings.
  - Apply the mutation operator over the offspring.
  - Copy the offspring to population P (t + 1).
  - Evaluate P (t + 1).
  - Replace the worst chromosome of P(t + 1) by the best chromosome found so far.
- Set t = t + 1.
- Return the best chromosome found.

Figure-1. Pseudo code of genetic algorithm.

APPROACHES TO CELL FORMATION

The manufacturing industries having batch production environment are determined to achieve reduced lead time, reduced setup time, and increased machine utilization. Cellular manufacturing stands as one of the efficient proposition of achieving the goal in this direction.

Cellular manufacturing overcomes major problems of batch-type manufacturing including frequent setups, excessive in-process inventories, long throughput times, complex planning and control functions etc. and provides the basis for implementation of manufacturing techniques such as Just-In-Time (JIT) and Flexible Manufacturing Systems (FMS). There are different problems regarding manufacturing cell formation that can be solved by modern methods as shown in Table-1.

Table-1. Different clustering problems and suitable solution methodologies.

Clustering problem	Methodological approaches
Machine grouping	Genetic algorithm [9]
Part grouping	Fuzzy ART [10] genetic algorithm [11]
Machine and part grouping based on the PFA	Modified adaptive resonance theory [12], [13], [14], [15], [16], [17], [18], [19], [20]

The aim of the PFA technique is finding the families of components and associated groups of machines for group layout by a progressive analysis of the information in route cards. It is based on the idea that parts with similar routes can be made in the same group, and it finds both a division of machines into groups and of parts into families of parts, which they make. The similarity coefficient approach was first suggested by [3]. The basis of these methods is to measure the similarity between each pair of machines and then to group the machines into families based on their similarity measurements. Rajagopalan and Batra [4] suggested the use of graph theory to form machine groups. Later on, Boctor [5] performed the application of linear programming in cell formation. Waghodekar and Sahu [6] proposed an algorithm called MACE to solve the GT problem.

Since 1990 the applications of soft computing techniques to GT problems have been encouraging [7]. The literature concerning CMS using major soft-computing techniques like fuzzy set theory, meta-heuristics, and artificial neural networks have been discussed by Sudhakara Pandian and Mahapatra [8]. Summarily, possible cell formation methods, traditional and modern ones are depicted in Figure-2.

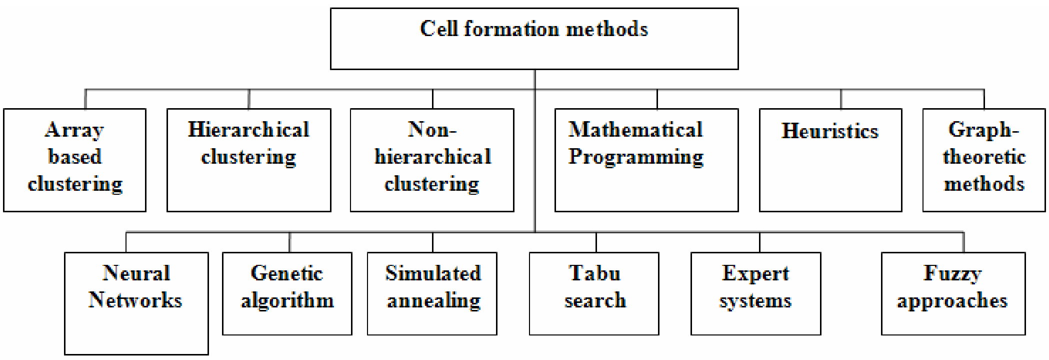

Figure-2. Taxonomy of cell formation methods.

