**Journal of Intelligent Manufacturing manuscript No.**
(will be inserted by the editor)

# Generalized Cell Formation: Iterative vs Simultaneous Resolution with Grouping Genetic Algorithm

**Emmanuelle VIN and Alain DELCHAMBRE**

Received: date / Accepted: date

**Abstract** For each industrial, lean manufacturing is "The method" to improve productivity and reduce cost. One of the tools for lean is cellular manufacturing. This technique reduces the factory to several small entities, which are easier to manage. The main difficulty of the cellular manufacturing is the creation of these small entities called cells. When the flexibility is used during the production process, two problems emerge. The first problem consists to solve a Resource Planning Problem in allocating the operations on a specific machine. The second problem concerns the Cell Formation Problem where the machines are grouped into cells. The algorithm proposed in this paper is based on a simultaneous resolution of two interdependent problems. This paper proves the efficiency of the simultaneous resolution comparing to the sequential resolution with iterations. To compare only the resolution way, a unique grouping genetic algorithm is used and adapted for both cases.

**Keywords** Cellular · Genetic Algorithm · Lean Manufacturing · Flow Analysis

## 1 Introduction

For each manager in industry, lean manufacturing is an essential method to improve productivity and to reduce cost. Several tools, such as just-in-time, kanban, or SMED, are known to achieve lean objectives. A less well-known tool, cellular manufacturing, implies a new organization as well as a new mindset.

Cellular production systems are an important application of group technology, which consists of decomposing systems into sub-systems, as well as

F. Author
50, Av. Franklin Roosevelt - CP 165/56
Tel.: +32-26-502814
Fax: +32-26-502856
E-mail: emmanuelle.vin@ulb.ac.be

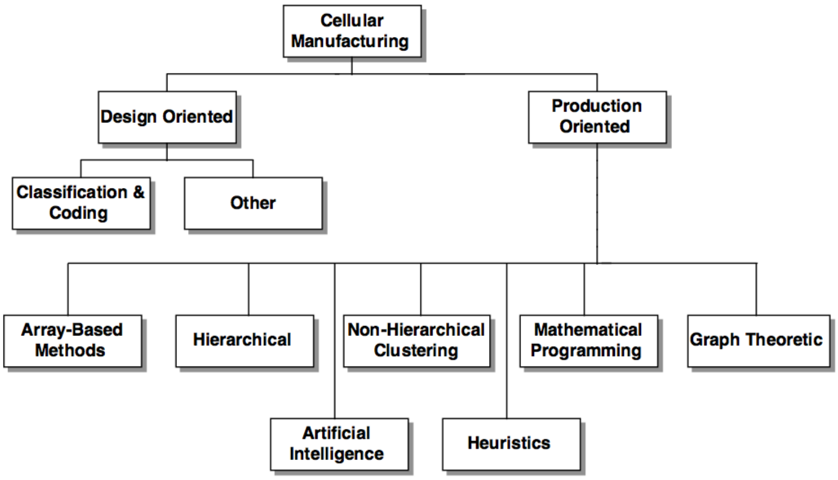



**Fig. 1** Classification of CFP technics.

grouping components together. Cellular manufacturing systems are based on the creation and management of several production cells. Complementary machines placed as close as possible to each other, and dedicated to a specific product family, compose these cells. One of the steps towards the implementation of these cells is precisely to define them, i.e. to resolve the Cell Formation Problem (CFP). Initially, the cell formation problem, and the methods used to create product families were relatively simple. Over time, the problem has evolved and become more difficult with data complexity. Progressively, new production parameters have been introduced, such as sequence operation, cost, alternative process plans, part volumes, machine capacity, labour-related factors, and flexibility. Vin (2010) proposes a classification and a complete review based on these parameters used for a more complex cell formation problem. During the last several years, the cell formation problem has been addressed in numerous works. Joines et al. (1996b) presents a complete review of production oriented manufacturing cell formation techniques. The used classification is represented in Fig. 1.

Vin (2010) proposed an original algorithm to solve a cell formation problem in presence of alternative processes and routes. These flexibilities increase the complexity of the Cell Formation Problem. Indeed, before grouping the machines into cells, each operation must be allocated to a specific machine. This problem is a Resource Planning Problem. For each part, it is necessary to choose a process and a particular routing. The particularity of this algorithm is based on the simultaneous treatment of both problems. In this paper, the efficiency of the simultaneous treatment is highlighted by comparing it with a sequential treatment with two embedded algorithms. To assure the comparison of the type of treatment, we use the same genetic algorithm, the same specification of the selection process, the genetic operators, and all others characteristics relative to the genetic algorithm. The heuristics used to initialize the population or reconstruct the solutions (after using the genetic operators) are exactly similar. The sole difference between both resolutions concerns a specific rate explained in section 4.2.

This paper is organized as follow: in section 2, we present the state of the art for the concept of alternative process/route and flexibility in the cell formation problem. This section concludes with a summary table of attributes addressed by authors when flexibility is used. Section 3 describes the mathematic formulation of the problem including the parameters, constraints and cost function. The proposed method (SIGGA, Simultaneous Grouping Genetic Algorithm), based on the grouping genetic algorithm (GGA), is described with the different characteristics of the implementation in section 4. A comparison of the simultaneous method with the iterative method is presented in section 5, followed by our conclusion.

## 2 State of the Art

Most often, researchers focused on using machine-part incidence matrix that provides only one fixed single route for each part which indicates the sequence of machines used to process each part. This assumption obviously ignores real situations for which each operation of a particular part may be performed on alternative machines. Parts always have more than one production route. Moreover, the existence of functionally similar/identical machines leads to alternative routings that can be used to obtain better cell design. Thus machine flexibility must be considered while forming cells in order to make the problem more realistic.

Generally, the design of independent manufacturing cells may not be possible without duplication of bottleneck machines, but at the same time, the duplication requires additional capital investment. Although the design of independent manufacturing cells requires additional investment, more simplified production planning and scheduling functions might provide enough savings to justify them. Considering that a part has two or more process plans and that each operation associated with a part can be processed on alternative machines, it may be possible to design independent manufacturing cells without much additional investments. Also, consideration of alternative process plans may greatly improve the flexibility and the efficiency of the grouping (Kusiak, 1987).

In the next section, the terms of alternative process and flexibility are developed.

### 2.1 Alternative processes

- A *process plan* is characterized by a sequence of machine types (lathe, etc.) required to transform raw material into some finished goods. This sequence is based on the operation sequence where each operation is characterized by a machine type regrouping all machines able to perform this operation.
- A *process route* or *routing* represents the sequence of specific machines or work centers on which a part passes through to complete its processing, as defined by the process plan.

A vast majority of the solution procedures used the Product Flow Analysis (*PFA*[1]) approach and assumed that each part has only one process route. In this case, the cell formation problem corresponds to a simple grouping problem of machines. Ignoring the alternative process may reduce the possibility of forming independent manufacturing cells. Taking into account these alternative process plans offers several benefits, such as allowing for a smaller number of machines, higher machine use, a reduced interdependence between cells, and an improved system throughput rate (Kusiak, 1987; Kang and Wemmerlov, 1993b). Advantages and possibility of having multiple process routes are discussed by Nagi et al. (1990).

The cell formation problem incorporating alternative process plans is called the **generalized group technology problem**. Kusiak (1987) was one of the first to consider these alternative processes. The presence of alternative routings is typical in many discrete, multi-batch, small lot size production environments (Defersha and Chen, 2006).

Different approaches permit to take into account the alternative processes or routings. But not all authors have the same specification in the definition of an alternative process. Alternatives may exist in any level of a process plan. The manner to take into account the processes or routings can be represented by the following six cases (classified by increasing order of alternative character):

1. Fixed routing (process route)
2. Routing with replicate machines
3. Routing with alternate machines for some operations
4. Several fixed routings
5. Fixed process plan
6. Alternative process plans

Each case is illustrated by an example of processes $Pr_{ij}$ for a part $P_i$ (for the $j^{th}$ process of part $i$) requiring five operations $\{O_{ij1} - O_{ij2} - O_{ij3} - O_{ij4} - O_{ij5}\}$. Following each case, the process $Pr_{ij}$ will be represented by a sequence of machine(s), $M_m$, or a sequence of machine type(s), $TM_t$. These notations are fully explained in Sec. 3.1

[1] PFA provides well-established, efficient and analytical engineering method for planning the change from process organization to product organization (Burbidge, 1971).

