A Multiple Objective Grouping Genetic Algorithm for the Cell Formation Problem with Alternative Routings

E. VIN, P. DE LIT and A. DELCHAMBRE

Université libre de Bruxelles (ULB), CAD/CAM Department,50 av. F. D. Roosevelt, CP165/14, B-1050 Brussels, Belgium. E-mail:{emmanvin, pdelit, adelch}@ulb.ac.be

This paper addresses the cell formation problem with alternative part routings, considering machine capacity constraints. Given processes, machine capacities and quantities of parts to produce, the problem consists in defining the preferential routing for each part optimising the grouping of machines into manufacturing cells. The main objective is to minimise the inter-cellular traffic, while respecting machine capacity constraints. To solve this problem, the authors propose an integrated approach based on a multiple-objective grouping genetic algorithm for the preferential routing selection of each part (by solving an associated resource planning problem) and an integrated heuristic for the cell formation problem.

Keywords: Cell formation; group technology; alternative routings; grouping genetic algorithm; multiple objectives.

1. Introduction

After first trials in American companies (Flanders, 1925), group technology (GT), a theory of management based on the principle that similar things should be done similarly, has developed in the Soviet Union in the late 1940s and early 1950s (Mitrofanov, 1966). In the 1970s, Burbidge (1975) developed this systematic planning approach according to which products needing similar operations and a common set of resources are grouped into families, and resources are grouped into small-sized production subsystems. The main goal of this cellular manufacturing (CM) is to bring together the advantages of both the flow and the job shop production (mainly the flexibility of job shop and efficiency of flow shop) and to reduce the complexity of the control (Kusiak, 1987). During the last years, the cell formation problem (CFP) has been addressed in numerous works because designing CM systems is known as a difficult problem (it is actually NP-hard). Historical accounts

Survey of approaches accounts on the solving of the CFP are proposed in (Kusiak, 1988), (Mungwattana, 2000), and (Mahesh and Srinivasan, 2002).

This paper addresses the CFP with alternative part routings, taking into account machine capacity constraints. It proposes a new integrated approach based on a multiple objective grouping genetic algorithm (MOGGA) to solve the routing selection problem, and an embedded heuristic to simultaneously tackle the cell formation problem. The remainder of this paper is organised as follows: the relevant literature is presented in section 2; it is followed in section 3 by the description of the problem; the main features of the MOGGA are described in section 4; the adaptation of the MOGGA to the problem tackled is presented in section 5; two case studies are presented and analysed in section 6; conclusions are finally drawn in section 7.

2. Literature Overview

Many researchers have recognised the need to take into account different parameters in the CFP, such as the process sequence, the production volume and the alternative process.

Process sequences: In production-oriented methods, only one route (sequence of machines) is assigned to each part. Most of the time, it is not possible to show the routing of a part in an part-machine incidence matrix. This is because an entry in such a matrix only indicates whether a machine is used to process a part, and does not express the number of times a machine is needed, and in which order machines need to be visited (e.g. King (1980), Chandrasekharan and Rajagopalan (1986), Shafer and Rogers (1993), Srinivasan and Narendran (1991), Srinivasan (1994), Sathiaraj and Sarker (2002)). The consideration of process sequences complicates the cell formation problem and many CFP algorithms do not deal with this issue.

Part production volume: An important criticism against resolutions based on binary incidence matrices is that they do not take into account other information, such as production volumes, that may significantly influence cell formation. The traffic between cells is proportional to the production volumes and is necessary for instance to compute the cost of transportation. Some authors (e.g. Burbidge (1963), Burbidge (1971), DeWitte (1980), Mosier and Taube (1985)) avoid this stumbling partially by taking into account production volume, once the cells have been determined, but do not consider those volumes to form the cells.

Alternative process: By ignoring alternative processes, one may miss possible independent manufacturing cells groupings. Some researches take these alternative processes into account by considering several possible machining sequences for each part (e.g. (Kusiak, 1987); (Gupta, 1993); (Logendran *et al.*, 1994); (Lee *et al.*, 1997); (Vivekanand and Narendran, 1998); (Caux *et al.*, 2000); (Mahesh and Srinivasan, 2002)). The number of machines used in the sequence can vary and each part has one or more process plans. The problem can be summarised on the search for the good routing among the ones proposed. Other studies take these alternative processes into account by defining the process as a sequence of machine types (e.g. (Askin *et al.*, 1997); (Diallo *et al.*, 1993); (Baykasoglu and Gindy, 2000); (Yin and Yasuda, 2002)). Each machine type is capable of doing a specific operation. In this case, the problem is to allot each operation to a machine among those belonging to the required type. In such a case, the algorithm has more choices to set the preferred routing.

The cell formation problem consists in two fundamental tasks, namely, *machine-cell* formation and *part-family* formation, which can be treated sequentially or simultaneously. With the introduction of the alternative processes, the part-family formation consists to select the preferential routing for each part. There are three main strategies for forming part-machine cells:

- Formation of part families (selection of preferential routing) first, then determination of machine cells.
- Formation of machine cells first, then determination of part families (selection of preferential routing).
- Simultaneous formation of machine cells and selection of preferential routings.

Many authors use the first strategy. Gupta (1993) proposed a two-step algorithm to solve this problem. One routing is definitely determined for each part, respecting machine capacity constraints. Next, cell formation is achieved. The drawback of this method is its sequential approach. Routing selection is performed once and the flexibility given by alternative routings is not used to minimise inter-cellular traffic. Nagi *et al.* (1990) proposed an 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) used the third strategy and proposed an approach based on simulated annealing and a branch-and-bound algorithm in order to perform routing selection and inter-cellular traffic minimisation simultaneously. Each part has several possible process plans in which each operation is performed on a given machine.

Most of authors develop a uni-criterion method to solve the cell formation problem. The main objective is to create cells by minimising the traffic between cells or the quantity of products transferred between cells. The ideal case is the one where all products are manufactured in an independent cell. In this case, there is no traffic between cells. In practice the cell formation is a more complex problem are which does not depend on a simple criterion as the minimisation of the traffic between cells of machines.

The cell formation problem to be optimised depends on multiple objectives to reach (such as cost, flows between machines, flexibility, etc.). These criteria are contradictory and cannot be optimised at the same time. Increasing the performance of a solution for one objective usually leads to a decrease in the performance for the others. In the cell formation case, we can duplicate several machines to minimise the traffic and to create cells independent completely. But, this increasing of the number of machines generates an increasing of the material cost. These duplicated machines can also have an utilisation of some machines lower than the profit-earning threshold. Thus, the new solution will be perfect for the flow criterion (minimisation of the flows between cells) but this solution will be worse for the other criteria such as the cost.

The difficulty in the multi-criteria problem is to find a consistent cost function representing a single measure of quality for a solution. A value for each criterion to be optimised can be computed and the difficulty is then to choose a solution, which "is good for each criterion". In fact, a solution that optimises one criterion could be worst possible for the others as explained here under. Moreover, each criterion may have a particular importance, expressed by weight. These types of problems are known as *multi-criteria decision problems*.

In this paper, the authors present a method to solve the cell formation problem considering the three main data sets; process sequence to evaluate the flows between cells, part production volumes to take into account the capacity constraints and alternative routings to improve the cell formation. Several routings are available for each part with a unique defined manufacturing process (sequence of machines types). The third strategy is used to improve simultaneously the formation of machine cells and the determination of the part families (i.e. the selection of the preferential routing).