VOL. 6, NO. 6, JUNE 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

REAL POSSIBILITIES OF GENETIC ALGORITHM

As stated in the introduction section, GA can handle typical optimization problem such as cell formation problems, scheduling problems, transportation problems, resources allocation problems and handling of logistics in supply chain systems etc… but not limited to. In GA a candidate solution represented by sequence of genes called chromosome. A chromosome potential is called its fitness function, which is evaluated by the objective function. A set of selected chromosomes is called population and the population is subjected to generations (number of iterations). In each generation crossover and mutation operators are performed to get new population. A GA has a number of advantages. It can quickly arrive at good solution set. As worse cases are simply discarded, they will never affect the solution. In general GA will not know the rules of the problem, but it works by its own rules. This is a very useful strategy of GA for problems of complex in nature. In GA, it is enough for a problem solver to choose the appropriate coding. The coding is nothing but another form of one of the solutions. If the solutions are coded in different combinations then the GA will start its searching operation using its operators known as selection, crossover and mutation as shown in Figures 3 and 4, respectively. All these methods are probabilistic approaches. The proper stopping criteria will be given as input for the GA to stop its searching process. This is done purely based on experience of the problem solver. Based on the stopping criteria the GA will stop running and give the solution what it finds at that point of time.

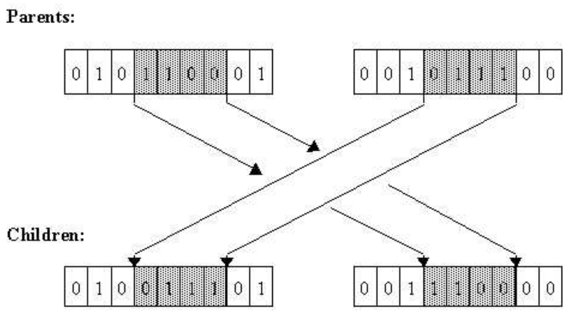

Figure-3. Crossover on binary coded representation of chromosomes.

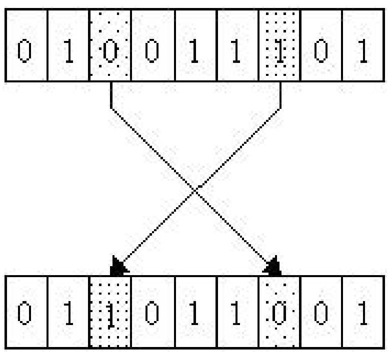

Figure-4. Mutation within chromosome.

In cell formation GA works well and finds good solution as given in different literature studies. It gives better and better solutions even if the problem grows with high level of complexity. The following representation is used in typical cell formation problem solved using GA. This representation is popularly known as real coding (Figure-5).

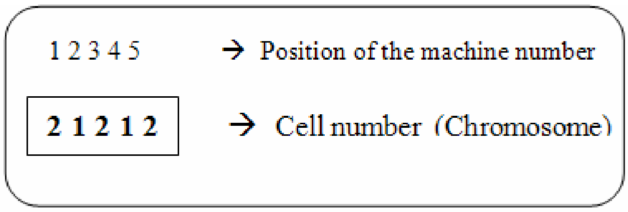

Figure-5. Representation of chromosome in cell formation problem using real coding.

Depends on number of cells to be accommodated in the layout, the number of genes on a chromosome get increased. In the above example there are only two cells and hence there are only 1 and 2 values present in the chromosome of 5 genes (5 machines).

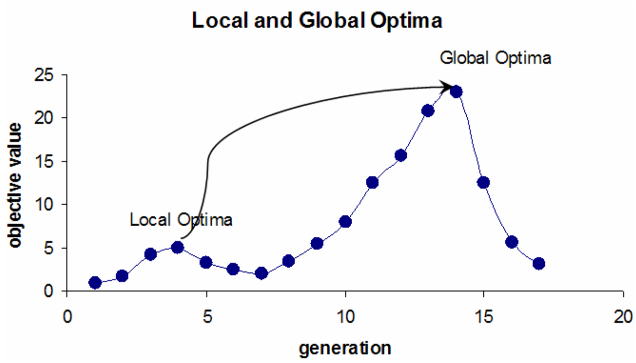

Figure-6. Local and global optima.

In GA, mutation has a special character that at times it can jump from local optimal value to global optimal value thereby within quicker time the required solution could be obtained as shown in Figure-5.

Table-2 shows typical cell formation problems with problem size and the results obtained in terms of the Computer Process Unit (CPU) time taken to produce the results using genetic algorithm.