- Case 1. *One sequence of specific machines* defines the processing of each part: In this case, the operation sequence is considered but there is no routing nor process alternatives (Chandrasekharan and Rajagopalan, 1986; Goncalves and Resende, 2002; Stawowy, 2006).
  Example: $Pr_{ij} = \{M_1 - M_2 - M_3 - M_4 - M_5\}$. There is just one way to manufacture each part.
- Case 2. The processing of each part is represented by *one sequence of specific machines in presence of replicated machines*: In certain studies, authors do not define machine types for the entire park of machines but propose doubling or tripling the machines. These machines are functionally identical (Nagi et al., 1990; Ramabhatta and Nagi, 1998).
  Example: $Pr_{ij} = \{M_1 - (M_2 \text{ or } M_3) - M_4 - (M_2 \text{ or } M_3) - M_5\}$. The operations $O_{ij2}$ and $O_{ij4}$ can be achieved by the machines $M_2$ and $M_3$. In this case, we can group all machines into machine types. We can define four machine types ($TM_1 = \{M_1\}$, $TM_2 = \{M_2 \text{ and } M_3\}$, $TM_3 = \{M_4\}$ and $TM_4 = \{M_5\}$). The number of possibilities to manufacture the part amounts to 4 ($1 \times 2 \times 1 \times 2 \times 1 = 4$).
- Case 3. The processing of each part is represented by *one sequence of alternative machines*: In this case, we cannot group all machines into machine types. There is no consistency between machines assigned to operations. During the allocation of operations on specific machines, all combinations are possible following the choice of machines for each operation (Vivekanand and Narendran, 1998; Mungwattana, 2000).
  Example: $Pr_{ij} = \{(M_1 \text{ or } M_3) - M_2 - (M_2 \text{ or } M_3) - M_4 - (M_5 \text{ or } M_3)\}$. The machine $M_1$ and $M_3$ cannot be defined as belonging to the same type because they do not appear together for the next operations. It is impossible to correctly define any machine types dedicated to any operation types. In this example, there are 8 possible combinations to manufacture this part ($2 \times 1 \times 2 \times 1 \times 2 = 8$).
- Case 4. The processing of each part is represented by *several sequences of specific machines*: Some researches take these alternative processes into account by considering several possible machining sequences for each part (Kusiak, 1987; Gupta, 1993; Logendran et al., 1994; Lee et al., 1997; Sofianopoulou, 1999; Caux et al., 2000; Adenso-Diaz et al., 2001; Mahesh and Srinivasan, 2002; Wu et al., 2009). In this case, the number of machines used in the sequences can vary and each part has one or more process plans. The problem is summarized by the research of the good routing among those proposed.
  Example: The part $P_i$ has different possible routings: $Pr_{i1} = \{M_1 - M_2 - M_3 - M_4 - M_5\}$, $Pr_{i2} = \{M_3 - M_2 - M_6 - M_4 - M_5\}$ and $Pr_{i3} = \{M_6 - M_5 - M_3 - M_4\}$. The part can have two different routings with the same number of operations like $Pr_{i1}$ and $Pr_{i2}$, but it is not mandatory like shown by $Pr_{i3}$. In this example, the problem can be seen as the choice between the three routings for the proposed part to optimize the complete problem.
- Case 5. The processing of each part is represented by *one sequence of machine types*: The operation is defined by its tooling requirements. If the

tool is available on more than one machine, these machines are considered as belonging to the same type. Each machine type is capable of doing a specific operation. In this case, the problem is to find for each operation the good machine among all those belonging to the required type (Kusiak, 1987; Askin et al., 1997; Baykasoglu and Gindy, 2000; Diallo et al., 2001; Suresh and Slomp, 2001; Yin and Yasuda, 2002; Defersha and Chen, 2006). Example: The part $P_i$ has one process: $Pr_{ij} = \{TM_1 - TM_2 - TM_3 - TM_4 - TM_5\}$. In this case, the problem consists in choosing the best machine for each operation. There are many possible combinations. If all machine types $TM_i$ contain $n_i$ machines, the number of allowed combinations is equal to $n_1 \times n_2 \times n_3 \times n_4 \times n_5$.
- Case 6. The processing of each part is represented by *several sequences of machine types*: This case is the combination of the case 4 and 5. The number of machine types used in each process may be different. Each machine type comprises all machines able to perform the operation of its type. In this case, the number of possible combinations for each part explodes. The literature about these alternative processes is poor (Sankaran and Kasilingam, 1990; Mohamed, 1996; Sofianopoulou, 1999; Uddin and Shanker, 2002).
  Example: The part $P_i$ has three alternative processes: $Pr_{i1} = \{TM_1 - TM_2 - TM_3 - TM_4 - TM_5\}$, $Pr_{i2} = \{TM_1 - TM_5 - TM_6 - TM_4 - TM_2\}$ and $Pr_{i3} = \{TM_1 - TM_3 - TM_6 - TM_5\}$. In this case, for each part, the problem is resumed with the choice among the allowed processes of the best combination of machines (routing) for each part. The number of possible combinations is three times higher than in case 5 $((n_1 \times n_2 \times n_3 \times n_4 \times n_5) + (n_1 \times n_5 \times n_6 \times n_4 \times n_2) + (n_1 \times n_3 \times n_6 \times n_5))$.

All cases are summarized in Table 1 here below. In cases 2 to 5, we have alternative routings to different degrees. We consider the last case (6) as being sole alternative process in term of sequence of machine types. The use of these alternative processes is less frequent. Suresh and Slomp (2001) precise that in most firms no more than one process plan, and one set of tooling exist for each operation. However, inclusion of alternative process plans for each part type requires a major re-examination of process plans and shop floor practices, and major investments in tooling.

## 2.2 Use of alternative

Kim *et al.* consider three types of flexibility in process planning in term of data (Kima et al., 2003):

- *Operation or routing flexibility* relates to the possibility of performing an operation on alternative machines with, possibly distinct processing times and costs (Lin and Solberg, 1991).
- *Sequencing flexibility* corresponds to the possibility of interchanging the sequence in which manufacturing operations are performed.

| Case | Process | $O_{k1}$ | $O_{k2}$ | $O_{k3}$ | $O_{k4}$ | $O_{k5}$ |
|---|---|---|---|---|---|---|
| Case 1 | $Pr_{ij}$ | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ |
| Case 2 | $Pr_{ij}$ | $M_1$ | $M_2$ or $M_3$ | $M_4$ | $M_2$ or $M_3$ | $M_5$ |
| Case 3 | $Pr_{ij}$ | $M_1$ or $M_3$ | $M_2$ | $M_2$ or $M_3$ | $M_4$ | $M_5$ or $M_3$ |
| Case 4 | $Pr_{i1}$ | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ |
| | $Pr_{i2}$ | $M_3$ | $M_2$ | $M_6$ | $M_4$ | $M_5$ |
| | $Pr_{i3}$ | $M_6$ | $M_5$ | $M_3$ | $M_4$ | |
| Case 5 | $Pr(p_k)$ | $TM_1$ | $TM_2$ | $TM_3$ | $TM_4$ | $TM_5$ |
| Case 6 | $Pr_{i1}$ | $TM_1$ | $TM_2$ | $TM_3$ | $TM_4$ | $TM_5$ |
| | $Pr_{i2}$ | $TM_1$ | $TM_5$ | $TM_6$ | $TM_4$ | $TM_2$ |
| | $Pr_{i3}$ | $TM_1$ | $TM_3$ | $TM_6$ | $TM_5$ | |

**Table 1** Summary of the different notions of alternative routings and processes for a part $k$.

– *Processing flexibility* is determined by the possibility of producing the same manufacturing feature with alternative operations or sequences of operations (Benjaafar and Ramakrishnan, 1996).