The problem can be summarised as the search for solutions respecting machine capacity constraints and compromising among the conflicting objective. The authors solve this multi-criteria cell formation problem with a Multiple Objective Grouping Genetic Algorithm for the task allocation to machines, using an integrated heuristic for the cell formation.

3. Problem Description

3.1. Notation

k,l	Product index.
p	Number of products.
Q_k	Demand of product *k*.
BS_k	Batch size of product *k*.
i,j	Operation index.
NO_k	Number of operation in the process of product *k*.
n	Machine index.
m	Number of machines.
tm	Number of machine types.
OM_n	Number of operation assigned to machine *n*.
t_{kin}	Processing time for operation *i* of product *k* on machine *n*.

HL_n	Higher utilisation limit of machine n.
LL_n	Lower utilisation limit of machine n.
d_n	Availability of machine n.
SP_{kl}	Similarity between two products k and l.
S_{kl}	Modified similarity between two products k and l.
MM_k	Proportion of different machines needed to achieve the process of product k (number of machines / number of operations).

3.2. Formulation

The problem can be decomposed in two distinct sub-problems: the grouping of operations on resources, yielding flows between the machines, and the grouping of these into cells to minimise the inter-cellular traffic.

The necessary data and the hypotheses are presented hereunder. Consider a set M = $\{m_1, m_2, \ldots, m_m\}$ of m machines in a given manufacturing system. Each machine n, unique, is characterised by an availability parameter d_n, which is equal to its capacity value times her availability rate. This latter value takes the possible failures into account. The authors also define a set P = $\{p_1, p_2, \ldots, p_p\}$ of p products. One and only one process (a sequence of NO_k operations $\{o_{k1}, o_{k2}, \ldots, o_{k\,NOk}\}$) is defined for each product k.

Each operation is not performed on one given machine, but 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 authors defines a set $T = \{tm_1, tm_2, \ldots tm_t\}$ of tm machine types with different capabilities in terms of operation types. Each machine belongs to at least one type and can belong to several types if it is a multi-functional machine. With this hypothesis, a product has several potential routings available for a specific process. This concept is illustrated in Fig 1. Four operations, and thus four machine types define a product. As can be seen, a given machine m_l may belong to several types (for instance m_1 belongs to tm_1 and tm_3. An example of preferential routing is $\{m_1, m_3, m_1, m_7\}$.

The duration of each operation can be fixed for the considered machine type, or particularised to a specific machine. The algorithm performs the cell formation simultaneously with the choice of preferential routing

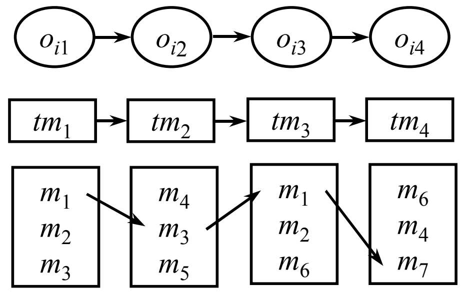


Fig. 1. One process corresponds to several potential routings.

To orient the grouping, the authors used a similarity coefficient between products k and l (SP_{kl}), computed according to Irani's method (Irani and Huang, 2000), (the method is briefly explained in section 5.5).

To circumvent the difficulty of data collection, costs are taken into account through a lower utilisation limit (LL_n) for each machine n. It is a fraction of the machine availability d_n. The limit will be set near 100 % if the machine is expensive and its use mandatory. On the other hand, this limit will be lower than 50 % if the machine is cheap, if it must not be highly loaded to be profit-earning and if a new similar machine could be introduced in the system (this helps creating independent cells).

A higher utilisation limit (HL_n) is also defined for each machine. It is used to impose some flexibility to the system: if there is a failure on a machine, the production can be reoriented to non-fully loaded resources. If the user wants a high flexibility, he will fix HL_n at a relatively low value (70 % for instance). As illustrated in Fig. 2 (U is the actual machine utilisation), these two limits are not considered as hard constraints, but are rather used in the evaluation of a proposed solution. If the user does not want to work with these two limits, the default value will be used (HL_n = 100 %, LL_n = 0 %).

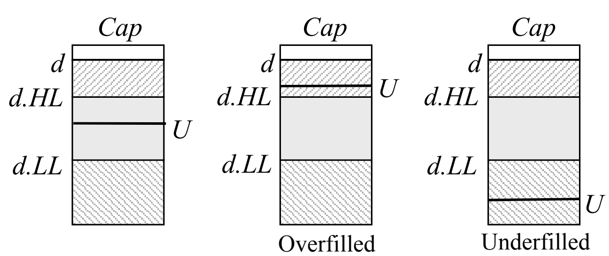


Fig. 2. Three cases of machine utilisation (normal, overfilled, under filled).

4. Multiple Objective Grouping Genetic Algorithm

The genetic algorithms (GAs) are an optimisation technique inspired by the process of evolution 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 value of the objective function of the point of the search space defined by the (decoded) chromosome.

They have been proved a successful optimisation method for three main reasons (Goldberg, 1989):

- their flexibility to conjugate themselves with specific heuristics adapted to the given problem;
- the power of their genetic operations on the chromosomes to perform a global search rather than a local one in the solution space;
- their ability to be adapted to many kind of constraints, linear or not, and any kind of cost functions, continuous, discrete, single criterion or multiple objectives.

Falkenauer pointed out the weaknesses of standard GAs when applied to grouping problems, and introduced the grouping genetic algorithm (GGA) (Falkenauer, 1998), 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. A description of the GGA encoding and operators is available in (Falkenauer, 1998).

Applying GAs to multi-objective problems addresses two difficult topics: searching large, complex solution spaces and deciding among multiple objectives. The selection of a solution from a set of possible ones on the basis of several criteria can be considered itself as a delicate and intriguing problem. A discussion on this topic is available in (Rekiek *et al.*, 2001). The proposed solution selection approach is based on a merge of a search and multi-criteria decisions as illustrated in Fig. 3, and gave birth to MOGGA. The MOGGA has been previously presented in (Rekiek *et al*., 2001). The authors here recall its main features. To come out of the multiple objectives problem stated by the cost function, the authors used the multi-criteria decision-aid method called PROMETHEE II (Brans and Mareschal, 1994). The complete description of this method is out of the scope of this paper. It is however important to know that it computes a net flow ϕ which is a *kind of fitness* for each solution. This “fitness” yields a ranking, called the PROMETHEE II complete ranking, between the different solutions in the population. The relative importance of the different objectives are set thanks to weights associated to each criterion. An essential feature of these weights is that their influence is independent from the evaluation of each criterion. Indeed, the outranking or outranked character of each solution is computed thanks to a mapping of the criteria evaluations onto [0,1], through preference functions used to establish in-two comparisons between alternatives for each criterion. So for a given set of preference functions, weights set to (0.5, 0.5) for a problem with two criteria mean that both criteria are given the same importance, independently from the exact nature of their underlying evaluations.

