Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle

A. Hertz
B. Jaumard
C. C. Ribeiro
W. P. Formosinho Filho

A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 28, nº 3 (1994), p. 303-328.

<http://www.numdam.org/item?id=RO_1994__28_3_303_0>

Recherche opérationnelle/Operations Research

(vol. 28, n° 3, 1994, p. 303 à 328)

A MULTI-CRITERIA TABU SEARCH APPROACH TO CELL FORMATION PROBLEMS IN GROUP TECHNOLOGY WITH MULTIPLE OBJECTIVES (*)

by A. HERTZ (1), B. JAUMARD (2),
C. C. RIBEIRO (3) and W. P. FORMOSINHO FILHO (4)

Communicated by Pierre TOLLA

Abstract. – *Group technology techniques are now widely used in many manufacturing systems. Several algorithms have been proposed for the optimal design of efficient manufacturing cells. The cell formation problem must take into account several objectives: the number of bottleneck operations, the number of bottleneck machines and/or parts, the intercell flow, the intracell workload balancing, the subcontracting costs, the machine duplication costs, and the workload of the busiest machine or cell, among others. In this paper, we propose a multi-criteria methodology for solving the cell formation problem with multiple objectives. This approach is based on the use of the tabu search heuristic for solving a sequence of single-objective, multiconstrained subproblems, in which each objective is taken and optimized in turn, following their order of relative importance. The subproblems are tackled by a strategic oscillation strategy. Computational results concerning an application to a bi-criteria problem are reported for instances with up to 100 machines and 1,000 parts.*

Keywords: Group technology, cellular manufacturing, cell formation, multiple objectives, tabu search.

Résumé. – *Les techniques de groupement technologique sont aujourd'hui largement utilisées dans de nombreux ateliers de fabrication. Un grand nombre d'algorithmes ont été proposés pour la conception optimale d'îlots efficaces de fabrication. Le problème de formation de cellules doit considérer plusieurs critères: le nombre d'opérations goulots, le nombre de machines et/ou pièces goulots, le flot intercellulaire, l'équilibrage des charges intracellulaires, les coûts de sous-traitance, les coûts de duplication des machines, et la charge de travail de la machine ou de l'îlot le plus chargé, entre autres. Dans cet article, nous proposons une méthodologie multi-critère pour le problème de formation d'îlots avec objectifs multiples. Cette approche est fondée sur l'utilisation de*

(*) Received September 1993.

(1) École Polytechnique Fédérale de Lausanne, Département de Mathématiques, MA-Ecublens, CH-1015 Lausanne, Switzerland, E-mail: hertz@eldi.epfl.ch.

(2) École Polytechnique de Montréal, Département de Mathématiques Appliquées, Case Postale 6079, Succursale A, Montréal, Québec, Canada H3C 3A7, E-mail: brigitt@crt.umontreal.ca

(3) Catholic University of Rio de Janeiro, Department of Computer Science, Rua Marquês de São Vicente 225, Rio de Janeiro 22453, Brazil, E-mail: celso@inf.puc-rio.br

(4) Catholic University of Rio de Janeiro, Department of Electrical Engineering, P.O. Box 38063, Rio de Janeiro 22452, Brazil, E-mail: walter@inf.puc.rio.br

l'heuristique de recherche tabou pour résoudre une suite de sous-problèmes à un seul objectif sous plusieurs contraintes, dans lesquels chaque objectif est considéré et optimisé à son tour, en suivant leur ordre d'importance relative. Une technique du type oscillation stratégique est utilisée pour résoudre les sous-problèmes. Nous présentons des résultats numériques concernant une application à un problème bi-critère pour des cas ayant jusqu'à 100 machines et 1 000 pièces.

Mots clés : Groupement technologique, production cellulaire, formation de cellules, objectifs multiples, recherche tabou.

1. INTRODUCTION

Group technology is a manufacturing philosophy based on the idea that parts which require similar operations and machines corresponding to these operations should as much as possible be grouped together into part families and machine cells. Group technology offers an interesting alternative to batch production and its use in a manufacturing system simplifies the flow of parts and tools, reduce set-up times, throughput times and work-in-process inventory, and improves design and manufacturing efficiency; see *e.g.* [19, 20, 23, 30, 46]. The manufacturing system based on machine cells is called *cellular manufacturing* and the process of assigning parts and machines to cells is called *cell formation.*

Ideally, machine cells should be mutually independent, each one processing its own family of parts. However, in practice, some parts will need to be processed by machines assigned to different cells. These are called *bottleneck parts* and *bottleneck machines*, respectively. Bottleneck elements entail intercell moves which make flow lines in the manufacturing system slower and more complex. The negative impact of intercell moves in cellular manufacturing is thoroughly discussed in Garza and Smunt [3]. In order to simplify the flow lines, the manufacturers may choose whether to subcontract bottleneck parts or to buy (duplicate) bottleneck machines, instead of carrying out intercell moves. The decision of accepting intercell moves rather than subcontracting parts or duplicating machines is usually a managerial decision based on a compromise between economic and technological criteria [29].

The cell formation problem has been dealt with by many authors and many heuristic approaches have been proposed in the litterature; see *e.g.* [21, 26, 32, 33, 34, 36, 40, 41, 42, 44]. Most of them consist in the formation of cells in such a way to minimize the number of bottleneck elements or to balance the intercell flow (*see* e.g. Miltenburg and Zhang [28] for a survey including comparative computational experiences) or the intracell load, and consider constraints such as the number of cells or the maximum number of elements per cell. These approaches usually do not consider the whole set of possible economic decisions. Kumar and Vannelli [22] proposed two heuristics for part subcontracting. The first one aims at determining the smallest number of parts to be subcontracted while eliminating all bottleneck elements, assuming that all subcontracting costs are equal. The second heuristic looks for the parts that should be subcontracted in order to minimize the subcontracting cost. Both are greedy heuristics and do not take into account either intercell moves or machine duplication. Seifoddini [35] presented a machine duplication heuristic which takes into account the duplication cost and the associated reduction in the intercell material handling cost (see also [36]). Logendran [27] discussed the economical advantages of machine duplication in the long run, taking into account the sequence of operations of parts and the budgetary limitations faced by the company.