The first and the second type, respectively, involve alternative machines and alternative sequences, but the operations to be performed are fixed. Allowing these flexibilities can provide better performance in mean flow time, throughput, and machine use (Lin and Solberg, 1991). We can make the correspondence between these three flexibilities and the definitions of alternatives presented in previous section 2.1. The operation flexibility corresponds to the definition of a type of machine containing the set of machines able to achieve the operation, i.e. the alternative process routes. The two other ones require the use of the alternative process plans. If there are several ways to design a product, different sequences of distinct operations can be proposed. Moreover, if the constraints in the product conception allow it, several sequences of operations can be defined in switching the order of operations. The use of alternative process plans does not make distinction between sequencing and processing flexibility. As explained in section 2.1, these routing and process alternatives increase the number of approaches one can use to form manufacturing cells.

To these three flexibilities, we add a last one: the *routing coexistence*. These flexibilities allowed by the alternative routing and process can be used in two ways. The majority of authors use alternative routings and alternative process to optimize the cell configuration in choosing the best route (Sofianopoulou, 1999; Mungwattana, 2000; Caux et al., 2000; Akturk and Turkcan, 2000; Baykasoglu et al., 2001; Jayaswal and Adil, 2004). On the other hand, to enjoy this flexibility, several authors prefer to allow alternative routings to coexist. This coexistence is possible when the batches of parts are split in smaller batches, or when the problem is solved during different periods like in dynamic reconfiguration cells (Sankaran and Kasilingam, 1990; Gupta, 1993; Heragu and Chen, 1998; Defersha and Chen, 2006; Tavakkoli-Moghaddam et al., 2008; Neto and Filho, 2010; Kia et al., 2012).

### 2.3 Why Hybrid Methods?

In using these flexibilities and/or the alternative process/routings, the Cell Formation Problem becomes a Generalized Cell Formation Problem. In that case there are two problems to solve.

- Selecting which process to produce each part as well as allocating the operator to a specific machine. This represents a specific problem of *resource planning*.
- Grouping machines and parts into cells. This is a *cell formation* problem.

With both problems, a question arises: how to solve these two groupings and in which order?

The resolution can be sequential, iterative or simultaneous. The sequential resolution finds a solution for the second problem based on the result found for the first problem or conversely. When two problems are inter-dependent, a good solution for the first problem does not imply a good solution for the second problem. To avoid the inconvenient of the sequential resolution, the iterative resolution is based on several iterations of the sequential resolution. But in this case, the second solution significantly depends on the first solution. Finally, the simultaneous resolution permits to optimize both problems simultaneously during each iteration, but it is more complex to solve.

As the Cell Formation Problem is well known to be a NP-hard problem, the meta-heuristic seems the most appropriate to solve it, in particular when the alternative process/routings are used. To treat the large-scale Generalized Cell Formation Problems, a large range of metaheuristics can be used. Gosh et al. (2011) present a complete review of these metaheuristics used in the cellular manufacturing. Amongst them, Genetic Algorithm is believed to be the most robust unbiased stochastic search algorithm for sampling a large solution space (Ghosh et al., 2010). These algorithms are particularly efficient for the problems with a high complexity. However authors used them also to optimize classical cell formation problems where the grouping efficacy is maximized (Sara and Ozcelik, 2012).

### 2.4 Comparative Table to Conclude

The literature cited above shows that several meta-heuristic approaches have been used to solve the cell formation problem with alternative process routings. In her thesis, Vin (2010) presents a large review of the literature, including a classification of the authors using hybrid methods to solve Generalized Cell Formation Problem with alternative routings and/or processes.

We present the Table 2 that summarizes this classification, based on a set of 34 published articles dealing with alternative routings and/or processes and their corresponding system features[2]. All features used for this comparative table are detailed here under (for a complete description, see (Vin, 2010)). They

[2] The reviews of Mansouri et al. (2000) and Defersha and Chen (2006) incorporate a wider range of input data and cell formation criteria.

represent the main features to take into account when solving a Generalized Cell Formation Problem. These parameters are divided into three categories: data parameters, flexibility parameters and resolution methods:

1. Data

| | |
|---|---|
| Production Volume (PV) | Volume of each parts to produce. |
| Machine Capacities (MC) | Available capacities of the machines. |
| Operating Time (OT) | Required time to achieve an operation on a machine. |
| Operation Sequences (OS) | Sequence of operations required to construct a part. |

2. Flexibility

| | |
|---|---|
| Routing Flexibility (RF) | Alternative sequence of machines or a sequence of machine type or a sequence of machines with machines replicated to define the process routes of each part. |
| Process Flexibility (PF) | Alternative process plans to produce each part. |
| Sequencing Flexibility (SF) | Alternative sequence with different arrangements of defined operations. |
| Routing Coexistence (RC) | Alternative routing existing in the same solution (routing flexibility). |

3. Resolution methods

| | |
|---|---|
| Resolution (R) | hierarchical (H), iterative (I) or simultaneous (S) |
| Multi-Objective (MO) | if the authors use several criteria to evaluate the solution. |
| Large-Scale Problems | (0 if the authors enumerate all alternative routings) (LS) |

By taking into account the production volume, it is possible to deal with the capacity of cell formation and to use the machine capacities as constraint. The use of operating times allows the computation of the machines' load. The operations sequences permit to insert the evaluation of the movements between machines and cells.

Dealing with routing, process and sequencing flexibilities allows the design of independent manufacturing cells without much additional investment as explained in section 2.3. Allowing the coexistence of alternative routings implies working with a lot splitting.

The effectiveness of the algorithm significantly depends on the strategy used for the resolution (hierarchically, iteratively or simultaneously). Indeed, the allocation of operations on machines and the cell formation problem can be solved successively, in any order. Iterations can be applied on the previous strategy to improve the solution. Therefore, the methods based on the simultaneous strategy will be preferred to make the solution evolving on the basis of both problems. It is interesting to propose a method suitable to solve large-scale problems using a multi-criteria evaluation to be applied to industrial cases.

| | Data | | | | Flexibility | | | | Resolution | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Authors | PV | MC | OT | OS | RF | PF | SF | RC | R | MO | LS |
| Kusiak (1987) | - | - | - | x | x | - | - | - | H | | |
| Choobineh (1988) | x | x | x | x | x | x | - | - | H | | |
| Nagi et al. (1990) | x | x | x | x | x | - | - | - | I | | |
| Sankaran and Kasilingam (1990) | x | x | - | - | x | - | - | x | H | | |
| Kusiak and Cho (1992) | - | - | - | - | x | - | - | - | I | x | |
| Rajamani et al. (1992) | x | x | - | - | x | - | - | - | S | | |
| Kang and Wemmerlov (1993a) | - | x | - | x | x | - | - | - | H | x | |
| Logendran et al. (1994) | x | x | x | x | x | - | - | - | H | | |
| Joines et al. (1996a) | - | - | - | x | x | - | - | - | I | - | |
| Hwang and Ree (1996) | - | - | - | - | x | - | - | - | H | - | 0 |
| Kazerooni et al. (1997) | x | - | x | x | x | - | - | - | H | - | |
| Gravel et al. (1998) | x | x | x | - | x | - | - | - | I | x | 0 |
| Heragu and Chen (1998) | x | x | - | - | x | - | - | - | ? | - | 0 |
| Ramabhatta and Nagi (1998) | x | x | x | x | x | - | - | - | I | | 0 |
| Chen and Heragu (1999) | - | - | - | - | x | - | - | x | H | - | x |
| Lozano et al. (1999) | x | x | x | x | x | - | - | x | H | - | 0 |
| Moon and Gen (1999) | x | x | x | - | x | x | - | - | S | | x |
| Sofianopoulou (1999) | - | - | - | x | x | x | - | - | S | | 0 |
| Caux et al. (2000) | - | x | - | - | x | - | - | - | S | | 0 |
| Won (2000) | - | - | - | - | x | - | - | - | I | x | |
| Zhao and Wu (2000) | x | x | x | x | x | - | - | - | I | x | |
| Adenso-Diaz et al. (2001) | - | - | - | x | x | - | - | - | S | - | x |
| Baykasoglu et al. (2001) | x | x | x | x | x | - | - | - | S | x | x |
| Mahesh and Srinivasan (2002) | x | x | x | x | x | - | - | - | H | - | 0 |
| Uddin and Shanker (2002) | x | x | x | - | x | x | - | - | I | - | |
| Vin et al. (2003) | x | x | x | x | x | - | - | - | I | x | x |
| Jayaswal and Adil (2004) | x | x | x | x | x | - | - | - | I | - | |
| Solimanpur et al. (2004) | x | x | x | x | x | - | - | - | S | x | x |
| Wu et al. (2004) | - | - | - | - | x | - | - | - | I | - | |
| Hu and Yasuda (2005) | x | - | - | x | x | - | - | - | S | - | x |
| Defersha and Chen (2006) | x | x | x | x | x | - | - | x | S | - | |
| Jeon and Leep (2006) | x | x | x | - | x | - | - | x | H | - | 0 |
| Mahdavi et al. (2006) | x | x | x | x | x | - | - | - | I | - | x |
| Nsakanda et al. (2006) | x | x | | - | x | x | - | - | I | - | x |
| Tavakkoli-Moghaddam et al. (2008) | x | x | x | x | x | x | x | x | S | 0 | |
| Wu et al. (2009) | - | - | - | - | x | - | - | - | S | - | |
| Cao et al. (2009) | x | x | x | x | x | x | x | x | I | - | |
| Vin (2010) | x | x | x | x | x | x | x | - | S | x | x |
| Tavakkoli-Moghaddam et al. (2012) | x | x | x | x | x | - | - | - | I | x | 0 |