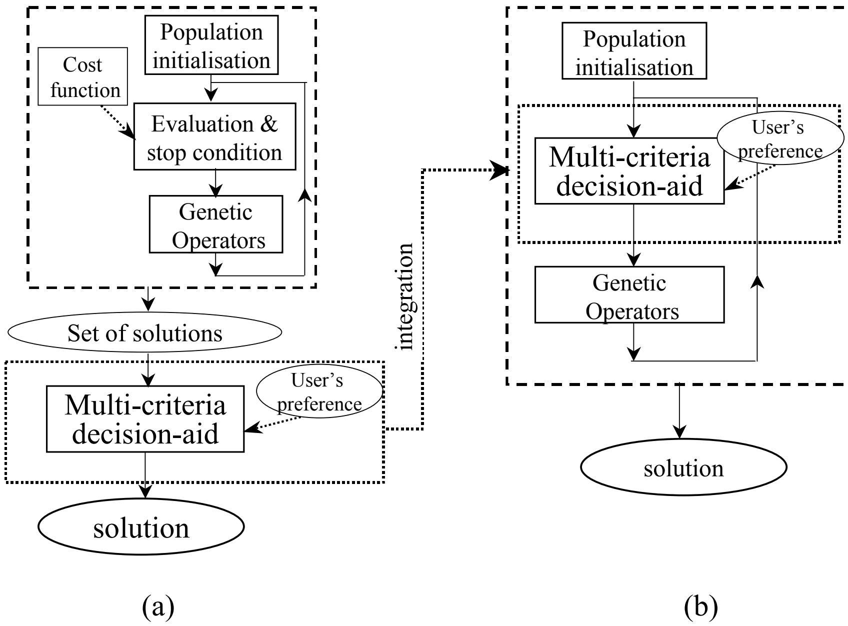


Fig. 3. Solution selection for multiple objectives: (a) classical GA; (b) proposed selection approach integrating search and decision making.

Thus, the solutions are not compared according to a cost function yielding an absolute fitness of the individuals as in a classical GA, but are compared with each other thanks to flows, depending on the current population.

5. Iterative Solving Approach – Cell Formation MOGGA

5.1. Overview

The cell formation problem is decomposed in two distinct sub-problems as explained in section 3:

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

There are two different ways to address these two sub-problems. The first consists in allocating the operations on a machine and in embedding a module to group the machines into cells. The second consists in grouping the machines into cells and then in embedding a module to allocate operations on these machines.

The proposed algorithm is based on the first solution. It allocates all generic operations on specific machines, which yields flows between them, and then searches for a grouping of these machines to minimise the inter-cellular traffic.

The whole problem is solved with a cell formation MOGGA (CF-MOGGA) which flowchart is illustrated in Fig 4. The MOGGA generates different acceptable operation allocations on machines, and a cell formation heuristic is integrated in this MOGGA to propose a good grouping of machines into cells. This approach allows to simultaneously tackle the two distinct sub-problems *at each CF-MOGGA generation*.

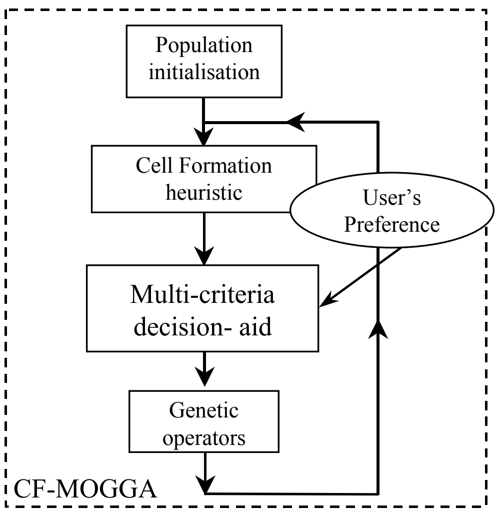


Fig. 4. CF-MOGGA: adapted MOGGA with integrated cell formation heuristic.

The best solution evolves on the basis of several criteria used by the embedded multi-criteria decision-aid algorithm (see section 5.5). The CF-MOGGA algorithm is schematically the following:

Generate an initial population using a resource planning (RP) heuristic

Apply the cell formation heuristic to each individual

Order individuals using PROMETHEE II

repeat

  Select parents;

  Recombine best parents from the population;

  Replace worst individuals of the population by the children;

  Mutate several parents from the population;

  Apply the cell formation heuristic to each individual;

  Use PROMETHEE II to order the new population;

until a maximum number of generation without improvement is reached.

After the initialisation, the best solutions are selected to be parents for the next generation of the CF-MOGGA. The worst solutions are deleted. Genetic operators are applied, yielding a new population of chromosomes. These genetic operators are the ones presented by Falkenauer (Falkenauer, 1998) for the grouping genetic algorithm. The population is evaluated after the application of the cell formation heuristic, and a new generation of the algorithm is started.

The exploration of the search space is terminated when the stop condition is reached. In this case, it is a maximum number of generations specified by the user. Generally the best solution is no more improved after hundred generations.

5.2. Encoding and Genetic Operators

The solution encoding and the genetic operators follow the classical pattern of a grouping genetic algorithm. Like for the classical GAs, a population of chromosomes is maintained. Each chromosome is the encoding of a possible solution and represents the groups of operations on machines. The population is initialised with a specific resource planning heuristic respecting two hard constraints (which cannot be violated): first, every operation is assigned to an accessible machine of the type required by the operation; second, the availability of a machine cannot be overstepped.

5.3. RP Heuristic

The initialisation RP heuristic is an adaptation of a classical first fit (FF) heuristic. The set of operations is randomised. The operations are taken in the order presented, the machines are ordered by decreasing utilisation, and each operation is allocated into the first machine capable of accommodating it among the used resources. If it is impossible to respect the hard constraints, a new machine of the required type is introduced in the system.

The heuristic used to reconstruct individuals in the population is slightly different: it follows a first fit descending scheme. The operations are sorted and allocated to the resources by decreasing duration order.

5.4. Cell Formation Heuristic

As mentioned previously, this heuristic is run before the evaluation of the chromosomes generated by the CF-MOGGA.

The allocation of operations on machines generates traffics between machines. A flow matrix between machines is then computed. This matrix is not a binary representation of the flows: all flows between machines are expressed in terms of number of destination work hours. This representation of flows takes into account the storage in front of each machine. Each element of the matrix is computed as follows:

$$x_{n1-n2} = \sum_{k=1}^{NP} \sum_{i=1}^{NO_k - 1} \delta_{k\,i\,n1} \delta_{k(i+1)n2} Q_k t_{k(i+1)n2} \tag{1}$$

where Q_k is the demand of product k (number of items per period), t_{kin} is the operating time of operation i of product k on machine n, and $\delta_{kin1}\delta_{k(i+1)n2}$ is equal to one if the operation i of product k is achieved on machine n_1 and the next operation i+1 is performed on machine n_2.

The computation of the flow matrix is illustrated on the following example. Suppose that product P_1 has a process with four operations and the chosen routing is (m_1-m_3-m_6-m_7). In this case, one has the three following flows:

$x_{\text{m1-m3}} = t_{123}\ Q_1$;

$x_{\text{ m3-m6}} = t_{136}\ Q_1$;

$x_{\text{ m6-m7}} = t_{147}\ Q_{1.}$

For example, the first flow ($x_{m1-\ m3}$) represents the operating time on machine m_3 of the second operation of the product p_1. By multiplying the term t_{123} by the quantity of product 1 to produce, Q_1, one obtains the number of work hours necessary to achieve the second operation.

These flows are computed for each process and all flows from machine n_1 to machine n_2 are summed to form the x_{n1-n2} coefficient.

The flow matrix is computed for each chromosome. The cell formation heuristic, based on Harhalakis's heuristic (Harhalakis *et al.*, 1990) is then applied, providing a grouping of machines into cells. This heuristic can be schematised as follows:

Allocate each machine to an independent cell.