Since several objectives may interfer in the process of cell design, some authors have recently developped multi-criteria approaches to the cell design problem. As bottleneck parts belonging to several cells might be subcontracted, an important objective is to minimize the resulting subcontracting cost while minimizing the intercell flow. Along the same lines, bottleneck machines which belong to several cells may be duplicated depending on their prices; so another objective is to minimize the machine duplication costs. Other objectives are the minimization of the workload of the busiest machine, the optimization of cell utilization, and the maximization of part production within a single cell, among others. Wei and Gaither [45] proposed a multiobjective greedy heuristic which takes into account four criteria (minimization of the bottleneck cost, maximization of the average cell utilization, minimization of the intracell and the intercell load imbalances) through the maximization of a weighted additive utility function. Venugopal and Narendran [43] gave a genetic bi-criteria algorithm whose objectives are the minimization of the volume of intercell moves together with the total within cell load variation. Shafer and Rogers [37] developed three goal programming models for considering four different objectives, which are taken into account through a weighted function: reduce set-up times, minimize intercell moves, minimize investment in new equipment, and maintain acceptable utilization levels. Hertz, Jaumard and Ribeiro [14] presented a tabu search approach for minimizing a multiple criteria weighted objective function taking into account the number of bottleneck elements, bounds on the number of elements in each cell, intracell load balance and some other technological criteria. More recently, Jaumard and Ribeiro [18] proposed an exact column generation approach for dealing with several formulations of the cell formation problem, all of them based on the set partitioning problem and using different models and objective functions for considering some of the criteria usually found in practice.

A two-phase approach is also often used to solve the cell formation problem. In a first phase, a heuristic procedure obtains a reasonably good grouping of machines and parts into cells. The extra cost resulting from the bottleneck elements is minimized in a second phase, *i. e.*, one decides which machines should be duplicated and which parts should be subcontracted. The still remaining bottleneck parts need to be processed in more than one cell and will thus induce intercell moves. Hertz, Jaumard and Ribeiro [15] gave an exact polynomial time algorithm for solving the second phase problem, assuming that the good grouping needed as the input for this phase has already been found by some heuristic procedure.

We propose in this paper a multi-criteria approach for solving the cell formation problem in group technology with multiple objectives, based on the use of the tabu search metaheuristic. We discuss in Section 2 several objectives which appear very often in the literature: cardinality constraints, the number of bottleneck operations, the number of bottleneck machines (or bottleneck parts), the intercell flow, and the intracell workload balancing. The metaheuristic tabu search is presented in Section 3. The basic features of the implementation strategy for the application of tabu search to the cell formation problem are described in Section 4. A multi-phase, iterative approach is then proposed for solving the multi-criteria cell formation problem. The approach described in Section 5 basically consists in solving a sequence of single-objective, multi-constrained subproblems, in which each objective is taken and optimized in turn, following their order of relative importance. A strategic oscillation technique is used to solve each subproblem. Computational results concerning an application to a bi-criteria problem are reported in Section 6. Some concluding remarks are discussed in the last section.

2. OBJECTIVES OF THE CELL FORMATION PROBLEM

Let $\mathcal{M} = \{ M_1, M_2, \cdots, M_m \}$ and $\mathcal{P} = \{ P_1, P_2, \cdots, P_p \}$ be, respectively, the set of machines and the set of parts, which have to be arranged into n cells $C_1, C_2, \cdots, C_n$. For any machine M_i (resp. part P_j), we denote by $cell\,(M_i)$ (resp. $cell\,(P_j)$) the cell to which it is assigned in the current solution. Let $A = \{ a_{ij} \}$ be the $m \times p$ matrix such that $a_{ij} = 1$ if and only if part P_j must be (at least partially) processed by machine M_i, $a_{ij} = 0$ otherwise.

We denote by α_j the cost of subcontracting part P_j, by β_i the duplication cost of machineM_i and by γ_{ij} the processing time of part P_j by machine M_i (γ_{ij} may also sometimes denote the material handling cost associated with the intercell move corresponding to taking part P_j from a cell to another one where machine M_i is located). Processing times may also comprise set-up costs within the cells to which the bottleneck parts are moved. Duplication and subcontracting costs can somehow take into account the reduction in the processing cost within the cell containing the bottleneck elements. A *bottleneck operation* is a pair $(M_i,\ P_j)$ such that M_i and P_j are assigned to different cells and $a_{ij} = 1$.

As discussed in Section 1, several objectives might often be taken into account in the framework of the cell formation process. In this paper, we are interested in designing a multi-criteria approach for cell formation, based on the tabu search metaheuristic, which could allow for simultaneously considering different objectives. We propose to take into account various constraints and objective functions, some of which are described below.

Every solution of the cell formation problem may be characterized by the set of machines and parts within each cell C_k, $k = 1, \cdots, n$. Let $x_i^k = 1$ if machine M_i is assigned to cell C_k, $x_i^k = 0$ otherwise. Analogously, let $y_j^k = 1$ if part P_j is assigned to cell C_k, $y_j^k = 0$ otherwise. Then, the number m^k (resp. p^k) of machines (resp. parts) within cell C_k is given by:

$$m^k = \sum_{i=1}^{m} x_i^k$$

$$p^k = \sum_{j=1}^{p} y_j^k.$$

In order to maintain a good level for system performance and reliability in case of machine failure, for instance, it might be wise, in some cases, to impose cardinality constraints on the number of machines and/or parts assigned to one cell. Since it is *a priori* difficult to identify a precise value which will guarantee to overcome the negative impact of the intercell flow (too few machines per cell) while ensuring a higher performance than a job-shop system (too many machines per cell), let $[\underline{m},\ \overline{m}]$ (resp. $[\underline{p},\ \overline{p}]$) be the range values for the cardinality constraints on the number of machines (resp. parts) within each cell. Since most approaches look for approximate solutions, these constraints may be viewed as soft constraints which may be taken into account through the minimization of the following objective functions:

- $f_1 = \sum_{k=1}^{n} \max\{0,\ \underline{m} - m^k\}$
- $f_2 = \sum_{k=1}^{n} \max\{0,\ m^k - \overline{m}\}$
- $f_3 = \sum_{k=1}^{n} \max\{0,\ \underline{p} - p^k\}$
- $f_4 = \sum_{k=1}^{n} \max\{0,\ p^k - \overline{p}\}$

Bottleneck operations entail intercell moves which make flow times in the manufacturing system slower and more complex. Then, among the objectives to be considered, one of the most important ones (which appear very often in the literature) is the minimization of the number of bottleneck operations, *i.e.*, pairs machine-part $(M_i,\ P_j)$ not assigned to the same cell such that part P_j must be processed by machine M_i:

- $f_5 = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{p} a_{ij}\,[x_i^k\,(1 - y_j^k) + y_j^k\,(1 - x_i^k)]$