**Table 2** Attributes used in the present study and in a sample of recently published articles.

In the comparative Table 2, the cell with the symbol "$x$" means that the authors deal with the feature (i.e. production volume). At the contrary, the symbol "$-$" means that authors do not use the feature. If the cell is empty, the authors do not specify this feature. If there is a 0 in the cell for the last criterion (LS), the author(s) enumerate(s) all the routings and/or processes to deal with all alternatives (for instance, in the incidence matrix or a similarity coefficient). The enumeration of all solutions implies that the method tends to be very consuming of computational resources. For this last feature, the hypothesis that the method is not applicable to large-scale problem can be reasonably assumed.

There are a lot of papers that treat the cell formation problem and propose new methods to solve this problem with multi-criteria evaluation that are adaptable for large-scale problem. However, papers dealing with routing flexibility and relevance for industrial problems are not frequent. Literature proposing solutions to solve Generalized Cell Formation Problem with routing flexibility and process flexibility are really lacking. The reason is that the complexity of the problem increases rapidly with the use of alternative machines and alternative process.

Vin (2010) described an original treatment of the Generalized Cell Formation Problem with a simultaneous grouping genetic algorithm completely parallelized. In this study, we demonstrate the advantages of choosing this simultaneous resolution by comparing it to the sequential resolution in an iterative process. For an efficient comparison, the algorithm used in both cases is the same algorithm, with a few modifications in parameters.

## 3 Mathematical Formulations

### 3.1 Notations - Indices

| | |
|---|---|
| $t$ | Machines types index ($TM_t$ = Machine type $t$). $t$ = 1,2,..,$n_T$ with $n_T$ = number of machine types |
| $m, n$ | Machines index ($M_m$= Machine $m$). $m$ = 1,2,..,$n_M$ with $n_M$ = number of machines |
| $i$ | Products index ($P_i$ = Product $i$). $i$ = 1,2,..,$n_P$ with $n_P$ = number of products |
| $j$ | Process index ($Pr_{ij}$ = Process $j$ of product $i$). $j$ = 1,2,..,$npr_i$ with $npr_i$ = number of processes for product $i$ |
| $k$ | Operations index ($O_{ijk}$ = Operation $k$ of process $j$ for product $i$). $k$ = 1,2,..,$no_{ij}$ with $no_{ij}$ = number of operations belonging to process $j$ of product $i$ |
| $c$ | Cells index ($C_c$ = Cell $c$) $c$ = 1,2,.., $n_C$ |

### 3.2 Parameters

| | |
|---|---|
| $A_m$ | Availability of the machine $m$. |
| $Q_i$ | Quantity of product $i$. |
| $BS_i$ | Batch size of product $i$. |
| $FT_i$ | Factor transport of product $i$. |
| $T_{ijk}$ | Average operating time for operation $O_{ijk}$. |
| $T_{ijkm}$ | Operating time if $O_{ijk}$ on machine $m$. |
| $S_c$ | 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 $m$ is unique and characterized by an availability parameter $A_m$, which is equal to its capacity value. Each machine belongs to at least one type and can belong to several types in case of a multi-functional machine. Each product is defined by a set of processes (Process = a sequence of $npr_i$ operations $O_{ij\ 1}$, $O_{ij\ 2}$, ..., $O_{ij\ npr_i}$). Each operation is defined as being accomplished on one machine type (lathe, grinding machine, etc.). Therefore each operation can be performed on all machines of a particular 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, $T_{ijkm}$). With these specifications, each product has several potential routings available for a specific process. The product is also characterized by its demand ($Q_i$), the batch size ($BS_i$) corresponding to the number of products transferred between two machines and the factor transport ($FT_i$). The last factor allows specifying the characteristics of the transport. Bigger is the product (cumbersome or fragile), greater is this factor. The variable $S_c$ represents the maximum size of the cell $c$. This size can be identical for all cells or not.

### 3.3 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).

These decision variables are illustrated with the following instance. 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 2 of part 1 is used in the solution. This variable implies that all other variables $x_{1j\neq 2}$ of the same part ($P_1$) 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 assignment of machines in each cell.

### 3.4 Constraints

The following constraints imposed in the model are presented in three categories: constraints on the data set and the input parameters, constraints during the treatment and constraints on the final solution.

**Initial Constraints**

The cell size and the number of cells introduced by the user must be adequate to contain all machines, $n_M$ (Eq. 1). There must be sufficient machine capacity to produce the specified part mix. The cell size must be specified. Upper and lower bounds ($UB_c$ and $LB_c$) can be used instead of a specific number. These parameters can be different for each cell and represent the maximum and minimum size of each cell $c$. And finally, the number of cells, $n_C$, in the system must be specified *a priori*.

$$\sum_{c=1}^{n_C} S_c \geq n_M \tag{1}$$

**Processing Constraints**

The machine charge cannot exceed the machine capacity (Eq. 2). Eq. 3 represents the process selection: only one process can be chosen by part. Eq. 4 defines that the total number of machines allocated inside cell $c$ must respect the lower and upper limits.

$$\sum_{i=1}^{n_P} \sum_{j=1}^{npr_i} \sum_{k=1}^{no_{ik}} Q_i \cdot T_{ijkm} \cdot y_{ijkm} \leq A_m \qquad \forall m \tag{2}$$

$$\sum_{j=1}^{npr_i} x_{ij} = 1 \qquad \forall i \tag{3}$$

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

**Final Constraints**

The solutions must satisfy the processing constraints and the last ones presented here below. Eq. 5 requires that each operation belonging to a selected process must be assigned to one machine. Each used machine must be allocated in one cell (Eq. 6).

$$\text{if } (x_{ij} = 1) \Rightarrow \sum_{m=1}^{n_M} y_{ijkm} = 1 \qquad \forall i, j, k \tag{5}$$

$$\sum_{c=1}^{n_C} z_{mc} = 1 \qquad \forall m \tag{6}$$

3.5 Cost function : minimization of Intra-Cellular traffic

$\phi_{mn}$ Traffic from machine $m$ to machine $n$ expressed in terms of number of batches pondered by the transport factor. For all used processes, the operating sequence of each part is analyzed and the traffic is sum if two consecutive operations are assigned respectively to the machines $m$ and $n$. We have:

$$\phi_{mn} = \sum_{i=1}^{n_P} \left( \sum_{j=1}^{npr_i} x_{ij} \cdot \left( \sum_{p=1}^{n_{O_{ik}}-1} \left( y_{ijkm} \cdot y_{ij(k+1)n} \right) \left( \frac{Q_i}{BS_i} \cdot FT_i \right) \right) \right) \quad \forall m \neq n \tag{7}$$