Repeat

> Search the maximum flow ($x_{\text{mi-mj}} + x_{\text{mj-mi}}$) between two cells in the matrix;
>
> Verify if the two cells can be grouped together in respecting the size constraints.
>
> **If** the grouping is allowed:
>
> > Group the cells;
> >
> > Update the number of cells;
> >
> > Update the sizes of the cells;
> >
> > Convert the inter-cellular flow to intra-cellular flow;
> >
> > Update the flow matrix;
>
> **If** the grouping is not allowed:

Search the next maximum flow between two cells in the matrix.

until all allowed groupings have been performed.

5.5. Cost Function

The worth of each chromosome is measured by its fitness, which is the flow ϕ of PROMETHEE II. Five criteria are taken into consideration in the comparison of solutions: similarity between products (*RS*), use of multi-functional machines (*RM*), flexibility (*RF*), cost (*RC*), and cell formation coefficient (*RG*). The principal criterion to evaluate the chromosomes of the population is the quality of formed cells provided by the cell formation heuristic. This is an originality of the proposed approach: one of the outputs of the heuristic (the cells) is used to evaluate the quality of the solutions proposed by the CF-MOGGA.

The weight of each criterion is defined by the user. In the different case studies, the authors chose to give more importance to the cell formation criterion than the others. They also estimated the similarity between operations assigned to each machine to be important because it will allow to reduce the setup time between each operation. Finally, the authors gave more importance to the cost criterion than to the flexibility criterion.

The different criteria are explained hereunder.

5.5.1. Maximise the Similarity Coefficient (*RS*)

The coefficients SP_{kl} (presented in the section 3) are computed during the pre-treatment using the Irani's method (Irani and Huang, 2000). They propose a merger coefficient, a comparison of strings representing the manufacturing sequences to detect common substrings. Their measure is based on the longest common substring (the longest string of consecutive operations) that appears in two original sequences. Consider for instance the sequences $S_1 = (abcd\boldsymbol{b}e)$, $S_2 = (abcd\boldsymbol{f}e)$ and $S_3 = (abcde)$. The two solutions S_1 and S_2 have one operation (respectively *b* and *f*) in their sequence preventing the complete similarity with S_3, and the longest common substring is *abcd*. But S_1 is more similar to S_3 than S_2 because the supplementary operation is operation already in the sequence. The merger coefficient is adapted to take into account this difference, yielding a coefficient SP_{kl}.

The similarity factor *RS* is then computed as the sum of RS_n coefficients for each machine:

$$RS = \frac{1}{m}\sum_{n=1}^{m} RS_n$$

with

$$RS_n = \frac{\sum_{l=k+1}^{p}\sum_{k=1}^{p}\left(\sum_{j=1}^{NO_l}\sum_{i=1}^{NO_k} \mathrm{SP}_{\mathrm{kl}} \cdot \delta_{kin} \cdot \delta_{ljn}\right) + \sum_{k=1}^{p}\left(\sum_{j=i+1}^{NO_k}\sum_{i=1}^{NO_k} \mathrm{SP}_{\mathrm{kk}} \cdot \delta_{kin} \cdot \delta_{kjn}\right)}{\frac{OM_n}{2}\left(OM_n - 1\right)},$$

where δ_{kin} is equal to 1 if operation i of the part k, O_{ki}, is assigned to machine n, OM_n is the number of operations assigned to machine n, NO_k is the number of operations in the process of product k, m is the number of machines used in the studied configuration.

The computation of this coefficient RS_n is illustrated in Fig. 5 with machine 3 performing four operations (O_{13}, O_{25}, O_{42}, O_{11}). We must compare the similarity between parts P_1, P_2 and P_4. Supposing that one has the following similarity coefficients: $SP_{12} = 0.68$, $SP_{14} = 0.83$, $SP_{24} = 0.23$, the S_{kl} coefficients, with $q = 0.4$ and $p = 0.8$ are: $S_{12} = 0.68$, $S_{14} = 1$, $S_{24} = 0$.

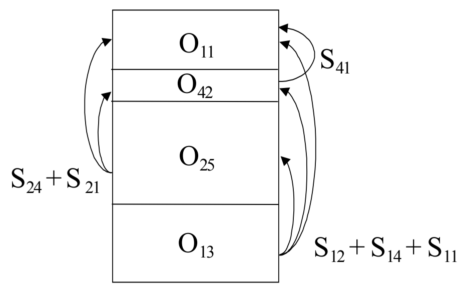


Fig. 5. Example of similarity utilisation.

The similarity coefficient RS_3 for machine M_3 is the sum of six coefficients S_{kl} between products resulting from four operations and compared two by two:

O_{13} and O_{25} => P_1 and P_2;

O_{13} and O_{42} => P_1 and P_4;

O_{13} and O_{11} => P_1 and P_1;

O_{25} and O_{42} => P_2 and P_4;

O_{25} and O_{11} => P_2 and P_1;

O_{42} and O_{11} => P_4 and P_1.

The similarity between two operations belonging to the process of the same product is equal to 1. When a similarity coefficient is not defined between two products, the default coefficient is 0 and the two products are considered as strongly dissimilar. So one has

$$RS_3 = (0.68 + 1 + 1 + 0 + 0.68 + 1)/(3 + 2 + 1) = 0.727.$$

This coefficient represents the average of the similarity between all products assigned on machine M_3.

5.5.2. Minimise the Multi-Functional Machine Coefficient (*RM*)

In the cell formation problem, it is important to use a minimum number of machines to achieve all operations belonging to the process of a part. To reduce this number, the multi-functional machines can be used. The more the number of operations of this process on these machines, the less the traffic between machines.

The multi-functional coefficient is introduced for this reason. The parameter MM_k is defined as the sum of different machines used in the chosen routing for product k, divided by NO_k (the number of operations in the process of product k):

$$MM_k = \sum_{n=1}^{m} \left(\prod_{i=1}^{NO_k} \delta_{kin} \right) \Big/ NO_k$$

where $\prod_{i=1}^{NO_k} \delta_{kin}$ is equal to 1 if one or more operations of the product k are performed on machine n. The coefficient MM_k represents the proportion of different machines needed to achieve the process of product k.

The multi-functional coefficient RM is the average of the MM_k coefficients on all products:

$$RM = \left(\sum_{k=1}^{p} MM_k \right) \Big/ p$$

This coefficient represents the average proportion of different machines used to achieve the process of a product and is illustrated in Fig. 6 and Fig. 7. In these examples, one can see different routings generated for a unique process. If one considers this criterion alone, the algorithm will prefer the solution of Fig. 7 to the one of Fig. 6, because it tries to minimise the MM_k coefficients.

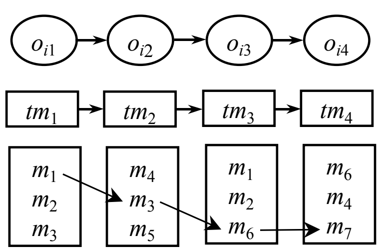


Fig. 6. Choice of a bad routing for the product i based on the multi-functional criterion ($MM_i = 4/4$).

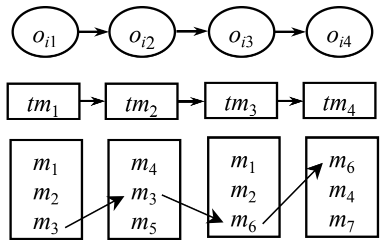


Fig. 7. Modification of the chosen routing for the product i to improve the multi-functional criterion ($MM_i = 2/4$).