Maybe more interesting than minimizing the number of bottleneck operations, it might be more useful to minimize the number of bottleneck machines (depending on their duplication coste, additional machines might be bought and installed in other cells) and/or the number of bottleneck parts (depending on their production cost and on the material handling costs, some of them might be subcontracted). The minimization of the number of bottleneck machines and that of the number of bottleneck parts correspond to the minimization of the following functions, respectively:

- $f_6 = \sum_{k=1}^{n} \sum_{i=1}^{m} \min\left\{1,\ \sum_{j=1}^{p} a_{ij}\,x_i^k\,(1 - y_j^k)\right\}$
- $f_7 = \sum_{k=1}^{n} \sum_{j=1}^{p} \min\left\{1,\ \sum_{i=1}^{m} a_{ij}\,y_j^k\,(1 - x_i^k)\right\}.$

Other objectives are the optimization of the intercell and the intracell workload balancing. These objectives may be dealt with by the minimization of the functions below. If the coefficient γ_{ij} stands for the material handling cost associated with the intercell move corresponding to processing part P_j by machine M_i located in another cell, function f_8 below represents the intercell load, *i. e.*, the total material handling cost of all bottleneck operations. Conversely, if the coefficient γ_{ij} denotes the processing time of part P_j by machine M_i, then the intracell workload of a given cell is given by the sum of the processing times of all operations performed by the machines assigned to this cell to the parts also assigned to it. The optimization of the intracell workload balance may then be achieved by the minimization of function f_9, which is given by the maximum, among all cells, of the deviation of the intracell workload with respect to the average intracell workload:

- $f_8 = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{p} \gamma_{ij} \left[x_i^k (1 - y_j^k) + y_j^k (1 - x_i^k)\right]$
- $f_9 = \frac{1}{2} \max_{k=1, \cdots, n} \left| \sum_{i:M_i \in C_k} \sum_{j:P_j \in C_k} \gamma_{ij} - \frac{1}{n} \sum_{k=1}^{n} \sum_{i:M_i \in C_k} \sum_{j:P_j \in C_k} \gamma_{ij} \right|$

Other objective functions may consider economical decisions, *i. e.*, the trade-offs between subcontracting costs, machine duplication costs and material handling costs due to intercell flows, see *e. g.* Hertz, Jaumard and Ribeiro [14] and Jaumard and Ribeiro [18].

3. TABU SEARCH

The presentation of the *tabu search* metaheuristic in this section follows that of Porto and Ribeiro [31]. To describe the tabu search metaheuristic, we first consider a general combinatorial optimization problem (P) formulated as to

minimize $f(s)$

subject to $s \in S$,

where S is a discrete set of feasible solutions. Local search approaches for solving problem (P) are based on search procedures in the solution space S starting from an initial solution $s_0 \in S$. At each iteration, a heuristic is used to obtain a new solution s' in the neighborhood $N(s)$ of the current solution s, through slight changes in s. Every feasible solution $\bar{s} \in N(s)$ is evaluated according to the cost function $f(\cdot)$, which is, eventually, locally optimized. The current solution moves smoothly towards better neighbor solutions, enhancing the best obtained solution s^*. The basic local search approach corresponds to the so-called hill-descending algorithms, in which a monotone sequence of improving solutions is examined, until a local optimum is found.

Any hill-descending algorithm depends on two basic mechanisms: the initial solution heuristic and the neighbor search heuristic. The first should be capable of building from scratch an initial solution s_0. The neighbor search heuristic determines new neighbor solutions from a given current solution. In the most simple algorithm, it could be stated as a complete search for a neighbor solution with the lowest cost. In the case of the cell formation problem, the cost of a solution may be given by any appropriate combination of the objective functions f_1 through f_9 (or any other ones) described in Section 2.

A move is an atomic change which transforms the current solution, s, into one of its neighbors, say $\bar{s}$. Thus, *movevalue* $= f(\bar{s}) - f(s)$ is the difference between the value of the cost function after the move, $f(\bar{s})$, and the value of the cost function before the move, $f(s)$. With these definitions, the description of a hill-descending algorithm in Figure 1 is straightforward.

```
begin
  Generate an initial solution s0
  s, s* ← s0
  repeat
    bestmovevalue ← ∞
    for (all candidate moves) do
    begin
       Let s̄ be the neighbor solution associated with the current candidate move
       movevalue ← f(s̄) − f(s)
       if (movevalue < bestmovevalue) then
       begin
          bestmovevalue ← movevalue
          s' ← s̄
       end
    end
    if (bestmovevalue < 0) then s* ← s'
    s ← s'
  until (bestmovevalue ≧ 0)
end
```

Figure 1. – Basic hill-descending algorithm.

Hill-descending algorithms always stop in the first local otpimum. To avoid this drawback, several metaheuristics have been proposed in the

```
begin
  Initialize the short term memory function
  Generate the starting solution s0
  s, s* <- s0
  while (number of moves without improvement < maxmoves) do
  begin
    bestmovevalue <- ∞
    for (all candidate moves) do
      if (candidate move is admissible, i. e., if it does not belong to the tabu list) then
      begin
        Obtain the neighbor solution s̄ by applying candidate move to the current solution s
        movevalue <- f(s̄) - f(s)
        if (movevalue < bestmovevalue) then
        begin
          bestmovevalue <- movevalue
          s' <- s̄
        end
      end
    end
    Update the short term memory function
    if (f(s') < f(s*)) then s* <- s'
    s <- s'
  end
end
```

Figure 2. – Basic description of the tabu search metaheuristic.

literature, namely genetic algorithms, neural networks, simulated annealing, and tabu search [9]. They all have an essential common approach: the use of certain mechanisms which permit that the search for neighbor solutions take directions of increasing the cost of the current solution in a controlled way, as an attempt to escape from local optima. The current solution may not be the best solution so far encountered, which means that the best solution must be maintained separately throughout the execution of the algorithm. This class of techniques are called metaheuristics, because the process of finding a good solution (eventually the optimal one) consists in applying at each step a subordinate heuristic which has to be designed for each particular problem [5, 9, 17].

Tabu search is an adaptive procedure which may be used for solving combinatorial optimization problems, which guides a hill-descending heuristic to continue exploration without becoming confounded by the absence of improving moves, and without falling back into a local optimum from which it previously emerged [6, 7, 8, 17, 24]. Briefly, the tabu search metaheuristic may be described as follows. At every iteration, an admissible move is applied to the current solution, transforming it into its neighbor with the smallest cost. Contrarily to a hill-descending scheme, moves towards a new solution that increases the cost function are permitted. The reverse move should be always prohibited along some iterations, in order to avoid cycling. These restrictions are based on the maintenance of a short term memory function which determines for how long a tabu restriction will be enforced or, alternatively, which moves are admissible at each iteration. Figure 2 gives a procedural description of the basic tabu search metaheuristic.