$\Phi_{Intra}$ Intra-cellular traffic computed as the sum of the traffic between all machines belonging to the same cell:

$$\Phi_{Intra} = \sum_{c=1}^{n_C} \left( \sum_{m=1}^{n_M} \sum_{n=m+1}^{n_M} (z_{mc} \cdot z_{nc}) (\phi_{mn} + \phi_{nm}) \right) \tag{8}$$

$\Phi_{tot}$ Total traffic in the system computed as the sum of the traffic between machines allocated to all cells:

$$\Phi_{Tot} = \sum_{m=1}^{n_M} \sum_{n=m+1}^{n_M} (\phi_{mn} + \phi_{nm}) \tag{9}$$

The function to minimize is the percentage of intra-cellular traffic:

$$\frac{\Phi_{Intra}}{\Phi_{Total}} \tag{10}$$

## 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. Chromosome worth is measured by its fitness, which is often simply the objective functional value of the search space point defined by the (decoded) chromosome. Falkenauer (1998) pointed out the weaknesses of standard GAs when applied to grouping problems. He 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.

As explained in the introduction, the Generalized Cell Formation Problem (GCFP) consists to solve two problems: the process selection with the assignment of each operation on a specific machine able to achieve it (Resource Planning problem: operation/machine); the grouping of machines into independent cells (Cell Formation problem: machine/cell). Both problems are interdependent because groups (machines) of the RP problem are precisely the objects to group in the CF problem. As shown in Fig. 2, each solution for RP problem (the black point in the RP search space) is associated to another large search space specific for the CF problem. For this reason, the classical resolution of the Generalized Cell Formation Problem consists to use a meta-heuristic to solve the first problem with an embedded heuristic to solve the second problem. For each solution of the first problem (for instance, in the

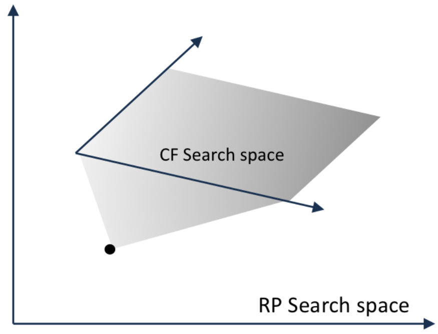



**Fig. 2** Embedded search spaces.

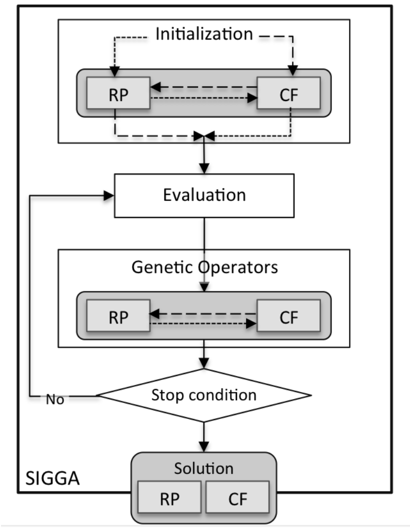
RP Search space), another technics must be used to explore the second search space or just give a good solution failing to the optimal solution.

On the basis of this technic, Vin et al. (2006) presented a adapted GGA (RPGGA with an embedded CF heuristic) in two steps. The algorithm is based on an iterative sequential method. First, a population of Resource Planning solution (operation/machine) is initialized in the genetic algorithm. Next, for each solution and during each generation of the GGA, a heuristic is applied to complete the chromosome with a valid solution for the second problem, the Cell Formation Problem (machine/cell): we are grouping the machines into cells. This method was favorable except for complex cases where the Cell Formation search space is too large to be explored with a heuristic. The heuristic cannot find the best solution. Next solution was to embed another metaheuristic inside the first one (RPGGA with embedded CFGGA) to be able to make an efficient search in this large search space when the cases are complex Vin (2010). The disadvantage of this technic is the resolution time. There are as many calls of the second metaheuristic as the number of chromosome in the population times the number of generations. For large-scale problems, the resolution time explodes.

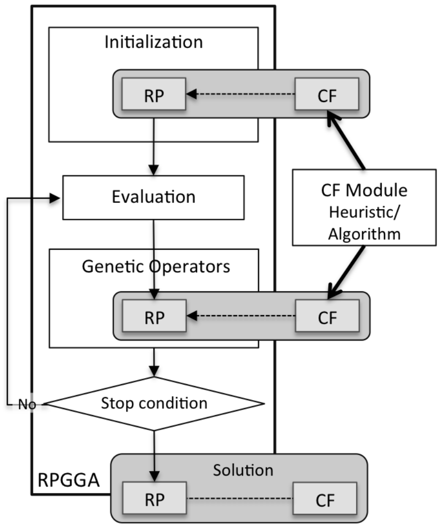
For these reasons, an adapted GGA able to solve simultaneously two interdependent problems has been developed. This algorithm is presented in the following section. Next, this SIGGA will be compared to both methods cited above (GGA with an embedded heuristic and GGA with another embedded GGA) to solve the Generalized Cell Formation Problem.

## 4.2 Description of the SIGGA

The SIGGA (Simultaneous resolution by a Grouping Genetic Algorithm) is presented in Fig. 3a (completely described by Vin (2010)). This algorithm is based on a classical GGA. But, each chromosome represents a valid solution to both problems (Resource Planning and Cell Formation).




(a)




(b)

**Fig. 3** (a) Adapted SIGGA (b) RPGGA with an integrated CF-module.

A population of chromosomes must be initialized. Both parts of each chromosome must be constructed to provide a valid solution respecting all constraints. A chromosome can be constructed following the manual way: all operations are allocated on a specific machine respecting, in particular, the capacity constraints (RP); theses allocations imply traffics or flows between machines when the product passes from a machine to another to be processed; on the basis of these traffics, the machines are grouped in minimizing the traffic between the groups, the cells (CF). This construction is represented in dotted line on Fig. 3a. It is meant that the RP initialization precedes the CF initialization ($RP \rightarrow CF$). In this case, a heuristic based on the traffic (flow-oriented heuristic) is used to solve the second problem (CF). But the traffic cannot be optimized for the first problem (RP). A classical First Fit heuristic (basic heuristic) respecting all hard constraints is used.

Conversely, a chromosome can also be constructed in the other way: all machines are grouped randomly into cells (CF), respecting the maximum cell size; the operations are allocated to a specific machine minimizing the traffic between the cells (RP). This is the situation in dash line where the RP initialization follows the CF initialization ($CF \rightarrow RP$). In this case, a specific heuristic (flow-oriented heuristic) based on the traffic is used to solve the second problem (RP) while the first problem is treated with a classical First Fit heuristic (basic heuristic) respecting the hard constraints.

This parameter of construction ($RP \rightarrow CF$ or $CF \rightarrow RP$) is a characteristic saved in the chromosome. The heuristics (basic or flow-oriented) used to construct the solutions generate only valid solutions respecting all hard constraints defined in section 3.4. The flow-oriented heuristic can never be used to construct the first problem (RP or CF) because there are not traffics between the machines yet. When the flow does exist, it is always a classical First Fit heuristic (basic heuristic) that is used. Once the machines are filled with operations or grouped into cells, the specific flow-oriented heuristic can then be used to complete the second part of the chromosome (CF or RP) by minimizing the intercellular traffic.

To control the initialization, two parameters, $ri_1$ and $ri_2$, are introduced ($ri_1 + ri_2 = 100$). These rates correspond to the proportion of the population that is constructed following the dotted line process or the dash line process respectively. In the SIGGA, both initialization rates are set to 50% to ensure a population sufficiently diversified.