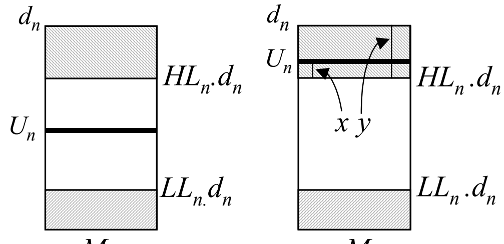
This coefficient takes into account the frequency of used machines but not the sequence of these machines. The authors considered that the cell formation coefficient (presented below) takes this aspect into account in minimising the traffic between cells. So the algorithm will prefer routing ($m_3 - m_3 - m_6 - m_6$) to routing ($m_2 - m_4 - m_2 - m_4$), because the first solution generates one flow between two machines while the second generates three traffics between two machines.

5.5.3. Minimise the Flexibility Coefficient (*RF*)

A pursued objective is to respect the workshop target flexibility. In section 3, the authors introduced the higher machine utilisation limit (HL), expressing the wish to leave a portion of the machine availability free. These limits are not hard constraints: a penalty for the non-respect of these conditions is rather introduced. An overstepping of high limit ($HL_n d_n$) fixed for each machine is penalised, and the RF coefficient is computed as:

$$RF = \frac{1}{m}\sum_{n=1}^{m}\max\left\{0, \frac{(U_n - HL_n \cdot d_n)}{(d_n - HL_n \cdot d_n)}\right\}.$$

Each penalty is computed as the relative overstepping of the limit; it is illustrated on the following example (Fig. 8). In the first case (Fig. 8a), the utilisation of the machine is below the higher limit ($HL_n d_n$), the evaluation of the penalty coefficient is negative and the effective value considered in the sum is null. In the other case (Fig. 8b)), the evaluation of the penalty is positive and equal to x/y.



(a) (b)

Fig. 8. Two cases of machine utilisation: (a) no penalty; (b) relative penalty equal to *x*/*y*.

In the flexibility coefficient *RF*, the sum of all penalties is divided by the number of machines used. This allows to take into account the variability of the number of machines between the different solutions. The algorithm tries to minimise the *RF* parameter to limit the penalties due to the non-respect of what can be somehow seen as a required flexibility.

5.5.4. Minimise the Cost Coefficient (*RC*)

As mentioned in section 3, the authors used the lower limit to somehow reflect the cost of the production system. If a machine is underfilled, it will not be profitable, and a penalty is introduced on each machine whose utilisation (U_n) is lower than the lower limit ($LL_n\, d_n$). The coefficient *RC* introduced here expresses the average of all penalties produced by the non-respect of the lower limit.

$$RC = \frac{1}{m}\sum_{n=1}^{m} \max\left\{0, \frac{(LL_n \cdot d_n - U_n)}{LL_n \cdot d_n}\right\}.$$

Fig. 9a shows a good utilisation of the machine M_n. This situation does not lead to a penalty because the evaluation of the expression is negative. On the other hand, the situation of Fig. 9b leads to the penalty *x*/*y*.

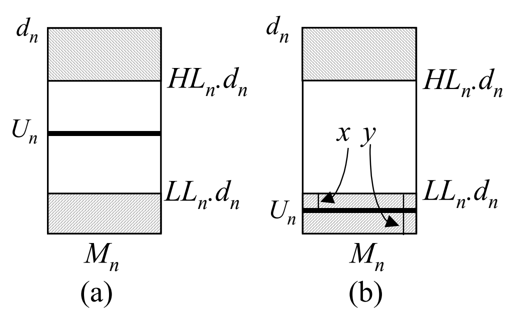


Fig. 9. Case (a) represents a machine utilisation generating no penalty. In the case (b), machine is under-filled and generate a relative penalty equal to *x*/*y*.

The penalty for this cost coefficient is a relative measure taking into account the relative importance of non-respect of lower limit imposed by the user.

5.5.5. Maximise the Cell Formation Coefficient (*RG*)

This coefficient is the most important in the solution evaluation. Many authors consider this sole coefficient when the grouping is based on one criterion. It is computed after the application of a cell formation heuristic explained in section 5.4 based on Harhalakis' algorithm (Harhalakis *et al.*, 1990). When the flow matrix is computed for a considered allocation of operations, the total sum of the flows in this matrix is equal to the total flow:

$$\Phi_{total} = \sum_{n_1=1}^{n} \sum_{n_2=1}^{n} x_{n_1-n_2}$$

During the heuristic, each grouping of cells is followed by the transformation of the inter-cellular flow into an intra-cellular flow. To pursue the grouping, it is necessary to update the flow matrix. For each grouping of cells *i* and *j*, the algorithm sums the two lines *i* and *j* and the two columns *i* and *j*. After this update, the grouping can continue with the search for the maximum flow value in the updated matrix.

Once cells have been formed, the intra-cellular flow (Φ_{intra}) is computed

$$\Phi_{intra} = \sum_{c=1}^{nbCell} \left(\sum_{n_1 \in C_c} \sum_{n_2 \in C_c} x_{n_1-n_2} \right) = \sum_{c=1}^{nbCell} (T_{cc})$$

where

$$T_{cd} = \sum_{n_1 \in C_c} \sum_{n_2 \in C_d} x_{n_1-n_2} \ .$$

T_{cd} represents the total flow between two cells *c* and *d*. So the factor T_{cc} expresses the total flow between all machines belonging to cell *c*.

To obtain the cell formation coefficient, Φ_{intra} is divided by the total flow between machines (Φ_{tot}).

$$RG = \Phi_{intra} / \Phi_{tot}$$

This criterion represents the proportion of the flow that occurs inside cells.

6. Case Studies

The proposed CF-MOGGA has been implemented on a Windows workstation in the C++ programming language, and tested on several case studies. Two of them are presented.

6.1. First Case Study (Vivekanand and Narendran, 1998)

This case study considers 12 parts and 6 machines. Some data about machine type has been completed to get all information needed by the algorithm. The data used to test the method is presented in Tab. 1. In this table, each operation *j* (O_{ij}) of the product *i* is characterised by a machine type *T* and one or more operating times on a specific machine. The availability of machines is expressed in minutes/week: m_1: 4200; m_2: 4260; m_3: 4980; m_4: 5400; m_5: 4620; m_6: 5340. The lower limit has been set to 40 % for each machines and the higher limit to 90 %. Each part type requires up to four operations, with up to three alternative machines to perform each of them. The demand of each part is: p_1: 110; p_2: 120; p_3: 80; p_4: 150; p_5: 50; p_6: 60; p_7: 100; p_8: 50; p_9: 50; p_{10}: 90; p_{11}: 50; p_{12}: 90 units/week. The authors did not introduce any specific value for the similarity between products. The algorithm uses the default value (0) for the similarity between two different products and the default value (1.0) for two operations belonging to the same process of a product. The weights associated to the different criteria are: *RS*: 3, *RM*: 1, *RF*:1, *RC*: 2, *RG*: 8.