The performance of an algorithm using the tabu search metaheuristic is intimately dependent on the basic characterizing parameters, namely the time that the short memory function enforces a certain move to be tabu, and the maximum number of iterations, *maxmoves*, during which there may be no improvement in the best solution. The tabu tenure is an important feature of the tabu search algorithm, because it determines how restrictive is the neighborhood search. If it is too small, the probability of cycling increases. If it is too large, there is a possibility that all moves from the current solution are tabu and the algorithm may be trapped. However, it should be pointed out that cycle avoidance is not an ultimate goal of search process. In some instances, a good search path will result in revisiting a solution encountered before. The broader objective is to continue to stimulate the discovery of new high quality solutions. One implication of choosing stronger or weaker tabu restrictions is to render smaller or longer tabu tenures appropriate [10].

For large problems, where $N(s)$ may have too many elements, or for problems where these elements may be costly to examine, the aggressive choice orientation of tabu search makes it highly important to isolate a candidate subset (or a fraction) of the neighborhood, and to examine this subset instead of the entire neighborhood [10]. Successful applications of tabu search for combinatorial problems have been reported in the litterature, see *e. g.* [1, 2, 6, 12, 13, 16, 17, 25, 38, 39, 47] among many other references. Other advanced features, improvements, extensions, and implementation strategies of the basic tabu search procedure will be commented in the next section.

4. IMPLEMENTATION STRATEGY FOR CELL FORMATION

The basic tabu search metaheuristic is now specialized into a specific algorithm for the cell formation problem with multiple objectives.

Feasible solutions and admissible moves. Each possible solution s for the cell formation problem is defined by the set of machines and parts assigned to each one of the n cells. A solution is feasible if all cardinality constraints on the minimum and maximum number of machines and parts within each cell are respected. Let S denote the set of feasible solutions. A neighbor solution $\bar{s} \in N(s)$ is obtained by taking any element $e \in \mathcal{M} \cup \mathcal{P}$ (*i.e.*, either a machine or a part) assigned to cell C_s in the current solution and transfering it to another cell C_t, without violating any cardinality constraint. A move may then be characterized by a triple $(e,\ s,\ t)$, where $e \in \mathcal{M} \cup \mathcal{P}$, C_s is the cell where element e is currently assigned, and C_t is the cell to where element e is transfered, with $C_s \neq C_t$. A move is admissible if it is not prohibited. The cardinality of the neighborhood, and consequently the number of admissible moves, is $O((m+p) \cdot n)$.

Partial neighborhood exploration. As the cardinality of the set of neighbor solutions is very large, update formulae and advanced data structures should be used for keeping and updating the move values at each iteration. However, other implementations of the tabu search metaheuristic for other problems (see *e. g.* [12, 16]) have shown that it may also be successfully used without the examination of the whole neighborhood. In our implementation for the cell formation problem, we decided to examine only a fraction of the whole neighborhood.

Tabu list. A chief mechanism for exploiting memory in tabu search is to classify a subset of the moves in a neighborhood as forbidden (tabu). The classification depends on the history of the search, particularly manifested in the recency or frequency that certain move or solution components, called *attributes*, have participated in generating past solutions. Some choices of attributes may be better than others [10]. An attribute is defined to be *tabu-active* when its associated reverse attribute has occurred within a stipulated interval of recency in past moves. An attribute that is not tabu-active is said to be *tabu-inactive*. The condition of being tabu-active or tabu-inactive is called the *tabu status* of an attribute. A move may contain tabu-active attributes, but still may not be tabu if these attributes are not sufficient to activate a tabu restriction. A move can be determined to be tabu by a restriction defined over any set of conditions on its attributes, provided these attributes are currently tabu-active.

The short memory function is represented by a finite list of tabu moves. Let $(e,\ s,\ t)$ be the move made from the current solution, where $e \in \mathcal{M} \cup \mathcal{P}$ is an element (*i.e.*, either a machine or a part) currently assigned to cell C_s which is transfered to cell C_t. Then, the reverse move $(e,\ t,\ s)$ should be prohibited along some iterations. The attribute which must be made tabu-active is defined as the pair $(e,\ s)$, thus prohibiting element e to be reassigned to cell C_s during a certain number of iterations, which is called the tabu tenure of the attribute. An $(m + p) \times n$ matrix *tabu* may then be used to implement this short term memory. This matrix is initialized with zeroes. Whenever a move $(e,\ t,\ s)$ is made tabu, we set *tabu* $(e,\ s)$ to the current iteration counter plus the number of iterations *niter* along which the move will be non-admissible, *i.e.*, considered as a tabu move. Matrix *tabu* may then be used to keep track of the tabu status of every move.

Extended tabu lists. As discussed in the previous paragraph, the tabu attribute which is made active at each non-improving iteration has been defined as the pair $(e,\ s)$ (element e is prohibited to be reassigned again to cell C_s during a certain tabu tenure). In some situations, it may be desirable to increase the number of available moves that receive a tabu classification. This may be achieved either by increasing the tabu tenure or by changing the tabu restriction [10]. Several other applications of tabu search (see *e.g.* [12, 25]) have shown that frequently it may be interesting to turn tabu-active certain attributes that not only avoid the reversal move towards the original solution, but also avoid a great variety of other moves towards other solutions which resemble the original one.

TABLE I
Approaches for the tabu lists.

List type	Tabu attribute	Description
T_1	$(e,\ \cdot,\ s)$	element e is prohibited to be reassigned to cell C_s
T_2	$(e,\ \cdot,\ \cdot)$	element e cannot be reassigned to any other cell
T_3	$(\cdot,\ t,\ \cdot)$	no element assigned to C_t is allowed to be reassigned
T_4	$(\cdot,\ \cdot,\ s)$	no new element is allowed to be reassigned to cell C_s

Using this approach, we consider other more restrictive tabu attributes following a move (e, s, t), such as: (i) prohibiting element e from being reassigned to any other cell (tabu list type T_2); (ii) prohibiting any element assigned to cell C_t from being reassigned to any other cell (tabu list type T_3); (iii) prohibiting any element from any other cell to be reassigned to cell C_s (tabu list type T_4). Table I describes the different approaches for the tabu lists used in this work.