After this initialization stage, the population is evaluated and the chromosomes are sorted. During this evaluation stage, the fitness of each chromosome (characterizing its quality) is computed thanks to the cost function (Eq. 10, percentage of intracellular flow). The chromosome of the population presenting the best percentage of intracellular flow is saved. In order to evolve to the optimal solution, different genetic operators are applied after a specific selection (tournament strategy). Indeed, this process selection converges much faster than roulette wheel selection (Zhong et al., 2005). During this step, the genetic operator rates, $ro_1$ and $ro_2$, are introduced because the chromosome is composed by two parts. They express the probability to apply the operators on the first problem ($ro_1$) or on the second one ($ro_2$). If $ro_1 + ro_2 > 100$, the operators can be applied on both parts of the chromosome (with a probability equal to $ro_1 + ro_2 - 100$). In SIGGA, both rates are set to 100%. In this way, the operators are applied to the complete chromosome to make both problems evolve simultaneously. These specific operators, mutation and crossover, are not an analyzed characteristic in this paper but they are detailed in a previous paper by Vin (2010). They are used to make evolve the population toward the best solution. We study specifically this method in this paper.

After the application of genetic operators, each chromosome is reconstructed following the same way than the initialization ($RP \rightarrow CF$ or $CF \rightarrow RP$). The first part is reconstructed with the basic heuristic and the second part with the flow-oriented heuristic. Then, a new generation is started. The algorithm stops when the maximum number of generations is reached or when the algorithm finds the solution without flow between cells.

## 5 Comparison of resolutions

### 5.1 Utilization of the SIGGA for the sequential resolutions

As explained in Sec. 4.1, the sequential resolution consists of integrating another module inside the main metaheuristic. This module can be a heuristic (not optimal) or another metaheuristic (greedy in CPU and bad resolution time). The algorithm SIGGA is compared against four others sequential resolutions using the same chromosome containing two parts for both problems (RP and CF):

- a GGA treating the RP problem with an integrated heuristic to treat the CF problem (= RPGGA with an embedded $HeurCF$) ;
- a GGA treating the RP problem with another integrated metaheuristic to treat the CF problem (= RPGGA with an embedded $GGACF$);
- a GGA treating the CF problem with an integrated heuristic to treat the RP problem (= CFGGA with an embedded $HeurRP$);
- a GGA treating the CF problem with another integrated metaheuristic to treat the RP problem (= CFGGA with an embedded $GGACF$);

The RPGGA is a genetic grouping algorithm addressing the RP problem where all the operations are allocated to a specific machine. To group these machines into cells and take into account the minimization of the intercellular traffic in the evaluation of the RPGGA, the CF-part must be constructed with an integrated module. This integrated module is called to construct the CF-part of the chromosome each time the RP-part is modified, i.e. after the initialization and after all the uses of the genetic operators. During all the treatment by the RPGGA, the genetic operators used to make evolved the population never modify the CF-part.

In Fig. 3(b), the flow chart of the RPGGA with embedded CF-module is shown. This algorithm allows solving both problems (RP and CF) in a sequential way with iterations in the main GGA. The module called to construct the second part can be a heuristic or another algorithm/metaheuristic. In this paper, both cases have been tested to construct this second part:

- the RPGGA with an embedded $HeurCF$: a good solution for CF-part is found thanks to the specific heuristic flow-oriented ($HeurCF$ in both figures: Fig. 4 and Fig. 5);
- the RPGGA with an embedded $GGACF$: to complete each RP-part, an optimal solution for the CF-part is found by a $GGACF$ (specific GGA to solve CF problem).

In the second case, a new GGA ($GGACF$) is called upon for each individual chromosome in the population and for each modification of the RP-part by the operator. As shown in Fig. 2, the integrated GGA allows exploring the second search space (CF) associated to each solution of the first search space (RP). The integrated heuristic will find a good solution in this search space without exploration.

The CFGGA corresponds to a main GGA solving the CF problem where all the machines are grouping into cells. The allocation of all operations on these cells is made by an integrated RP module called each time the CFGGA changes the CF part of the chromosome. Fig. 3(b) where each RP is replaced by CF and conversely, represents this CFGGA with an embedded RP-module: heuristic ($HeurRP$) or another GGA ($GGARP$). The explanation is the same as previously in reversing all “RP” and “CF”.

| Resolution | SIGGA | RPGGA | CFGGA |
|---|---|---|---|
| $ri_1$ | 50 | 100 | 0 |
| $ri_2$ | 50 | 0 | 100 |
| $ro_1$ | 100 | 100 | 0 |
| $ro_2$ | 100 | 0 | 100 |

**Table 3** Adaptation of the SIGGA parameters for the sequential resolutions.

Thanks to the parameters introduced in the previous section (Sec. 4.2), the algorithm SIGGA can be used like a sequential resolution. Tab. 3 summarized the values of these parameters in each case. For the RPGGA, the rates for the first problem ($ri_1$ and $ro_1$) are set to 100% and the rates for the second problem ($ri_2$ and $ro_2$) are set to 0%. With these parameters, the SIGGA runs in the exactly same way than the RPGGA with the integrated CF-module. Indeed, all the chromosomes will be initialized in the way $RP \rightarrow CF$ and the operators will be applied only on the RP-parts.

## 5.2 Comparison of sequential and simultaneous resolutions

Five resolutions have been compared in terms of number of generations and the time necessary to achieve the best solution. These resolutions have been applied to a large benchmark of case studies. The results presented in this paper are based on an ideal case study whose resolution is possible by creating completely independent cells. In this case, there are 15 types of machines with one to three different machines, 50 products with 2 different processes (operations sequences). The search space for CF problem is evaluated around $10^{15}$ solutions and $10^{75}$ for the RP search space[3].

The Tab. 4 summarizes the results for these fives algorithms. The number of generations needed to achieve 100% of intracellular traffic is written as the number of seconds to achieve this result. Only two resolutions have values for these parameters because the others never achieve 100% of traffic inside the cells. The next two lines are relevant for these other resolutions. They correspond to the value of the cost function and the resolution time after 100 generations.

| Resolution | SIGGA | RPGGA with embedded | | CFGGA with embedded | |
|---|---|---|---|---|---|
| | | $HeurCF$ | $GGACF$ | $HeurRP$ | $GGARP$ |
| Nb of gen. to achieve 100% | 58 | - | - | - | 67 |
| Time to achieve 100% | 42 | - | - | - | 4356 |
| Cost function after 100 gen. | 100 | 95 | 83 | 97 | 100 |
| Time after 100 gen. | - | 67 | 15608 | 86 | - |

