An adapted Genetic Algorithm to solve Generalized Cell Formation.

Emmanuelle VIN*, Pascal FRANCQ and Alain DELCHAMBRE***

**BEAMS Department, Université libre de Bruxelles (ULB), Brussels, B-1050, Belgium.*
(Tel.: +(32) 2650-2814; e-mail: Emmanuelle.vin@ulb.ac.be)
***CoDE Department, Université libre de Bruxelles (ULB), Brussels, B-1050, Belgium.*

Abstract: This paper addresses the cell formation problem with alternative process plans and machine capacity constraints. Given alternative process plans, machine capacities and quantities of parts to produce, the problem consists in defining preferential process and routing for each part (grouping of operations into machines) optimizing machines grouping into manufacturing cells. The problem can be decomposed in two distinct sub-problems: operations grouping on resources, yielding flows between the machines, and grouping of these latter into independent cells. The objective of the proposed method is to optimize both groupings (operations on machines and machines into cell) minimizing of the inter-cellular moves. To solve simultaneously both grouping interdependent problems, we propose a modified grouping genetic algorithm (SIGGA). In our adapted grouping genetic algorithm, each chromosome is composed of two parts, one part for each problem. According to different application rates, the genetic operators are applied on the first, on the second problem, or on both problems. Finally, the population chromosomes simultaneously evolve in both problems.

1. INTRODUCTION

Cellular production systems are an important application of group technology, which consists in decomposing system into sub-systems and in grouping similar things together. Cellular manufacturing systems are based on the creation and the management of several production cells. These cells are composed by complementary machines placed as close as possible and dedicated to a product family. A principal problem for the implementation of these cells is precisely the cell formation problem.

During the last years, the cell formation problem has been addressed in numerous works. Several methods have been presented and can be classified by different ways.

Initially, the cell formation problem was really simplified and the methods used to solve the creation of product families were very simple (King 1980). With time, the problem has evolved with the data complexity. The authors took progressively into account new production parameters such as sequence operation (Gupta, 1993), cost (Choobineh, 1988), alternative process plans (Gupta 1993), part volumes (Logendran and al., 1994) , machine capacity (Sofianopoulou, 1999), labor-related factor (Suresh and Slomp 2001), flexibility (Defersha and Chen, 2006), … Suresh (Suresh and Slomp, 2001) proposes a classification and a review based on the parameters used for a more complex cell formation problem.

Joines (Joines and al. 1996) presents a complete review of production oriented manufacturing cell formation techniques. They proposed a classification to group the resolution methods into different categories as:

- Part Family identification methods (Classification and coding - Relational data bases)

- Part-Machine grouping methods (Array based methods - Hierarchical clustering (similarity coefficient) - Non-hierarchical clustering - Graph theory - Mathematical programming - Artificial intelligence - Search heuristic)

Methods classified in the artificial intelligence are group into Neural Network (Mahdavi and al., 2007) Simulated Annealing method (Wu and al. 2008), Tabu Search (Logendran and al. 1994) or Genetic Algorithm.

Genetic algorithms (GAs) are in order to explore and exploite a large space search to obtain a good solution. Many authors use them to solve a cell formation problem. Venugopal and Narendran (1992) and Onwubolu and Mutingi (2001) minimize the total cell load variation with a GA. Zhao and Wu (2000) and Uddin and Shanker (2002) used GA to solve a cell formation problem with alternative routes. Goncalves and al. (2004), Mahdavi and al. (2008) presented a method based on a genetic algorithm.

As explained in the following section, the alternative process plans transform the grouping problem (machines into cells) in two grouping problems. The first one consists in allocating a machine for each operation. This machine must be able to achieve this operation (RP, Resource Planning problem). The second grouping problem tries to make independent cells in assigning a cell for each machine (CF, Cell Formation problem). This generalized problem is often treated by combining Artificial Intelligent methods with another heuristic. The resolution can be sequential, semi-simultaneous or completely simultaneous. The sequential resolution finds a solution for the second problem based on the result found for the first problem or conversely (Gupta 1993). The semi-simultaneous resolution is based on several iterations of the sequential resolution (Kasiligam and Lashkari 1991, Goncalves and Resende 2002). The simultaneous resolution permits to optimize both problems simultaneously.