5. AN ITERATIVE MULTI-CRITERIA APPROACH

Multiple objectives as those presented in Section 2 may be dealt with by at least three different strategies. The first strategy, which is also the most direct one, consists in defining a weighted function $f = \sum_{i=1}^{9} w_i f_i$, with $w_i \geq 0$, $i = 1, \cdots, 9$, as in Hertz, Jaumard and Ribeiro [14]. However, one of the major difficulties with this approach is to assess weights taking into account both the relative importance of each objective and the heterogeneity of the various functions. Another drawback is the computing times required if any heuristic based on local search is used. In an exchange heuristic such as tabu search, the computations may often be speeded up by updating formulae which fastly compute from one iteration to the next the variations in the objective function associated which each move. However, the more the function is complex, the more expensive are the computing costs at each iteration, and hence the overall computing time. This behavior was observed in [14], where it was shown that no more than two objectives may be simultaneously considered if we want to have reasonable computing times.

Another way to obtain a multi-criteria approach is to define a hierarchy of functions. Typically, in any exchange heuristic, the best move is selected at each iteration. In practice, for problems such as cell formation in group technology, there are many ties at each iteration if only one objective is considered at a time. To circumvent this difficulty, a second objective can be used to break ties with respect to the first objective, and if necessary other objectives may be considered to break the remaining ties if any. Globally, the algorithm will look for a solution optimizing the second criterion, among all those with the best value for the primary criterion. A drawback of this approach is that it gives a large advantage to the first objective with respect to the other ones.

The third strategy is the one we investigate in detail in the remaining of this work. It consists in an iterative procedure, in which each objective is considered in turn while imposing threshold values on the other ones already examined. In other words, we first optimize the primary objective. We obtain a threshold for the value of this primary criterion by accepting some deterioration with respect to the best solution found. Secondary criteria are then optimized, under constraints on the value of the objectives already examined.

Let $f_1, f_2, \cdots, f_r$ be the sequence of objectives to be optimized, where f_1 is the primary objective, f_2 is the secondary one, and so forth. For a given solution s, feasible or not, let $f_l(s)$ be the value for solution s of the l-th objective, $l = 1, \cdots, r$. Our iterative multi-criteria approach consists in solving (by tabu search) each of a sequence of r problems. The problem (P_q) solved at stage $q = 1, \cdots, r$ of this scheme is:

$$(P_q) \quad \begin{cases} f_q^* = \text{minimum} & f_q(s) \\ \qquad \text{subject to} & f_l(s) \leq \bar{f}_l \qquad l = 1, \cdots, q-1 \\ & s \in S, \end{cases}$$

where $\bar{f}_l$ is a threshold on the value of objective f_l, computed *e.g.* as the maximum accepted deterioration from of the optimal value f_l^* of problem (P_l). The relative importance of each objective may then be controlled by increasing or decrasing the corresponding threshold.

The primary objective is dealt with directly by the tabu search implementation described in Sections 3 and 4. Secondary objectives are dealt with by a strategic oscillation scheme presented below. Strategic oscillation is one of the basic diversification approaches for tabu search. Glover, Taillard and de Werra [11] explain that the basic idea is to drive the search toward and away from selected boundaries of feasibility, either by manipulating the objective function (by the use of penalties or incentives) or by compelling the choice of moves that lead to specified directions.

For solving each problem (P_q), $q = 2, \cdots, r$, we use the same approach considered by Gendreau, Hertz and Laporte [4] for the vehicle routing problem. As long as we are developing an approximate scheme for the cell formation problem, the threshold constraints may be taken as soft constraints by means of penalties incorporated into the objective function. Let $\mu_l \geq 0$ be the penalty coefficient associated with the l-th objective, $l = 1, \cdots, q-1$.

The associated problem is then:

$$(P_q^{\text{penal}}) \quad \begin{cases} \text{minimize} & f_q(s) + \sum_{l=1}^{q-1} \mu_l \cdot \max\{0, f_l(s) - \bar{f}_l\} \\ \text{subject to} & s \in S. \end{cases}$$

Notice that for every $l \in \{1, \cdots, q-1\}$ such that the penalty coefficient μ_l is taken sufficiently large, constraint $f_l(s) \leqq \bar{f}_l$ will automatically be enforced. On the other hand, if the penalty coefficient μ_l is small for some $l \in \{1, \cdots, q-1\}$, the optimal solution of (P_q^{penal}) may violate the constraint associated with the corresponding term. The strategic oscillation scheme basically consists in periodically adjusting the penalty coefficients. The basic tabu search algorithm is applied to the penalty function:

$$f_{\text{penal}}(s) = f_q(s) + \sum_{l=1}^{q-1} \mu_l \cdot \max\{0, f_l(s) - \bar{f}_l\}.$$

At each iteration, a new solution $s' \in N(s)$ is obtained, where s is the current solution. If $f_{\text{penal}}(s')$ is less or equal than the best value already found for the penalty function, then the best (eventually unfeasible) solution s^* is updated, *i. e.* we set $s^* \leftarrow s'$. If the new solution s' is feasible (*i. e.*, if it satisfies all threshold constraints $f_l(s) \leqq \bar{f}_l$, $l = 1, \cdots, q-1$) and it improves the value of the best feasible solution found so far, then the best feasible solution s^*_{feas} is updated, *i. e.* we set $s^*_{\text{feas}} \leftarrow s'$. For every $l = 1, \cdots, q-1$ such that all solutions found during a series of a certain number of iterations (say, 10 iterations) of the tabu search algorithm violate constraint $f_l(s) \leqq \bar{f}_l$, the penalty coefficient μ_l is doubled. Contrarilly, the penalty coefficient μ_l is halved if the associated constraint is not violated by any solution found during this series of iterations. Gendreau, Hertz and Laporte [4] have shown that, with this type of rule, we may quickly reach appropriate values for the penalty coefficients μ_l, $l = 1, \cdots, q-1$, that produce a mix of feasible and unfeasible solutions. Obtaining unfeasible solutions is important since this helps in adding a diversification strategy to the search, leading from local optima. A procedural description of the overall strategic oscillation strategy for approximately solving each problem (P_q), integrated to the basic tabu search algorithm, is given in Figure 3.

6. APPLICATION TO A BI-CRITERIA PROBLEM

We present in this section some numerical results obtained from an application of the iterative multi-criteria approach proposed in Section 5. The strategic oscillation scheme was implemented in C and applied to a bi-criteria problem. All computational results reported below were obtained on a Sun Sparc 2 workstation.