**Table 4** Summarize of the results.

[3] We send the complete case on request

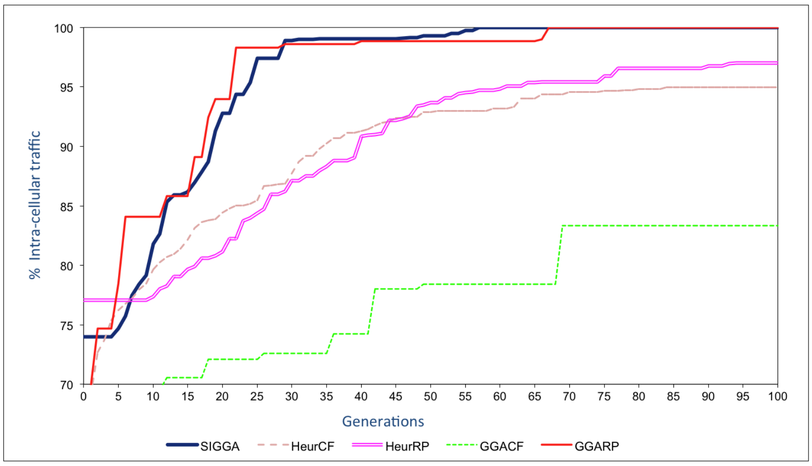



**Fig. 4** Intracellular flow in function of the generations.

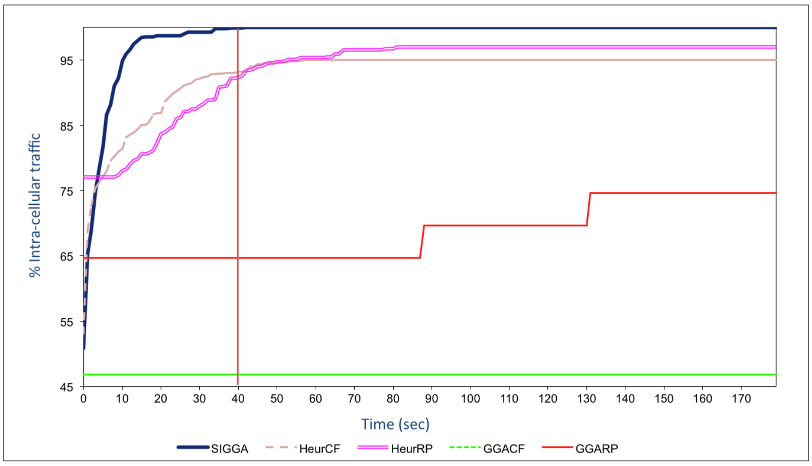



**Fig. 5** Evolution of the intracellular flow with the time.

Two graphics are presented. Fig. 4 expresses the evolution of the best solution (in term of percentage of intracellular traffic) in function of the number of generations. Fig. 5 represents this evolution in function of the resolution time.

Different observations can be made about these results:

- Only SIGGA and CFGGA with an embedded $GGARP$ achieve 100% of intracellular traffic before 100 generations (respectively 58 and 67 generations).
- Both GGA (RPGGA and CFGGA) with an embedded heuristic (respectively $GGACF$ and $GGARP$) achieve more than 95% of intracellular traffic after 100 generations.
- There is a significant difference in the evolution of the solutions comparing the RPGGA with embedded $GGACF$ and the CFGGA with the embedded $GGARP$. The explanation lies in the size of the search spaces. The main GGA evolves faster in a small search space. Then the main GGA will be faster in the CF search space ($10^{15}$ solutions for CF problem) than in the RP search space ($10^{75}$ solution). In increasing the number of generations, the RPGGA will converge also to 100% of intracellular traffic.
- Both resolutions of the RPGGA with embedded heuristic $HeurCF$ and CFGGA with embedded heuristic $HeurRP$ are similar in terms of generations and in terms of resolution time. The evolution of the best solution in function of the time is fast and regular. However, the best solution is not found after 100 generations.
- The RPGGA with an embedded $GGACF$ seems the worst resolution. The RP search space is so wide that it is a non-senses to embed another metaheuristic to generate the second part of the chromosome (CF-Part) as the solution for the first part can be poor (RP-Part).
- We can compare the CFGGA with an embedded $GGARP$ and with an embedded $HeurRP$. In terms of generations, the first resolution requires less generations to achieve the best solution. This is due to the fact that the integrated GGA approaches the optimal solution in the second search space while the heuristic sends back only a good solution.
  However, if we compare the resolution time, the use of the GGA greatly increases it. To achieve 97% of intracellular traffic, the CFGGA with an embedded $HeurRP$ need 86 seconds. At this time, only 65% of intracellular traffic is achieved by the CFGGA with an embedded $GGARP$.
- Comparing the resolution using SIGGA and the CFGGA with an embedded $GGARP$, the curbs on the Fig. 4 are very similar. It means that the number of generations to achieve the optimal solution is identical for both resolutions. However, when we look at the time required to achieve this result, the result is clear. The SIGGA amounts to 98% of intracellular traffic within 20 seconds. After only 42 seconds, the optimal solution is found. At the same time, the CFGGA with an integrated $GGARP$ is just at 67% of intracellular traffic.

## 6 Conclusion

An adapted genetic algorithm (SIGGA) has been presented by Vin (2010) to simultaneously solve two interdependent problems in Generalize Cell Formation Problem: the allocation of a specific machine to each operation and the machine grouping into cells. In this paper, after a presentation of the mathematical model, this innovative methodology is compared to the iterative resolutions with integrated heuristics or others genetic algorithms. We have used the same Grouping Genetic Algorithm with identical parameters to ensure that the methods of resolution (iterative or simultaneous) are comparable.

Indeed, the SIGGA has been developed to be adapted in both cases and to test all possible configurations (simultaneous GGA, GGA with embedded heuristics or GGA with another embedded GGA). The initial configuration of SIGGA depends only on initialization rates and operator rates.

The advantages of the simultaneous resolution SIGGA have been highlighted. SIGGA presents all the power of two embedded grouping genetic algorithms with the advantages of the resolution time of a simple genetic algorithm. The efficiency of this simultaneous resolution opens the door for the inclusion of another problem like the layout design. It is the ultimate step to produce a usable solution for the cellular manufacturing. Indeed, the industrials need a visual solution with the allocation of all machines to a specific place in the factory.

## References


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

M.S. Akturk and A. Turkcan. Cellular manufacturing system design using a holonistic approach. *International Journal of Production Research*, 38(1): 2327–2347, 2000.

R.G. Askin, H.M. Selim, and A.J. Vakharia. A methodology for designing flexible cellular manufacturing systems. *IIE Transactions*, 29(7):599–610, 1997.

A. Baykasoglu and N.N.A. Gindy. 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, 2000.

A. Baykasoglu, N.N.Z. Gindy, and R.C. Cobb. Capability based formulation and solution of multiple objective cell formation problems using simulated annealing. *Integrated Manufacturing System*, 12:258–274, 2001.

S. Benjaafar and R. Ramakrishnan. Measurement and evaluation of sequencing flexibility in manufacturing systems. *International Journal of Production Research*, 34:1195–1220, 1996.

J.L. Burbidge. Production flow analysis. *Production Engineer*, 50(4/5):139–152, 1971.

D. Cao, F.M. Defersha, and M. Chen. Grouping operations in cellular manufacturing considering alternative routings and the impact of run length

on product quality. *International Journal of Production Research*, 15(4): 989–1013, 2009.
C. Caux, R. Bruniaux, and H. Pierreval. Cell formation with alternative process plans and machine capacity constraints: a new combined approach. *International Journal of Production Economics*, 64(1-3):279–284, 2000.
M.P. Chandrasekharan and R. Rajagopalan. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24(2):451–464, 1986.
J.S. Chen and S.S Heragu. Stepwise decomposition approaches for large scale cell formation problems. *European Journal of Operational Research*, 113: 64–79, 1999.
F. Choobineh. A framework for the design of cellular manufacturing systems. *International Journal of Production Research*, 26(7):1161–1172, 1988.
F.M. Defersha and M. Chen. A comprehensive mathematical model for the design of the cellular manufacturing systems. *International Journal of Production Economics*, 103:767–783, 2006.
M. Diallo, H. Pierreval, and A. Quilliot. Manufacturing cells design with flexible routing capability in presence of unreliable machines. *International Journal of Production Economics*, 74(1):175–182, 2001.
E. Falkenauer. *Genetic Algorithms for Grouping Problem.* Wiley: New York, 1998.
T. Ghosh, P.K. Dan, S. Sengupta, and M. Chattopadhyay. Genetic rule based techniques in cellular manufacturing (1992-2010): a systematic survey. *International Journal of Engineering, Science and Technology*, 2(5):198–215, 2010.
J.F Goncalves and M. Resende. A hybrid genetic algorithm for manufacturing cell formation. Technical report, Rapport, 2002.
T. Gosh, S. Sengupta, M. Chattopadhyay, and P.K. Dan. Meta-heuristics in cellular manufacturing: A state-of-the-art review. *International Journal of Industrial Engineering Computations*, 2:87–122, 2011.
M. Gravel, A.L. Nsakanda, and W. Price. Efficient solutions to the cell-formation problem with multiple routings via a double-loop genetic algorithm. *European Journal of Operational Research*, 109:286–298, 1998.
T. Gupta. Design of manufacturing cells for flexible environment considering alternative routeing. *International Journal of Production Research*, 31:1259–1273, 1993.
S.S. Heragu and J.S. Chen. Optimal solution of cellular manufacturing system design: Benders' decomposition approach. *European Journal of Operational Research*, 107:175–192, 1998.
J.H. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor, Michigan: The University of Michigan Press, 1975.
L. Hu and K. Yasuda. Application of genetic algorithm for bin packing. *International Journal of Production Research*, pages 1–35, 2005.
H. Hwang and P. Ree. Routes selection for the cell formation problem with alternative part process plans. *Computers & Industrial Engineering*, 30(3): 423–431, 1996.