Nagi et al. (1990) proposed an (semi-simultaneous) iterative method solving the two distinct sub-problems: cell formation, tackled with a heuristic and routing selection, addressed with the Simplex method. The use of the simplex limits the size of the considered problem. Caux et al. (2000) proposed an approach based on simulated annealing and a branch-and-bound algorithm in order to perform routing selection and inter-cellular moves minimization simultaneously. Sofianopoulou (1999) proposed an adapted simulated annealing-based heuristic. Vin and al. (2005) proposed a semi-simultaneous method based on a genetic algorithm to solve the selection of preferential routing. Given an operation assignment, a heuristic finds the "best" associated grouping of machines into cell.

In this paper, we solve a cell formation problem with real alternative process plans. We propose a new adapted algorithm (SIGGA, Simultaneous resolution by a Grouping Genetic Algorithm) based on a grouping genetic algorithm (GGA) simultaneously to solve both problems:

- The routing selection problem and the allocation of operations on a specific machine, yielding flows between the machines (resource planning problem with several constraints and criteria);
- The grouping of machines into independent cells (cell formation problem).

In the section 2 the problem is described with all used parameters and all used constraints. The mathematic formulation is exposed in the section 3. The proposed method based on the grouping genetic algorithm GGA and the different characteristics of the implementation are explained in section 4. A case study is presented in section 5, before concluding in section 6.

2. DESCRIPTION OF THE PROBLEM

2.1. Alternative Routes/Processes

The Suresh and al.'s review (2001) does not make any distinction between different alternative process plans. First, Kusiak (1987) and Choobineh (1988) introduced the alternative routings in working with some duplicated machines. When the number of identical machines increases, types of machine can be defined. The process plan is represented like a sequence of machine type ($Pr = \{tm_1, tm_2, tm_3\}$) where tm_i represents the machine type i. Each type is composed of several machines able to achieve a type of operation. One process plan corresponds to alternative routings (=sequence of machines) (Askin et al., 1997); (Suresh and Slomp, 2001), (Yin and Yasuda, 2002), (Vin and al., 2005).

Others authors (Kusiak, 1987); (Gupta, 1993); (Logendran et al., 1994); (Caux et al., 2000); (Adenso-Diaz, 2001) use several processes defined like a sequence of machines. ($Pr_1 = \{m_1, m_2, m_3\}$, $Pr_2 = \{m_3, m_2, m_6\}$ where the product is defined by two process).

The cell formation becomes more complex when the alternative process plans are used and when each process is defined like a sequence of the machine type ($Pr_1 = \{tm_1, tm_2, tm_3\}$, $Pr_2 = \{tm_3, tm_2, tm_6,\}$ where the product is defined by two processes). To achieve each product, we need to choose which process is the best and for each operation, the best machine to manufacture it. The cell formation is reduced in the choice of process, the choice of routing in this process, and machine grouping into cells. Authors using these real alternative process plans are not frequent (Sofianopoulou, 1999), (Uddin and Shanker, 2001).

2.2. Notations

Indices

t Machines types index (TMt=Machine type t). t=1,2,..,n_t
m Machines index (M_m=Machine m). m=1,2,..,n_m
i Products index (P_i=Product i). i=1,2,..,n_p
j Process index (Pr_{ij}=Process j of product i). j=1,2,..,npr_i
k Operations index (O_{ijk}=Operation k of process j of product i). k=1,2,..,no_{ij}
c Cells index (c_c=Cell c) c=1,2,.., nc

Parameters

d_m Availability of machine m.
Q_i Quantity of product i.
T_{ijk} Average operating time of operation O_{ijk}.
T_{ijkm} Operating time if O_{ijk} on machine m.
n_c Maximum number of cells.
nm_c Maximum number of machines in cell c.

Necessary data and hypotheses are presented hereunder. A machine type has different capabilities in terms of operation types. Each machine m, unique, is characterized by an availability parameter d_m, which is equal to its capacity value times its availability rate. Each machine belongs to at least one type and can belong to several types if it is a multi-functional machine.

Each product is defined by a set process (Process = a sequence of npr_i operations $\{o_{ij1}, o_{ij2},\ldots, o_{ij\ npri}\}$). Each operation is defined as an operation type that can be accomplished on one machine type (lathe, grinding machine, etc.). So each operation can be performed on all machines belonging to its type. The duration of each operation can be fixed for the considered machine type (average operating time, T_{ijk}), or particularized to a specific machine (operating time, T_{ijkm}). Each product has several potential routings available for a specific process.