Op	T	Op. Time	Op. Time	Op. Time	Op	T	Op. Time	Op. Time	Op. Time
$O_{1\,1}$	1	m_1:7.68	m_3:6.72		$O_{7\,1}$	3	m_4:7.80	m_3:6.36	
$O_{1\,2}$	2	m_2:7.80	m_1:6.78		$O_{7\,2}$	2	m_3:6.84	m_1:6.36	
$O_{1\,3}$	1	m_3:6.72			$O_{7\,3}$	1	m_2:6.90	m_3:6.66	
$O_{1\,4}$	3	m_4:6.42	m_5:6.42		$O_{8\,1}$	1	m_3:6.00	m_1:6.36	
$O_{2\,1}$	3	m_4:7.74	m_5:6.48		$O_{8\,2}$	3	m_4:6.78	m_1:7.44	
$O_{2\,2}$	1	m_1:7.14	m_4:6.24		$O_{8\,3}$	1	m_5:6.00	m_1:7.26	
$O_{2\,3}$	3	m_5:6.24			$O_{8\,4}$	2	m_3:6.96	m_2:7.50	
$O_{3\,1}$	3	m_4:7.62	m_5:6.72		$O_{9\,1}$	2	m_3:6.06	m_2:7.14	
$O_{3\,2}$	1	m_4:7.44	m_3:7.80		$O_{9\,2}$	3	m_4:6.78	m_3:7.26	
$O_{4\,1}$	2	m_6:7.14	m_2:7.02		$O_{9\,3}$	1	m_3:6.12	m_1:6.18	
$O_{4\,2}$	3	m_4:6.90	m_2:6.06		$O_{10\,1}$	3	m_5:6.72	m_4:6.18	
$O_{4\,3}$	2	m_6:6.54			$O_{10\,2}$	2	m_5:7.26	m_2:6.06	
$O_{4\,4}$	2	m_6:6.00	m_2:6.78	m_1:7.62	$O_{10\,:3}$	2	m_2:6.90	m_5:6.54	
$O_{5\,1}$	2	m_3:7.20	m_2:7.32		$O_{10\,4}$	3	m_5:6.54		
$O_{5\,2}$	1	m_3:7.38	m_1:7.44		$O_{11\,1}$	1	m_3:7.20	m_1:7.44	
$O_{5\,3}$	3	m_1:7.20	m_2:6.66		$O_{11\,2}$	1	m_3:7.02	m_1:7.14	
$O_{5\,4}$	1	m_1:7.44	m_3:7.68		$O_{11\,3}$	1	m_1:7.80	m_3:7.74	m_5:7.44
$O_{6\,1}$	2	m_6:6.96	m_2:6.90		$O_{12\,1}$	3	m_3:6.54	m_4:6.60	m_5:6.60
$O_{6\,2}$	2	m_6:7.62	m_2:7.02		$O_{12\,2}$	3	m_4:7.62	m_5:6.48	
$O_{6\,3}$	2	m_3:6.42	m_1:6.90		$O_{12\,3}$	3	m_5:7.62	m_4:7.74	
$O_{6\,4}$	2	m_3:6.48	m_2:6.96	m_6:6.30					

Tab 1. Operations (*ij*: product *i*, operation *j*) and operating times on specific machines (first case study).

The results of the CF-MOGGA are presented in Tab. 2. Each line *i* of this table represents all operations assigned to this machine *i* and the utilisation of this machine. The machine is allocated in the cell mentioned.

Machine	Operations	Utilisation (minutes/week)	Cell
1	$O_{4\,1}$; $O_{4\,2}$; $O_{8\,3}$; $O_{9\,3}$; $O_{10\,4}$	3294.6	2
2	$O_{5\,1}$; $O_{6\,1}$; $O_{6\,3}$; $O_{7\,2}$; $O_{8\,4}$; $O_{9\,1}$; $O_{10\,3}$; $O_{10\,2}$	3738.0	1
3	$O_{1\,1}$; $O_{1\,2}$; $O_{1\,3}$; $O_{2\,1}$; $O_{2\,2}$; $O_{2\,3}$	4685.1	3
4	$O_{1\,4}$; $O_{3\,2}$; $O_{5\,3}$; $O_{7\,1}$; $O_{7\,3}$; $O_{8\,1}$; $O_{12\,1}$; $O_{12\,2}$; $O_{12\,3}$	5391.3	3
5	$O_{3\,1}$; $O_{5\,2}$; $O_{5\,4}$; $O_{6\,2}$; $O_{6\,4}$; $O_{8\,2}$; $O_{9\,2}$; $O_{10\,1}$; $O_{11\,1}$; $O_{11\,2}$; $O_{11\,3}$	4523.4	1
6	$O_{4\,3}$; $O_{4\,4}$;	1881.0	2

Tab. 2. Results of the cell formation procedure (first case study).

One obtains the following values for the different criteria:

- *RS* = 29.55% (average similarity of the operations allocated to a machine);
- *RM* = 63.89% (average proportion of different machines that are used to complete the manufacturing product [number of different machines/number of operations]);
- *RF* = 2.182 (higher limit overstepping penalty);
- *RC* = 0.119 (penalty for under-filled machines);

- RG = 71.8% (71.8% of the total traffic [16530.6 hours of work] is an intra-cellular traffic; the inter-cellular traffic is reduced to 4648.8 hours and the intra-cellular traffic is equal to 11881.8 hours).

6.2. Second Case Study (Askin, 1997)

This case study considers 19 parts and 10 machine types. The availability of a machine type is expressed in minutes/year and the maximum number of machines of each type is presented in Tab. 3. These availabilities are based on a total capacity: 220 days per year and 8 hours of functioning a day.

Machine Type	Availability (minutes/year)	Maximum machine	LL	HL
1	102000	2	0.63	0.9
2	108000	2	0.67	0.9
3	72000	1	0.4	0.9
4	108000	2	0.67	0.9
5	96000	2	0.6	0.9
6	96000	2	0.6	0.9
7	84000	1	0.5	0.9
8	108000	1	0.67	0.9
9	114000	2	0.7	0.9
10	96000	2	0.6	0.9

Tab. 3. Data about the machine types (availability, maximum number of machines, lower and higher utilisation limit).

The demand of each part is the following:

p_1: 900, p_2: 7700, p_3: 2000, p_4: 3000, p_5: 50, p_6: 4500, p_7: 3824, p_8: 464, p_9: 3120, p_{10}: 6496, p_{11}: 3690, p_{12}: 4140, p_{13}: 1686, p_{14}: 4135, p_{15}: 4805, p_{16}: 3928, p_{17}: 4475, p_{18}: 4452, p_{19}: 2508 units/year.

Each part type requires up to six operations defined by the machine type and the processing time . The difference with the first case study is that the processing time is defined once for all machines belonging to the type machine and not specified for each particular machine. The values are given in Tab. 4. The weights associated with the different criteria are: RS: 3, RM: 1, RF:1, RC: 2, RG: 8.

Op.	Type	Op. Time/type machine	Op.	Type	Op. Time/type machine	Op.	Type	Op. Time/type machine
O_{1-1}	8	6.70	O_{7-1}	1	4.15	O_{12-1}	8	3.11
O_{1-2}	4	4.20	O_{7-2}	9	5.02	O_{12-2}	4	3.17
O_{1-3}	9	5.20	O_{7-3}	5	3.02	O_{12-3}	7	4.03
O_{1-4}	1	3.40	O_{7-4}	10	2.51	O_{12-4}	2	3.85
O_{1-5}	2	5.90	O_{7-5}	6	2.52	O_{12-5}	6	2.53
O_{2-1}	7	3.64	O_{8-1}	7	4.36	O_{13-1}	10	4.25
O_{2-2}	1	4.20	O_{8-2}	1	5.21	O_{13-2}	2	3.93
O_{2-3}	4	3.17	O_{8-3}	4	4.03	O_{13-3}	6	2.54
O_{2-4}	5	3.03	O_{8-4}	3	7.09	O_{14-1}	10	2.51
O_{3-1}	8	3.55	O_{8-5}	5	3.17	O_{14-2}	2	3.79