S. Jayaswal and G.K. Adil. An efficient algorithm for cell formation with sequence data, machine replications and alternative process routings. *International Journal of Production Research*, 42(12):2419–2433, 2004.

G. Jeon and H.R. Leep. Forming part families by using genetic algorithm and designing machine cells under demand changes. *Computers & Operations Research*, 33(1):263–283, 2006.

J.A. Joines, C.T. Culbreth, and R.E. King. Manufacturing cell design: an integer programming model employing genetic algorithms. *IEE Transactions*, 28:69–85, 1996a.

J.A. Joines, R.E. King, and C.T. Culbreth. A comprehensive review of production-orien manufacturing cell formation techniques. *International Journal of Factory Automation and Information Management*, 3(3-4):225–265, 1996b.

S. Kang and U. Wemmerlov. A work load-oriented heuristic methodology for manufacturing cell formation allowing reallocation of operations. *European Journal of Operational Research*, 69(3):292–311, 1993a.

S.L. Kang and U. Wemmerlov. A work load-oriented heuristic methodology for manufacturing cell formation allowing reallocation of operations. *European Journal of Operational Research*, 69:292–311, 1993b.

M. Kazerooni, H.S. Luong, and K. Abhary. A genetic algorithm based cell design considering alternative routing. *Computer-Integrated Manufacturing Systems*, 10(2):93–108, 1997.

R. Kia, A. Baboli, N. Javadian, R. Tavakkoli-Moghaddam, M. Kazemi, and J. Khorrami. Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing. *Computers & Operations Research*, 39:2642–2658, 2012.

Y.K. Kima, K. Park, and J. Ko. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling". *Computers & Operations Research*, 30:1151–1171, 2003.

A. Kusiak. The generalized group technology concept. *International Journal of Production Research*, 25:561–569, 1987.

A. Kusiak and M. Cho. Similarity coefficient algorithm for solving the group technology problem. *International Journal of Production Research*, 30(11): 2633–2646, 1992.

C.Y. Lee, L. Lei, and M. Pinedo. Current trends in deterministic scheduling. *Annals of Operations Research*, 70:1–41, 1997.

Y.J. Lin and J.J. Solberg. Effectiveness of flexible routing control. *The International Journal of Flexible Manufacturing Systems*, 3:189–211, 1991.

R. Logendran, P. Ramakrishna, and C. Srikandarajah. Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans. *European Journal of Operational Research*, 32(2):273–297, 1994.

S. Lozano, F. Guerrero, I. Eguia, and L. Onieva. Cell design and loading in the presence of alternative routing. *International Journal of Operational Research*, 37(14):3289–3304, 1999.

I. Mahdavi, J. Rezaeian, K. Shanker, and Z.R. Amari. A set partitioning based heuristic procedure for incremental cell formation with routing flexibility. *International Journal of Production Research*, 44(24):5343–5361, 2006.

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

S.A. Mansouri, S.M. Moattar-Husseini, and S.T. Newman. A review of the modern approaches to multi-criteria cell design. *International Journal of Production Research*, 38(5):1201–1218, 2000.

Z. Mohamed. A flexible approach to (re)configure flexible manufacturing cells. *European Journal of Operational Research*, 95:566–576, 1996.

C. Moon and M. Gen. A genetic algorithm-based approach for design of independent manufacturing cells. *International Journal of Production Economics*, 20(60-1):421–426, 1999.

A. Mungwattana. *Design of Cellular Manufacturing Systems for Dynamic and Uncertain Production Requirements with Presence of Routing Flexibility*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2000.

R. Nagi, G. Harhalakis, and J. Proth. Multiple routings and capacity considerations in group technology applications. *European Journal of Operational Research*, 28(12):2243–2257, 1990.

A.R.P. Neto and E.V.G. Filho. A simulation-based evolutionary multiobjective approach to manufacturing cell formation. *Computers & Industrial Engineering*, 59:64–74, 2010.

A. Nsakanda, M. Diaby, and W.L. Price. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings". *European Journal of Operational Research*, 171(3):1051–1070, 2006.

D. Rajamani, N. Singh, and Y.P. Aneja. Selection of parts and machines for cellularization: a mathematical programming approach. *European Journal of Operational Research*, 62(1):47–54, 1992.

V. Ramabhatta and R. Nagi. An integrated formulation of manufacturing cell formation with capacity planning and multiple routings. *Annals of Operations Research*, 77:79–95, 1998.

S. Sankaran and G. Kasilingam. An integrated approach to cell formation and part routing in group technology manufacturing systems. *Engineering Optimization*, 16:235–245, 1990.

T. Sara and F. Ozcelik. A genetic algorithm with proper parameters for manufacturing cell formation problems. *Journal of Intelligent Manufacturing*, 23:1047–1061, 2012.

S. Sofianopoulou. Manufacturing cells design with alternative process plans and/or replicate machines. *International Journal of Production Research*, 37(3):707–720, 1999.

M. Solimanpur, P. Vrat, and R. Shankar. A multi-objective genetic algorithm approach to the design of cellular manufacturing systems. *International Journal of Production Research*, 42(7):1419–1441, 2004.

A. Stawowy. Evolutionary strategy for manufacturing cell design. *OMEGA. The International Journal of Management Science*, 34(1):1–18, 2006.

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

R. Tavakkoli-Moghaddam, N. Safaei, and F. Sassani. A new solution for a dynamic cell formation problem with alternative routing and machine costs using simulated annealing. *Journal of the Operational Research Society*, 59: 443–454, 2008.

R. Tavakkoli-Moghaddam, M. Ranjbar-Bourani, G.R. Amin, and A. Siadat. A cell formation problem considering machine utilization and alternative process routes by scatter search. *Journal of Intelligent Manufacturing*, 23: 1127–1139, 2012.

M.K. Uddin and K. Shanker. Grouping of parts and machines in presence of alternative process routes by genetic algorithm. *International Journal of Production Economics*, 76(3):219–228, 2002.

E. Vin. *Genetic Algorithms Applied To Generalized Cell Formation Problem.* PhD thesis, Ecole Polytechnique de Bruxelles, Universit Libre de Bruxelles, Belgium, 2010.

E. Vin, P.G. DeLit, and A. Delchambre. Une approche intgre pour rsoudre le problme de formation des cellules de production avec des routages alternatifs. In *MOSIM03 world symposium. April 23 - 25, 2003, France*, 2003.

E. Vin, P. Francq, and A. Delchambre. A grouping genetic algorithm (simoggas) simultaneously to solve two grouping problems applied to the cell formation problem with alternative process plans. In *Group Technology/Cellular Manufacturing (GTCM06)*, 2006.

P. Vivekanand and T.T. Narendran. Logical cell formation in fms, using flexibility-base criteria. *International Journal of Flexible Manufacturing Systems*, 10:163–181, 1998.

Y. Won. New p-median approach to cell formation with alternative process plans. *International Journal of Production Research*, 38(1):229–240, 2000.

T.H. Wu, J.F. Chen, and J.Y. Yeh. A decomposition approach to the cell formation problem with alternative process plans. *The International Journal of Advanced Manufacturing Technology*, 24(11/12):834–840, 2004.

T.H. Wu, S.H. Chung, and C.C. Chang. Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routing. *Expert Systems with Applications*, 36:3652–3661, 2009.

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

C. Zhao and Z. Wu. A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research*, 38(2):385–395, 2000.

J. Zhong, X. Hu, M. Gu, and J. Zhang. Comparison of performance between different selection strategies on simple genetic algorithms. In *Computational Intelligence for Modelling, Control and Automation, 2005 and Inter-*

*national Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference*, volume 2, pages 1115–1121, 2005.