The primary criterion is the minimization of the total intercell material handling cost. Cardinality constraints on the minimum and maximum number of machines and parts within each cell are taken as hard constraints. Let s be any feasible solution defined by cells $C_1, \cdots, C_n$. As defined in Section 2, let $x_i^k = 1$ if machine M_i is assigned to cell C_k, $x_i^k = 0$ otherwise. Analogously, let $y_j^k = 1$ if part P_j is assigned to cell C_k, $y_j^k = 0$ otherwise. As before, we denote by γ_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, p$, the material handling cost associated with the intercell move corresponding to taking part P_j from the cell where it is currently located and processing it by machine M_i located in another cell. The total intercell material handling cost corresponding to solution s is given by objective f_8, which was expressed in Section 2 as:

$$f_8(s) = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{p} \gamma_{ij} \, [x_i^k (1 - y_j^k) + y_j^k \, (1 - x_i^k)].$$

Accordingly, the problem solved by the tabu search algorithm in the first phase is then:

$$(P_1) \quad \begin{cases} f_8^* = \text{minimize} & f_8(s) \\ \quad\quad\;\; \text{subject to} & s \in S, \end{cases}$$

where S is the set of feasible solution satisfying the cardinality constraints. The primary objective deals with the optimization of a measure of the intercell workload. As the second objective, we attempt to somehow optimize the intracell load. A measure of the intracell load associated with solution s may be given by objective f_9, expressed in Section 2 as:

$$f_9(s) = \frac{1}{2} \max_{k=1, \cdots, n} \left| \sum_{i:M_i \in C_k} \sum_{j:P_j \in C_k} \gamma'_{ij} - \frac{1}{n} \sum_{k=1}^{n} \sum_{i:M_i \in C_k} \sum_{j:P_j \in C_k} \gamma'_{ij} \right|,$$

```
begin
  Initialize the short term memory function
  Generate the initial feasible solution s_0
  Set the number of iterations iter ← 1
  Set the initial values of the penalty coefficients μ_l, l = 1, ···, q − 1
  s, s*, s*_feas ← s_0
  while (maxmoves moves have not been made since the last improvement in f_q (s*_feas) or
f_penal(s*)) do
  begin
    if the number of iterations niter is a multiple of 10 then
    for all l = 1, ···, q − 1 do
    begin
      if constraint f_l(s) ≦ f̄_l was enforced by all the ten previous solutions, then set μ_l ← μ_l/2
      if constraint f_l(s) ≦ f̄_l was violated by all the ten previous solutions, then set μ_l ← 2 · μ_l
      end
    bestmovevalue ← ∞
    for (all candidate moves) do
      if (candidate move is admissible, i. e., if it does not belong to the tabu list) then
      begin
        Obtain the neighbor solution s̄ by applying candidate move to the current solution s
        movevalue ← f_penal (s̄) − f_penal (s)
        if (movevalue < bestmovevalue) then
        begin
          bestmovevalue ← movevalue
          s' ← s̄
        end
      end
    end
    Update the short term memory function
    if (f_penal(s') < f_penal (s*)) then s* ← s'
    if (s' is feasible and f_q (s') < f_q(s*_feas)) then s*_feas ← s'
    s ← s'
  end
end
```

Figure 3. – Strategic oscillation scheme for problem (P_q).

where γ'_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, p$, now represents the processing time of part P_j by machine M_i. A threshold $\bar{f}_8$ for the value of the primary objective is given by the maximum accepted deterioration from the optimal value f_8^* obtained as the solution of the problem (P_1) solved in the first phase, say *e.g.* $\bar{f}_8 = (11/10)\, f_8^*$. Then, the problem solved in the second phase by the strategic oscillation scheme integrated to the tabu search algorithm is:

$$(P_2) \quad \begin{cases} f_9^* = \text{minimize} & f_9(s) \\ \qquad \text{subject to} & f_8(s) \leqq \bar{f}_8 \\ & s \in S. \end{cases}$$

6.1. Description of the test problems

In order to evaluate the behavior and the effectiveness of the approach proposed in the previous section, we have generated fifteen test problems. Three classes of problems have been generated: (1) five problems with $m = 50$ machines and $p = 500$ parts, (2) five problems with $m = 100$ machines and $p = 500$ parts, and (3) five problems with $m = 100$ machines and $p = 1,000$ parts. For each problem we assume a number of ten production cells, *i. e.* $n = 10$.

The random problem generator consists in three phases. In the first phase, a structured matrix with 0-1 elements is generated. Each row of this matrix is associated with a machine, while each column is associated with a part. Each element a_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, p$, of this incidence matrix takes the value one if part P_j must be processed by machine M_i, zero otherwise. The matrix may be seen as formed by as many diagonal blocks as the number of cells (*i. e.*, ten in this case). The 0-1 elements of this matrix are generated accordingly with (1) a fixed percentage of 95% intra-block nonzero elements (*i. e.*, intra-cellular operations), and (2) a fixed percentage of 5% inter-block nonzero elements (*i. e.*, bottleneck operations). The second phase of the generator consists in a random permutation of the rows, followed by a random permutation of the columns, destroying the original block-diagonal structure of the machine-part incidence matrix. Finally, integer processing times γ_{ij} and material handling costs γ'_{ij} associated with the nonzero elements of the incidence matrix are randomly generated in the interval [1, 60].

The minimum number of machines or parts within each cell is fixed at one, *i. e.*, there should be a least one machine and one part per cell: $\underline{m} = \underline{p} = 1$. The maximum number of machines and parts per cell are taken, respectively, as $\overline{m} = m/n + m/5$ and $\overline{p} = p/n + p/25$ (where m is the total number of machines in the problem, p is the total number of parts in the problem, and $n = 10$ is the number of cells).

6.2. Computational results: First phase

We report here the computational results obtained with the application of the basic tabu search algorithm in the solution of the unconstrained problem (P_1) solved in the first phase of our approach.

We first studied the sensitivity of the partial neighborhood search (in terms of both the value of the best solution found and the computing times) with respect to the fraction of the neighborhood which is examined at each iteration of the tabu search algorithm. For this first experiment, the tabu tenure of each move was fixed at 20, while *maxmoves* was fixed at 50. Average results over the 15 test problems are reported in Table II.

Wee see from Table II that the algorithm is quite sensitive to the fraction of the neighborhood which is examined at each iteration. However, it may also be seen that, for larger neighborhood fractions, very large computing times are necessary for small gains in terms of solution quality. Fixing the neighborhood fraction at 20%, we have then investigated the behavior of the algorithm with respect to the stopping criterion *i. e.*, the value of *maxmoves*. The computational results are presented in Table III, from where we have decided to take $maxmoves = 150$.