3. FORMULATION

3. 1. Decision variables

$x_{ij} = 1$ if process j of product i is used (= 0 otherwise).
$y_{ijkm} = 1$ if operation O_{ijk} is achieved on machine m (= 0 otherwise).
$z_{mc} = 1$ if machine m is in cell c (= 0 otherwise).

When the algorithm assigns an operation O_{123} to a specific machine M_5, variable x_{12} is put at 1 to specify that process j of product i is used in the solution. This variable implies that all other variables $x_{1j\neq 2}$ of the same product (P_1) are put at 0. In this case, all operations belonging to $P_{1j\neq 2}$ cannot be used in the grouping solution. To complete this notation, decision variable y_{1235} is also equal to 1.

Decision variable z_{mc} is used to compute moves between cells as a function of the assignation of machines in each cell.

3.2 RP Constraints

$$\sum_{j=1}^{npr_j} x_{ij} = 1 \qquad \forall i \tag{1}$$

$$\sum_{i=1}^{np}\sum_{j=1}^{npr_j}\sum_{k=1}^{no_{ij}} Q_i . T_{ijkm} . y_{ijkm} \leq d_m \qquad \forall m \tag{2}$$

$$if\,(y_{ijkm} = 1) => Q_i . T_{ijkm} . > 0 \qquad \forall i, j, k, m \tag{3}$$

Constraint (1) represents the process selection. As explained above, only one process can be chosen by product. Second constraint defines the machine charge. This charge cannot exceed the machine availability. Constraint (3) determines that an operation assigned to a specific machine must have a strictly positive operating time. Indeed, if a machine cannot achieve an operation, the operating time T_{ijkm} of the operation O_{ijk} on the machine M_m will be null.

3.3. CF Constraints

$$\sum_{c=1}^{nc} z_{mc} = 1 \qquad \forall m \tag{4}$$

$$\sum_{m=1}^{nm} z_{mc} \leq n_c \qquad \forall c \tag{5}$$

Constraints (4) and (5) concern machine grouping into cells. The first one verifies that all used machines have been grouped. The second one confirms that each cell capacity is not exceeded. The maximum capacity can be different on each cell.

3.4. Cost Function

$$\phi_{mn} = \sum_{i=1}^{np}\left(\sum_{j=i}^{npr_i} x_{ij} \cdot \left(\sum_{k=1}^{no_{ij}-1}\left(y_{ijkm} \cdot y_{ijkn}\right)\cdot\left(Q_i \cdot T_{ij(k+1)n}\right)\right)\right) \tag{6}$$

$$\Phi_{\text{intracell}} = \sum_{c=1}^{nc}\left(\sum_{m=1}^{nm_c}\sum_{n=1}^{nm_c}\left(z_{mc} . z_{nc}\right) . \phi_{mn}\right) \tag{7}$$

$$\Phi_{\text{intercell}} = \Phi_{\text{Total}} - \Phi_{\text{intracell}} \tag{8}$$

$$\text{Cost function } : \; Min \frac{\Phi_{\text{intercell}}}{\Phi_{\text{Total}}} \tag{9}$$

The proposed method is a multicriteria method, but this paper is focused on one criterion: the minimization of inter-cellular moves. Equation (6) represents the move between two machines, m and n. It is computed on the basis of the sum of operating time to achieve on machine n for all products coming from machine m. This value can be computed when the first part of the chromosome is completed and the first problem is solved. To compute equation (7), the second part of the chromosome needs to be completed and a valid solution of cell assignment found. The intra-cellular moves into a cell c are the sum of moves between all machines assigned to this cell c. The total intra-cellular move is the sum of intra-cellular moves for each cell. The total moves can be different depending on process and routing choices. To compare two solutions and take into account this difference, the criterion to minimize is the relative inter-cellular moves (9). This criterion is the same as the maximization of the total intra-cellular move.

The whole problem is solved with a SIGGA whose flowchart is illustrated in Fig. 1. This algorithm is an adaptation of the Grouping Genetic Algorithm (GGA) explained in next section.

4. SIGGA

4.1. Origins

The genetic algorithms (GAs) are an optimization technique inspired by the evolution process of living organisms (Holland, 1975). The basic idea is to maintain a population of chromosomes, each chromosome being the encoding (a description or genotype) of a solution (phenotype) of the problem being solved. The worth of each chromosome is measured by its fitness, which is often simply the objective function value of the search space point defined by the (decoded) chromosome. Falkenauer (Falkenauer, 1998) pointed out the weaknesses of standard GAs when applied to grouping problems, and introduced the GGA, which is a GA heavily modified to match the structure of grouping problems. Those are the problems where the aim is to group together members of a set (i.e. find a good partition of the set). The GGA operators (crossover, mutation and inversion) are group-oriented, in order to follow the structure of grouping problems.