It is observed from Table-2, that whenever problem size gets increased the number of generations is increased to get the required optimum value and hence the CPU time also gets increased. But, as far as population size is concerned, it is fixed by the problem solver to any number without following any standard rule. But based on experience, for the smaller population size the convergence of GA will be smoother.

Table-2. Results of CPU time obtained from cell formation problem.

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

In order to compare CPU time of GA, with other techniques it can be taken example from a research work that has been focused on comparison of GA, SA and hybrid approach based on the genetic algorithm and artificial neural network (ANN) to design an real cellular manufacturing system [21]. The partial results from this research depicted in Table-3 show that there is significant difference between the ANN, GA and SA algorithms.

Table-3. Results of computational time obtained from cell formation problem.

Problem size	Computational time (sec.)		
	GA	SA	ANN
Small $_{min}$.	1	1	0,58
Small $_{max}$.	24	24	20
Medium $_{min}$.	120	76	60
Medium $_{max}$.	180	138	76
Large $_{min}$.	350	183	155
Large $_{max}$.	1521	673	550

The hybrid algorithm is computationally faster as compared to GA and SA and a quality of solutions produced by the proposed ANN algorithm is much better than those generated by the GA and SA. It also outlines a general potential of GAs to create hybrid approaches that can bring competitive solutions.

PRACTICAL OBSTACLES AND CHALLENGES OF GENETIC ALGORITHM

Even though there are many useful hypothetical cases found to be better in giving good solutions for cell formation problems, when the practicality rather than theoretical approaches is concerned, the question is whether GA still works better. The response from the researchers and the industrial engineers reflects a big gap. There are following useful points highlighted from this work to realize the practical obstacles and challenges facing the researchers and industrial engineers who are actively engaged in bridging the gap.

a) The GA is simply a probabilistic approach which means it works with probabilities and there is no deterministic rules governing GA. This shows that there is no guaranteed solution of our requirement. Instead, we assume what ever solutions are arrived from the output of GA is the best one.
b) There are many influencing parameters which work with random numbers in GA. For examples crossover and mutation types and their probabilities, selection method and probability of selection and number of iterations. These values are fixed by us without any practical knowledge but by simple refereeing the values suggested from literature for different environment.
c) While the great advantage of GA is the fact that they find a solution through evolution, this is also the biggest disadvantage. Evolution is inductive; in nature life does not evolve towards a good solution - it evolves *away* from bad circumstances. This can cause a species to evolve into an evolutionary dead end. Likewise, GA produces a suboptimal solution and we may not even know it. The property of *convergence* is a big phenomenon in GA. This is a major drawback too. For instance, when the problem solver gives all required input to start the GA search process for results, it gives certain values which we mean it as an optimal solution at that point of time. But actually that result may not be an optimal and there may be far better results available in reality. The reason is that, sometimes in a search process of GA there may be a local optimal point as shown in Figure-5 and if the GA falls in the local optimal point it can not go further to search for global optimal which is actually required by us. This property of falling within the local optimal space is known as convergence. This is a pitfall in GA searching procedure.
d) The crossover and mutation probabilities are fixed to be 0.5 and 0.1, respectively for the bench mark study as mentioned in the literature [12]. It is also stated that this probability can be varied depending upon the decision maker to tune the algorithm. But without knowing how to choose the values for crossover and mutation probabilities, how could the problem solver vary these values to find better results.
e) The chromosome representation may sometime results in the formation of an empty cell or violates some constraints. This means that GA will find some solutions which may not exist in reality or difficult to implement in practical situations.

These above mentioned obstacles are to be carefully analysed. They are the real challenges facing the

ARPN Journal of Engineering and Applied Sciences

industrial engineers who are interested in bridging the gap between the hypothetical analysis and the real time situations

CONCLUSIONS

In this study cell formation problem with GA is discussed. Form this study, it is understood that it is really beneficial for the manufacturing industries that produces goods in batches have to select Cellular Manufacturing Systems as one of their strategical methods for formation of layout and handling of logistics. The methodology adopted using GA have been even if convincing to some extent, from section 4, it is very clear that before considering the tool for CMS implementation following suggestion could be adopted so that the gap between theoretical, hypothetical studies and practical real time experiments will be reduced further:

(a) Various types of GA operators may be tested in cell formation problem to find out which is best for real time application and could be adopted in trials for finding out the best one.
(b) In multi-objective formulation, Pareto optimality may be tested instead of using objective function with weighted average.
(c) Coding should be selected in such a way that there should not be any non existing solutions.
(d) Convergence property has to be properly treated to find out global optima values.