O_{3-2}	7	3.75	O_{8-6}	1	3.21	O_{15-1}	10	2.51
O_{3-3}	4	3.30	O_{9-1}	1	4.22	O_{15-2}	2	3.75
O_{3-4}	5	3.05	O_{9-2}	4	3.19	O_{16-1}	8	3.36
O_{4-1}	8	3.34	O_{9-3}	3	4.98	O_{16-2}	4	3.24
O_{4-2}	4	3.24	O_{9-4}	5	3.03	O_{16-3}	2	3.99
O_{4-3}	9	5.04	O_{9-5}	1	2.22	O_{16-4}	6	2.54
O_{4-4}	1	2.28	O_{10-1}	1	4.15	O_{17-1}	8	2.97
O_{4-5}	2	3.98	O_{10-2}	4	3.13	O_{17-2}	9	3.04
O_{5-1}	8	3.00	O_{10-3}	9	5.02	O_{17-3}	2	3.77
O_{5-2}	9	5.02	O_{10-4}	5	3.02	O_{17-4}	6	2.52
O_{5-3}	1	2.17	O_{10-5}	10	2.51	O_{18-1}	10	2.51
O_{5-4}	2	3.79	O_{10-6}	6	2.52	O_{18-2}	2	3.82
O_{6-1}	9	5.02	O_{11-1}	1	4.34	O_{19-1}	10	2.52
O_{6-2}	5	3.02	O_{11-2}	9	3.05	O_{19-2}	2	3.88
O_{6-3}	10	2.51	O_{11-3}	10	2.52			
O_{6-4}	6	2.52	O_{11-4}	6	2.55			

Tab 4. Operation and processing time defined for the machine type.

6.2.1. First Solution

With the data presented above, the algorithm finds the solution presented in Tab. 5. Each machine is presented with its machine type and all operations allotted to this one. Next column represents the machine utilisation in minutes a year. Finally the last column contains the identification number of the cell in which the machine is assigned.

Mach.	Type	Operations	Utilisation (minutes/year)	Cell
1	1	O_{1-4}; O_{4-4}; O_{5-3}; O_{7-1}; O_{8-2}; O_{8-6}; O_{9-1}; O_{9-5}; O_{10-1}; O_{11-1}	92850.71	2
2	1	$O_{2-2;}$	32340.0	3
3	2	O_{1-5}; O_{15-2}; O_{17-3}; O_{18-2}; O_{19-2}; O_{5-4}; O_{12-4}; O_{13-2}; O_{14-2}	105363.29	1
4	2	O_{4-5}; O_{16-3}	27612.71	1
5	3	$O_{8-4; 9-3}$	18827.25	4
6	4	O_{1-2}; O_{12-2}; O_{16-2}; O_{2-3}; O_{3-3}; O_{4-2}; O_{8-3}; O_{9-2}; O_{10-2}	102514.67	2
7	4	-	0	-
8	5	O_{2-4}; O_{3-4}; O_{6-2}; O_{7-3}; O_{8-5}; O_{9-4}; O_{10-4}	84880.85	2
9	5	-	0	-
10	6	O_{6-4}; O_{7-5}; O_{10-6}; O_{11-4}; O_{12-5}; O_{13-3}; O_{16-4}; O_{17-4}	82766.64	1
11	6	-	0	-
12	7	O_{2-1}; O_{3-2}; O_{8-1}; O_{12-3}	54235.24	3
13	8	O_{1-1}; O_{3-1}; O_{4-1}; O_{5-1}; O_{12-1}; O_{16-1}	62664.19	2
14	9	O_{17-1}	13604.0	1
15	9	O_{1-3}; O_{4-3}; O_{5-2}; O_{6-1}; O_{7-2}; O_{17-2}; O_{10-3}; O_{11-2}	105701.85	2
16	10	O_{14-1}; O_{15-1}; O_{18-1}; O_{19-1}; O_{6-3}; O_{7-4}; O_{10-5}; O_{11-3}; O_{13-1}	93596.52	1
17	10	-	0	-

Tab. 5. Solution of the allocation of operations and grouping of machines for the case study Askin.

For this solution, the evaluation of the criteria yields:

- *RS* = 46.38 % (average similarity of the operations allocated to a machine);
- *RM* = 98.07 % (average proportion of different machines that are used to complete the manufacturing product [number of different machines/number of operations]);

- *RF* = 2.37 higher limit overstepping penalty;
- *RC* = 2.425 (penalty for under-filled machines);
- *RG* = 69.42% (69.42% of the total traffic [642 544.27 minutes of work] is an intra-cellular traffic ; the inter-cellular traffic is reduced to 195 514.6 minutes and the intra-cellular traffic is equal to 449 029.67 minutes).

Four machines are not used in this solution. This is due to the RP heuristic, which tries to minimise the number of machines used.

6.2.2. Second Solution

To compare the first solution with another solution using more machines, all availabilities are decreased of 20000 units. The algorithm proposes the solution reported in Tab 6.

Mach.	Type	Operations	Utilisation (minutes/year)	Cell
1	1	O_{6-2}; O_{7-3}; O_{11-4}; O_{12-2}; O_{12-3}; O_{12-5}; O_{16-3}	80533.3	1
2	1	O_{4-4}; O_{8-2}; O_{9-2}; O_{17-3}	44657.41	3
3	2	O_{4-5}; O_{5-3}; O_{7-2}; O_{7-4}; O_{14-1}; O_{17-1}; O_{19-1}	80719.63	2
4	2	O_{4-3}; O_{5-1}; O_{11-2}; O_{18-2}	52256.37	2
5	3	O_{2-4}; O_{9-4};	18827.35	4
6	4	O_{1-5}; O_{6-1}; O_{8-5}; O_{10-4}; O_{11-1}; O_{12-1}; O_{12-4}; O_{13-3}	78105.68	1
7	4	O_{8-6};	24408.99	3
8	5	O_{2-3}; O_{3-4}; O_{9-3}; O_{10-6}; O_{13-2}; O_{17-4}	61780.85	1
9	5	$O_{9-1;}$	23100	3
10	6	O_{1-2}; O_{15-1}; O_{17-2}; O_{3-1}; O_{4-2}; O_{7-5}; O_{10-5}	73357.15	2
11	6	O_{8-1}	9409.49	4
12	7	O_{2-2}; O_{7-1}; O_{8-3}; O_{16-1}	54235.24	3
13	8	O_{1-1}; O_{11-3}; O_{14-2}; O_{19-2}; O_{5-2}; O_{9-5}; O_{10-1}	62664.19	1
14	9	O_{1-4}; O_{15-2}; O_{16-2}; O_{16-4}; O_{13-1}	85210.38	1
15	9	O_{2-1}; O_{5-4}; O_{8-4}	34095.47	4
16	10	O_{1-3}; O_{3-2}; O_{3-3}; O_{6-3}; O_{6-4}; O_{10-2}; O_{10-3}	75256.51	2
17	10	O_{4-1}; O_{18-1}	18340.01	2

Tab 6. Solution of the allocation of operations and grouping of machines for the modified case study.

For this solution, evaluation of the criteria yields:

RS = 46.39%;

RM = 98.94%;

RF = 2.61;

RC = 3.29;

RG = 65.86%.

6.2.3. Comparison of the Two Solutions

In this first solution, the algorithm finds a good grouping with a cell formation coefficient equal to 69 %. Four machines are not used in this configuration but the grouping yields a better coefficient than the second solution, for which *RG* = 65%. Further studies will be undertaken to evaluate the effect of the minimisation of the number of machines on the reported solution.