We (Vin and al., 2005) presented an genetic algorithm in two steps. The used algorithm is based on a semi-simultaneous method. First, a population of RP solution (operation/machine) is initialized by a heuristic. Next, for each solution, a heuristic is applied to complete the chromosome with a valid solution for the second problem (machine/cell): machine grouping into cells. This method was good but for large scale problem, the heuristic can not find the best solution. Moreover, the method was limited to cases studies with alternative routings but with only one process (sequence of machine types).

4.2. Description of the SIGGA

The SIGGA (Simultaneous resolution by a Grouping Genetic Algorithm) is presented in the Fig. 1.

This algorithm is based on a classical GGA. A population of chromosomes is initialized. Each chromosome represents a valid solution to both problems: the process selection and the assignment of each operation on a specific machine able to achieve it (Resource Planning Problem: operation/machine); the grouping of machine into independent cells (Cell Formation Problem: machine/cell).

Fig. 1. Adapted SIGGA

Both problems are interdependent because groups of the first problem are precisely the objects to group in the second problem. Our objective is to find a "good" solution for both problems (RP, Resource Planning, and CF, Cell Formation).

A RP Heuristic initialized the first part of the chromosome while a CF Heuristic initialized the second part of each chromosome. These two heuristics generate only valid solutions respecting all hard constraints defined in section 3.2. After this separated initialization, the fitness of each chromosome is completed with its evaluation. The best chromosome is saved. In order to evolve to the best solution, different genetic operators are applied after a specific selection (tournament strategy). These genetic operators are applied on the complete chromosome to make both problems evolve simultaneously. After the application of genetic operators, both parts of each chromosome are reconstructed. And a new generation is started. The algorithm stops when the maximum number of generation is reached or when the algorithm finds the solution without flow between cells.

4.3. Coding of the chromosomes

The coding of chromosome is similar to the Grouping Genetic Algorithm (GGA) presented by Falkenauer (1998). However, this encoding is doubled. More concretely, let us consider the following chromosome:

ADBCEB:ADBCE: FFGFG:FG

The chromosome encodes the solution for a double grouping problem where the first problem solution, composed by 6 objects and 5 groups, can be written as

A={0}, B={2,5}, C={3}, D={1} and E={4}.

The second problem solution, composed of the 5 objects (equivalent to the groups for the first problem) and 2 groups, is:

F={0,1,3}, G={2,4}.

The visual encoding can be written as follows:

{0}{2,5}{3}{1}{4}:{0,1,3}{2,4}.

4.4. Initialisation

Our heuristic is based on a first fit with a first random part. The different objects are treated in random order. The heuristic is decomposed in several steps:

Step 1. Verify if the object can be used in order to respect the hard constraints with the actual solution in construction (one used process by product (RP), machine must be not empty to be grouped (CF)).

Step 2. Create a new group with a variable probability equal to

$$p = \frac{MaxGroups - NbGroups}{MaxGroups} \tag{10}$$

where: MaxGroups = maximum of allowed groups for the problem, and NbGroups = actual number of used groups.

Step 3. Find the first group able to accept the object in order to respect the hard constraints about capacity or compatibility (Capacity not exceeded (RP and CF), machine able to achieve the operation (RP)). If a group is found, the object is inserted. Otherwise, a new group is created to accept the object.

4.5. Genetic operators

The important point is that the genetic operators will work with the group part of the chromosomes, the standard object part of the chromosomes serving to identify which objects actually form which group. Note in particular that this implies that operators will have to handle chromosomes of variable length with genes representing the groups.

The tournament strategy is chosen to rank the chromosomes. The chromosomes ranked in the top half will be used as parents for the crossovers and the resulting children will replace the chromosomes in the bottom half.

The circular crossover get the population evolved.

Step 1. A crossing site is randomly selected in each parent.
Step 2. Groups selected by the crossing site of one parent are inserted at the second parent crossing site.
Step 3. New injected objects have priority. So, the existing groups containing objects that are already in the inserted groups are eliminated.
Step 4. The objects left aside are reinserted into the solution. It is the reconstruction phases.