Next, we have investigated the different types of tabu lists and the best tabu tenures for each of them. We have observed that the basic tabu list type T_1,

TABLE II

Sensitivity with respect to the neighborhood fraction.

Neighborhood fraction	Best value	Computing time (seconds)
1%	112,079.5	30.1
2%	106,231.9	41.1
5%	104,001.0	106.1
20%	102,395.8	401.3
40%	101,874.1	791.6

TABLE III

Sensitivity with respect to maxmoves.

Maxmoves	Best value	Computing time (seconds)
50	102,395.8	401.3
100	102,380.0	415.8
150	102,352.1	425.0
200	102,352.1	430.1

in which at each iteration we prohibit an element to be reassigned to the cell where it was before the corresponding move, is not very sensitive with respect to the tabu tenures. Let $niter^m$ and $niter^p$ be, respectively, the tabu tenures corresponding to moves associated to machines and parts. We observed that, for $niter^m$ and $niter^p$ ranging from 5 to 130, the best value of the objective function oscillated between 101,643.0 and 102,352.1 (a variation of less than 0.7%). We have chosen $niter^m = 105$ and $niter^p = 130$ as the tabu tenures which gave the best results within this range.

The use of the first type of tabu list T_1 pointed out to large values of the tabu tenures, e.g. $niter^m = 105$ and $niter^p = 130$ for problems of the size and type we studied. As expected, since the number of parts is much larger than that of machines, the most suitable tabu tenure for part moves is larger than the tabu tenure for machine moves. A similar study was performed for the other types of more restrictive tabu lists. The results are summarized in Table IV. For each type of tabu list (T_1, T_2, T_3, and T_4), we give the tabu tenures $niter^m$ and $niter^p$, the average value of the best solutions found for the 15 test problems, the average computing time, and the average number of iterations.

TABLE IV

Best results for each type of tabu list.

Tabu list type	$niter^m$	$niter^p$	Best value	Seconds	Iterations
T_1	105	130	101,643.0	252.5	1,035
T_2	15	15	102,105.4	191.3	905
T_3	20	25	97,541.9	251.2	951
T_4	45	45	87,959.8	167.8	1,595

We may see from Table IV that the use of the more restrictive tabu list of type T_4 had a major impact on the behavior of the algorithm, *i. e.*, in the quality of the best solution found. We may also observe that, as long as more restrictive lists are used, smaller tabu tenures may be used. Also interesting is that the best computing time is also obtained with that type of tabu list.

6.3. Computational results: Second phase

We now report the computational results obtained with the application of the strategic oscillation scheme in the solution of the second phase

TABLE V

Sensitivity with respect to the neighborhood fraction (strategic oscillation).

Neighborhood fraction	Best value	Computing time (seconds)
1%	9,925.8	10.3
2%	7,037.4	30.1
5%	6,144.6	95.1
20%	2,375.5	260.6
40%	2,988.3	569.7

constrained problem (P_2). The initial solution is taken as the best solution found in the first phase. A 10% deterioration with respect to the best value of the first objective is accepted, *i. e.*, we take the threshold $\bar{f}_1 = (11/10)\, f_1^*$.

As in the first phase, we first studied the sensitivity of the partial neighborhood search with respect to the fraction of the neighborhood which is examined at each iteration of the tabu search algorithm, fixing at 20 the tabu tenure of each move, while *maxmoves* was fixed at 150. Average results over the 15 test problems are reported in Table V.

We may observe an anomaly in the results presented in Table V: notice that when the neighborhood fraction was increased from 20% to 40%, the quality of the best solution deteriorated. Although this does not happen often, it may occur due to the heuristic nature of the tabu search algorithm. Again, as in the case of the computational results observed for phase 1, we may see that the algorithm is very sensitive to the fraction of the neighborhood which is examined at each iteration. Fixing the neighborhood fraction at 20%, we have then investigated the behavior of the algorithm which respect to the stopping criterion, *i. e.*, the value of *maxmoves*. From the computational results presented in Table VI, we choose *maxmoves* = 450.

We also investigated the different types of tabu lists and the best tabu tenures for each of them. We report in Table VII the results observed with the basic tabu list type T_1, together with those obtained with the use of each of the more restrictive tabu list types T_2, T_3, and T_4. For each type of tabu list we give the tabu tenures $niter^m$ and $niter^p$, the average value of the best solutions found for the 15 test problems, the average computing time, and the average number of iterations.

We may see from the final results in Table VII that the use of the more restrictive tabu list of type T_4 had a major impact on the behavior of the algorithm, *i. e.*, in the quality of the best solution found, confirming the

TABLE VI

Sensitivity with respect to maxmoves (strategic oscillation).

Maxmoves	Best value	Computing time (seconds)
100	2,489.8	208.3
150	2,375.5	260.6
200	2,185.4	306.5
250	2,025.0	446.6
300	2,020.0	475.0
350	2,019.3	476.3
400	2,019.1	490.3
450	2,018.4	504.9
500	2,018.4	518.3

TABLE VII

Results for the more restrictive tabu lists (strategic oscillation).

Tabu list type	$niter^m$	$niter^p$	Best value	Seconds	Iterations
T_1	20	40	2,307.1	525.9	1042
T_2	10	10	2,433.3	685.6	1357
T_3	10	20	1,440.5	317.1	985
T_4	30	30	1,127.6	710.0	2359

results obtained in the first phase. We have also investigated the behavior of the strategic oscillation scheme with respect to the number of iterations in each series. In fact, the value of the best solution found was further reduced from 1,127.6 to 1,109.0 when this number of iterations was reduced from 10 to 5 (*i. e.*, the number of iterations to be executed prior to halving or doubling the penalty coefficients). Moreover, we may see that, by appropriately tuning the different parameters of the tabu search algorithm, we have been able to considerably improve the best solution.

7. CONCLUSIONS

We have studied in this paper the cell formation problem in group technology with multiple objectives. We have shown that different objectives must often be simultaneously considered if one wants to find a good solution satisfying multiple technological and economical criteria. A multiphase, iterative approach was proposed for solving the multi-criteria cell formation problem. The approach basically consists in solving a sequence of single-objective, multi-constrained problems, in which each objective is taken and optimized in turn, following their order of relative importance. The other objectives, previously optimized, are dealt with by means of threshold constraints generated from their optimal values obtained from the solutions of the problems previously solved.