Finally, it can be stated that genetic algorithms seem to be the most widely reported optimization techniques in the given area. However, well-recognized classification of genetic algorithms is still missing; event though there are several approaches how to categorize GAs.

REFERENCES

[1] Miltenburg J. and Zhang. W. 1991. A Comparative Evaluation of Nine Well known Algorithms for Solving the Cell Formation Problem in Group Technology. Journal of Operations Management. 10(1): 44-72.

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

[3] Mcauley J. 1972. Machine grouping for efficient production. Production Engineer. 51(2): 53-57.

[4] Rajagopalan R. and Batra J.L. 1975. Design of Cellular Production System - A Graph Theoretic Approach. International Journal of Production Research. 13(6): 567-579.

[5] Boctor F.F. 1991. A Linear Formulation of the Machine-Part Cell Formation Problem. International Journal of Production Research. 29(2): 343-356.

[6] Waghodekar P.H. and Sahu S. 1984. Machine-Component Cell Formation in Group Technology. MACE. International Journal of Production Research. 22(6): 937-948.

[7] Venugopal V. 1999. Soft-Computing Based Approaches to the Group Technology Problem: A State-of-the-Art Review. International Journal of Production Research. 37(14): 3335-3357.

[8] SudhakaraPandian R. and Mahapatra S.S. 2007. Application of soft computing techniques for cell formation considering operational time and sequence. An unpublished thesis submitted at National Institute of Technology, Rourkela, India. pp. 42-50.

[9] Venugopal V. and Narendran T. T. 1992. A Genetic Algorithm Approach to the Machine Component Grouping Problem with Multiple Objectives. Computers and Industrial Engineering. 22: 469-480.

[10] Suresh N.C., Slomp J. and Kaparthi S. 1999. Sequence-dependent Clustering of Parts and Machines a Fuzzy ART Neural Network Approach. International Journal of Production Research. 37(12): 2793-2816.

[11] Zolfaghari S. and Liangm. 2003. A New Genetic Algorithm for the Machine/Part Grouping Problem Involving Processing Times and Lot Sizes. Computers and Industrial Engineering. 45: 713-731.

[12] Sudhakara Pandian R. and Mahapatra S.S. 2009. Manufacturing Cell Formation with Production Data using Neural Network. Computers and Industrial Engineering. 56: 1340-1347.

[13] Chandrasekharan M. P. and Rajagopalan R. 1989. Groupability: an analysis of the properties of binary data matrices for group technology. International Journal of Production Research. 27: 1035-1052.

[14] 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: 213-232.

[15] Kumar C. S. and Chandrasekharan M. P. 1990. Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. International Journal of Production Research. 28: 233-243.

[16] Logendran R. and Ramakrishna P. 1995. Manufacturing cell formation in the presence of lot splitting and multiple units of the same machine. International Journal of Production Research. 33: 675-693.

[17] Nagi R., Harhalakis G. and Proth J. M. 1990. Multiple routings and capacity considerations in group technology applications. International Journal of Production Research. 28: 2243-2257.

[18] Ballakur A. and Steudel H. J. 1987. A within-cell utilization based heuristic for designing cellular manufacturing systems. International Journal of Production Research. 25: 639-665.

[19] Boctor F. F. 1996. The minimum-cost, machine-part cell formation problem. International Journal of Production Research. 34: 1045-1063.

[20] Chandrasekharan M. P. and Rajagopalan R. 1986. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. International Journal of Production Research. 24: 451-464.

[21] J. Rezaeian, N. Javadian, R. Tavakkoli-Moghaddam, F. Jolai. 2010. A hybrid approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system. Applied Soft Computing Journal. DOI:10.1016/j.asoc.2011.03.013.