The mutation is the second operator applied to the population. The role of a mutation operator is to insert new characteristics into a population to enhance the search space of Genetic Algorithm. The idea is to randomly choose a few groups and to remove them from the solution. The objects attached to these groups are then reinserted into the solution.

The chromosomes reconstruction is based on the flow computing. For the part RP the object/operation is inserted in the group/machine belonging to the cell with maximum flow. For the part CF the object/machine is inserted in the group/cell with maximum flow. If it is not possible because of the capacity constraint, object is inserted randomly.

5. APPLICATION, CASE STUDY

The algorithm has been tested with different case studies found in the literature. In this article, we will present the four case studies used by Sofianopoulou. The advantage of these four cases is that they represent a set of the different processes presented in the section 2. The implementation of the SIMOGGA algorithm for these problems is coded in C++ and run on a Bi-Xeon 3.60Ghz Hyper Threading with 1 Go RAM.

For each problem, the solution is presented in a table where the cell composition is determined by the machines and the products assigned to each cell. When the product is written in parentheses in two cells, there are many moves in each cell. In this case, the product can be assigned independently to each cell. Furthermore, the selected process plan is defined in parentheses for each product with alternative process plans.

5.1. Problem 1

The first problem (P1) is an adaptation from Kusiak (1990) to take into account the processing sequence of each part and alternative routings (alternative machines sequences). We consider 5 products and 4 machines.

The following solution is founded for a maximum cell size nm_c=2. The index of the product defines the selected process: Cell 1 {Mach (1, 3)-Prod (2_2, 4_2, 5_2)} and Cell 2 {Mach (2, 4)-Prod (1_1, 3_2)}. Same solution has been found by Sofianopoulou and Kusiak with a number of inter-cellular moves equal to 0.

5.2. Problem 2

The second problem has the same particularity as the first one except for the size. The algorithm tries to group 20 products and 12 machines into 3 cells. The maximum cell size is equal to 5. The following solution is founded for 500 generations and the corresponding runtime was about 3 seconds.

Cell 1 {Mach (1, 4)-Prod (3, 15)};
Cell 2 {Mach (2, 6, 7, 9, 10)-Prod (2_2, 4, 5_1, 8, 10, 12_2, 14_2, 16, 19, 20 (6, 17_2))};
Cell 3 {Mach (3, 5, 8, 11, 12)-Prod (1, 7, 9, 11, 18 (13))}.

The solution contains the same number of inter-cellular moves (29) as the Sofianopoulou's solution. However, the selected process plans for each product is not the same and the machines are not allocated into the same cell. The principal difference however with Sofianopoulou's solution is the type of moves. In our algorithm, the cell formation heuristic is adapted to preferentially produce unidirectional moves. To simplify moves between cells, it is better to visit cell 1, cell 2 and cell 3 than to visit cell 1, cell 2 and to come back to cell 1. The proposed solution has 13 go-returns and 16 simple moves between cells. Sofianopoulou's solution has the opposite, 16 go-returns, and 13 simple moves.

5.3. Problem 3

Problem 3 is composed of 20 products and 14 machines (12 machine types). Two machines are duplicated. Each product is characterized by a unique process sequence. The utilization of duplicate machines implies one or several routings (machine sequence) for each product. As for problem 2, cell size is limited to 5. The algorithm was run for 500 generations in 6 seconds. The solution is presented hereunder.

Cell 1 {Mach (1, 4, 7, 10, 13)-Prod (3_2, 4, 8_2, 10_2, 14, 15_2, 20_2, (9_2, 13_2, 19_2))}
Cell 2 {Mach (2, 3, 5, 6, 11)-Prod (1_1, 2_1, 5, 11_3, 16_1, 17_1 (6_3, 9_2, 13))}
Cell 3 {Mach (8, 9, 12, 14)-Prod (7, 12_2, 18_2, (19_2))}

The same observations as for problem 2 can be done. The selected process plans are not equivalent but in this case, types of moves are equivalents.

5.4. Problem 4

Problem 4 is an application of the alternative process plans. Each product is defined by several processes (machine type sequence) and each machine type can contain one or two machines. This problem is composed by 30 products and 18 machines regrouped into 16 machine types. For this problem, the cell size is limited to 7 machines by cell. The solution is founded with 500 generations in 12 seconds. The best total number of inter-cellular moves found by the algorithm is 32, less than Sofianopoulou's solution (34).