Our multi-criteria algorithm allows to test the case studies with one criterion, the inter-cellular traffic, and then to compare the reported solution with the existing case studies in the literature. One compares this solution with a second solution taking into account the cost criterion. The difference between both solutions is weak for the inter-cellular traffic criterion (a decreasing lower than 3 %) but this difference is important for the cost criterion (a decreasing upper than 10 %). This is an example showing the important to take into account different criteria to form machine cells.

The algorithm, coded on C++, was tested on a Windows 2000 station (768 Mb Ram and 450 MHz). The time to solve a problem with 76 operations and 17 machines is 10 seconds. To solve a larger problem with 1620 operations and 56 machines, the resolution time is 75 seconds. The algorithm is really fast and allows to try different configurations for the set of data and different alternatives of the weights for all criteria.

7. Conclusions and Further Works

Group technology (GT) is the grouping of parts and/or machines in order to increase the production efficiency of part design and production. The initial step in applying GT to manufacturing is the identification of part families and/or the formation of machine groups.

In this thesis, we presented an original method to address the cell formation problem with alternative routings. We took into account the three most important production parameters in cell design as presented in section 2:

- The process sequence allows to define the flows between machines;
- The production volume allows to compute the real material moves between machines and between cells;
- The alternative routings allows to optimise the cell formation and to evaluate the flexibility into each cell.

The problem is decomposed into two sub-problems:

- The allocation of each operation on a specific machine to determine the preferential routing for each product;

- The cell formation by grouping machines while respecting the size constraints in terms of number of cells and size of cells.

Our approach (section 5) is based on a multiple-objective grouping genetic algorithm (MOGGA), taking several criteria into account to choose the preferential routing for each product. A cell formation heuristic is integrated in this MOGGA to group machines. The MOGGA optimises the cell formation by modifying the routings used for each product. The evaluation of the solutions is based on different criteria such as the cell formation evaluation, the similarity between products assigned to a machine, the cost and flexibility evaluation on the basis of limit of machine utilisation, etc.

The objective to form cells and part families was based on a double grouping (operations and machines). The final solution is a proposition of cells defining part families. The whole objective of the cell formation problem is to transform a job shop system to a cellular manufacturing system where all machines are physically moved to form the final cells. The proposed approach can be used to form only part families thanks to the cell solution. It would allow to keep the job shop layout and to choose the adequate routing for each part respecting the fictive grouping into cells.

In further studies, the embedded cell formation heuristic will be removed. The authors' objective is to create a new MOGGA able to solve two problems simultaneously at the same level. There will be two initialisation phases for each part of problem and two different genetic operators applied on each part of the solution. This new meta-heuristic will take into account grouping and separation constraints (machines must be placed in the same cell, or cannot) and constraints related to the qualification of the operators.

8. References

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

Baykasoglu, A. and Gindy, N. (2000) MOCACEF 1.0: multiple objective capability based approach to form part-machine groups for cellular manufacturing application. *International Journal of Production Research*, **38**(5), 1133-1161.

Brandon, J. A. (1996) *Cellular Manufacturing: Integrating Technology and Management*. Research Studies Press, Taunton, U.K.

Brans, J.-P. and Mareschal, B. (1994) The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decision support systems*, **12**, 297-310.

Burbidge, J. L. (1963) Production flows analysis. *The Production Engineer*, **42**, 742-752.

Burbidge, J. L. (1971) Production flows analysis. *The Production Engineer*, **50**(4-5), 139-152.

Burbidge, J. L. (1975) *The Introduction to Group Technology*. John Wiley & Sons, New York.

Caux, C., Bruniaux, R. and Pierreval, H. (2000) Cell formation with alternative process plans and machine capacity constraints: a new combined approach. *International Journal of Production Economics*, **64**(1-3), 279-284.

Chandrasekharan, M. P. and Rajagopalan, R. (1986) MOROC: an extension of rank order clustering for group technology. *International Journal of Production Research*, **24**(5), 1221-1233.

DeWitte, J. (1980) The use of similarity coefficient in production flows analysis. *International Journal of Production Research*, **18**, 503-514.

De Lit, P., Falkenauer, E. and Delchambre, A. (2000) Grouping genetic algorithms: an efficient method to solve the cell formation problem. *Mathematics and Computers in Simulation*, **51**, 257-271.

Diallo, M., Pierreval, H. and Quilliot, A. (1993) Manufacturing cells design with flexible routing capability in presence of unreliable machines. *International Journal of Production Research*, **31**(1), 175-182.

Falkenauer, E. (1998) *Genetic algorithm and grouping problems.* John Wiley & Sons, Chichester, UK.

Flanders, R. E. (1925) Design Manufacture and Production Control of a Standard Machine, *Transactions of the American Society of Mechanical Engineers*, **46**, 691-738.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, MA, .

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

Harhalakis, G., Nagi, R. and Proth, J.-M. (1990) An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research*, **28**(1), 185-198.

Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Irani, A. and Huang, H. (2000) Custom design of facilities layouts for multi-product facilities using layout modules. *IEEE Transaction on Robotics and Automation*, **16**(3), 259-267.

King, R. (1980) Machine-component grouping in production flow analysis: 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.

Kusiak, A and Chow, W. S. (1988) Decomposition of manufacturing systems. *IEEE Journal of Robotics and Automation*, **RA-4**(5), 457-471.

Logendran, R., Ramakrishna, P. and 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.

Lee, M. K., Luong, H. S. and Abhary, K. (1997) A genetic algorithm based cell design considering alternitive routings. *Computer-intergrated manufacturing systems*, **10**(2),93-107.

Mahesh, O. and Srinivasan, G. (2002) Incremental cell formation considering alternative machines. *International Journal of Production Research*, **40**(14), 3291-3310.

Mitrofanov, S. P. (1966) *The Scientific Principles of Group Technology*. British Library Lending Division (translation of original Russian text).

Mungwattana, A. (2000) Design of cellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility. PhD thesis, Blacksburg, Virginia.

Mosier, C. and Taube, L. (1985) Weighted similarity measure heuristics for the group technology machine clustering problem. *OMEGA International Journal of Management Sciences,* **13**(6), 577-583.

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

Rekiek, B., De Lit, P. and Delchambre, A., (2002) Hybrid assembly line design and user's preference. *International Journal of Production Research*, **40**(5), 1095-1111.

Rekiek, B., De Lit, P., Pellichero, F., L'Eglise, T., Fouda, P., Falkenauer, E. and Delchambre, A., (2001) A multiple objective grouping genetic algorithm for assembly line design. *Journal of Intelligent Manufacturing*, **12**(5-6), 469-485.

Sathiaraj, D. and Sarker, B. R. (2002) Common parts grouping heuristic: an iterative procedure to cell formation. *Production planning & control,* **13**(5), 481-489.

Shafer, S. M. and Rogers, D. F. (1993) Similarity and distance measures for cellular manufacture. Part II. *International Journal of Production Research,* **31**(6), 1315-1326.

Srinivasan, G. and Narendran, T. T. (1991) Grafics: a non technological clusrtreing algorithm fgor group technology application. *International Journal of Production Research,* **29**(3), 463-479.

Srinivasan, G. (1994) A clustering algorithm for machine cell formation in group technology using minimum spanning trees. *International Journal of Production Research,* **32**(2), 2149-2158.

Vivekanand, P. and Narendran, T. T. (1998) Logical Cell Formation in FMS, using flexibility-based criteria. *Int. J. Flexible manufacturing Systems*, **10**, 163-181.

Yin ,Y. and Yasuda, K. (2002) Manufacturing cells' design in consideration of various production factors . *International Journal of Production Research,* **40**(4), 885-906.