Cell 1 {Mach (1, 2, 7, 11, 13, 14, 15)-Prod (2_1, 4_1, 9, 11_1, 12_3, 15, 25, 30 (11, 17))}
Cell 2 {Mach (5, 8, 18)-Prod (8_3, 24_2, 29_6 (5))}
Cell 3 {Mach (3, 4, 6, 9, 10, 12, 16)-Prod (3_3, 6_3, 7, 101, 13_1, 14, 16_1, 18, 19, 20, 21_1, 22, 23_1, 26_1, 27, 28 (1, 17))}

Sofianopoulou needed 100 seconds to solve these two last problems. The presented SIGGA is fast and efficient with all types of data in term of alternative process plans and alternative routings.

8. SUMMARY

In the present paper, an adapted multi objective grouping genetic algorithm has been developed to solve a double grouping problem. The algorithm SIMOGGA can solve simultaneously two grouping problems. It is particularized to the cell formation problem for manufacturing systems. We took into account the three most important production parameters in cell design:

- The process sequence allows defining the flows between machines;
- The production volume allows computing the real material moves between machines and between cells;
- The use of real alternative process plans allows optimizing the cell formation.

The algorithm is applied with one flow criterion to four case study used by Sofianopoulou. These four cases are a good variety of the different processes utilization. The algorithm is efficient with both high and low flexibility cases.

REFERENCES

Adenso-Diaz, B, Lozano, S., Racero, J., Guerrero, F., 2001. Machine cell formation in generalized group technology. *Computers and industrial engineering*, 41(2), 227-240.

Askin, R., Selim, H., Vakharia, A., 1997. A methodology for designing flexible cellular manufacturing systems. *IIE transactions*, 29(7), 599-610.

Caux, C., 2000. Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Economics*, 64, 279–284.

Choobineh, F., 1998. A Framework for the Design of Cellular Manufacturing Systems. *International Journal of Production Research*, 26(7), 1161-1172.

Falkenauer, E., 1998. Genetic algorithm and grouping problem. *Wiley & Sons*.

Goncalves J.F. and Resende M.G.C. 2004. An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*, 47, 247–273

Gupta, T., 1993. Design of manufacturing cells for flexible environment considering alternative routing. *International Journal of Production Research*, 31(6), 1259-1273.

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

Joines, J. Culbreth, C., King, R., 1996. A Comprehensive Review of Production Oriented Cell Formation Techniques. *International Journal of Factory Automation and Information Management*, 3(3-4), 225-265.

Joines J.A., Culbreth C.T. and King R.E. 1996b. Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transaction*, 28, 69-85

King. 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

Kusiak, A., 1987. The Generalised Group Technology Concept. *International Journal of Production Research*, 25(3), 561-569.

Logendran, R., Ramakrishna P., Srikandarajah, C., 1994. Tabu search-based heuristics for cellular manufaturing systems in the presence of alternative process plans. *International Journal of Production Research*, 32(2), 273-297.

Mahdavi I., Javadi B., Fallah-Alipour K. and Slomp J. 2007. Designing a new mathematical model for cellular manufacturing system based on cell utilization. *Applied Mathematics and Computation*, 190(1), 662-670

Nagi, R., Harhalakis, G., Proth, J.-M., 1990. Multiple routings and capacity considerations in group technology applications. *International Journal of Production Research*, 28(12), 2243-2257.

Onwubolu GC. and Mutingi M. 2001. A genetic algorithm approach to cellular manufacturing systems. *Computer and Industrial Engineering*, 39, 125-144

Sofianopoulou, S., 1999. Manufacturing cell design with alterantive process plans and/or replicate machines. International *Journal of Production Research*, 37(3), 707-720.

Suresh, N.C., Slomp, J., 2001. A multi-objective procedure for labour assignments and grouping in capacitated cell formation problems. *International Journal of Production Research*, 39(18), 4103-4131.

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

Vin, E., De Lit, P., Delchambre, A., 2005. A Multiple Objective Grouping Genetic Algorithm for the Cell Formation Problem with Alternative Routings. *Journal of Intelligent Manufacturing*, 16(2), 189-206.

Venugopal, V. and Naredran, T. T. 1992. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers and Industrial Engineering*, 22, 469-480.

Wu T.H., Chang C.C and Chung S.H. 2008. A simulated annealing algorithm for manufacturing cell formation problem. *Expert Systems with Applications*, 34(3), 1609-1617

Yin, Y. and Yasuda, K., 2005. Manufacturing cells' design in consideration of various production factors. *International journal of production research,* 40(4), 885-906.

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.