We have used tabu search and a strategic oscillation scheme for the solution of each problem. Although many other approximate algorithms have already been proposed in the literature for the solution of cell formation problems in group technology, with their large number of variants and extensions, to the best of our knowledge this seems to be the first application of the tabu search metaheuristic in this area. We have also shown that the behavior of the tabu search approach may be considerably improved (both in terms of solution quality and computing time) by appropriately tuning the different parameters of the algorithm.

Computational results for a bi-criteria problem have been reported. These results show the effectiveness of the tabu search approach for solving cell formation problems. Moreover, they also show that the multi-phase, iterative approach proposed for solving the multi-criteria cell formation problem may become a useful tool for other multi-criteria problems. The idea of considering each objective in turn, in their order of importance, makes possible many types of analyses and applications.

As an example of a very useful type of analysis, the relative importance of each objective may be controlled by either increasing or decreasing each threshold. We illustrate this type of analysis in Figure 4, where the best value (average over the 15 test problems) found for the secondary objective of the bi-criteria problem discussed in Section 6 is plotted as a function of the maximum accepted level of deterioration (*i. e.*, the threshold) accepted for the primary objective with respect to its best known value. Despite of small anomalies observed for a few threshold values, which are due to the heuristic nature of the tabu search algorithm, these results indicate that the best solutions found by the strategic oscillation algorithm globally converge to those of the unconstrained problem when the maximum accepted level of deterioration increases. Although the choice of the appropriate thresholds is far from being innocuous, this difficulty may be partly circumvented by performing this type of sensitivity analysis.

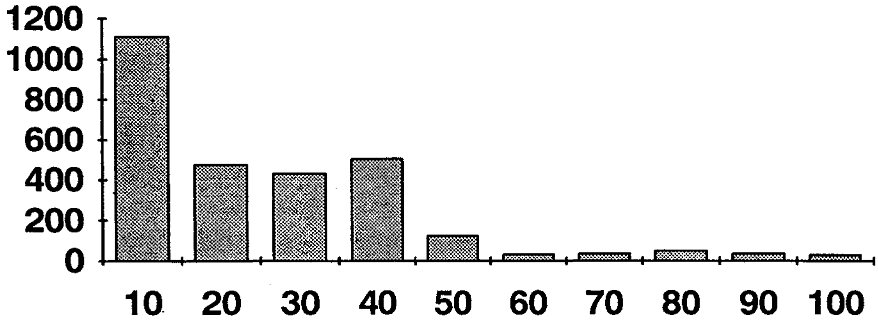

Figure 4. – Sensitivity of the secondary criterion with respect to the maximum accepted level of deterioration (in percent) of the primary criterion.

ACKNOWLEDGEMENTS

Work of the first author has been supported by an NSERC (Natural Sciences and Engineering Research Council of Canada) International Fellowship. Work of the second author has been supported by NSERC grant GP0036426 and FCAR (Fonds pour la formation de Chercheurs et l'Aide à la Recherche) grants 90NC0305 and 92EQ1048. Work of the third author was partially supported by the Brazilian National Research Council for Scientific and Technological Development, under research grant 302281/85-1. This work was done as part of the joint research project between the Department of Computer Science of the Catholic University of Rio de Janeiro and the Department of Applied Mathematics of the *École Polytechnique de Montréal*, in the framework of the cooperation agreement between the National Council for Scientific and Technological Development (Brazil) and the Natural Sciences and Engineering Research Council (Canada).

REFERENCES

1. S. G. de Amorim, J.-P. Barthelémy, C. C. Ribeiro, Clustering and Clique Partitioning: Simulated Anealing and Tabu Search Approaches, *Journal of Classification*, 1992, *9*, p. 17-41.
2. C. Friden, A. Hertz, D. de Werra, STABULUS: A Technique for Finding Stable Sets in Large Graphs with Tabu Search, *Computing*, 1989, *42*, p. 35-44.
3. O. Garza, T. L. Smunt, Countering the Negative Impact of Intercell Flow in Cellular Manufacturing, *Journal of Operations Management*, 1991, *10*, p. 92-117.
4. M. Gendreau, A. Hertz, G. Laporte, *A Tabu Search Heuristic for the Vehicle Routing Problem*, Publication ♯777, Centre de Recherche sur les Transports, Université de Montréal, 1991.
5. F. Glover, Future Paths for Integer Programming and Links with Artificial Intelligence, *Computers and Operations Research*, 1986, *13*, p. 533-549.
6. F. Glover, Tabu Search – Part I, *ORSA Journal on Computing*, 1989, *1*, p. 190-206.
7. F. Glover, Tabu Search – Part II, *ORSA Journal on Computing*, 1990, *2*, p. 4-32.
8. F. Glover, Tabu Search: A Tutorial, *Interfaces*, 1990, *20*, p. 74-94.
9. F. Glover, H. J. Greenberg, New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence, *European Journal of Operational Research*, 1989, *39*, p. 119-130.

10. F. Glover, M. Laguna, Tabu Search, in *Modern Heuristic Techniques for Combinatorial Problems*, C. R. Reeves Ed., Blackwell Scientific Publications, London, 1993, p. 70-150.

11. F. Glover, E. Taillard, D. de Werra, A User's Guide to Tabu Search, *Annals of Operations Research*, 1993, *41*, p. 3-28.

12. P. Hansen, E. L. Pedrosa Filho, C. C. Ribeiro, Location and Sizing of Off-Shore Platforms for Oil Exploration, *European Journal of Operational Research*, 1992, *58*, p. 202-214.

13. P. Hansen, M. V. Poggi de Aragão, C. C. Ribeiro, Boolean Query Optimization and the 0-1 Hyperbolic Sum Problem, *Annals of Mathematics and Artificial Intelligence*, 1990, *1*, p. 97-109.

14. A. Hertz, B Jaumard, C. C. Ribeiro, *A Tabu Search Approach to Cell Formation Problems in Group Technology*, TIMS XXX-SOBRAPO XXIII Joint International Meeting, Program Book, Rio de Janeiro, 1991, p. 19.

15. A. Hertz, B. Jaumard, C. C. Ribeiro, A Graph Theory Approach to Subcontracting, Machine Duplication and Intercell Moves in Cellular Manufacturing, *Discrete Applied Mathematics*, 1994, *50*, p. 255-265.

16. A. Hertz , D. de Werra, Using Tabu Search Techniques for Graph Coloring, *Computing*, 1987, *29*, p. 345-351.

17. A. Hertz, D. de Werra, The Tabu Search Metaheuristic: How We Used It, *Annals of Mathematics and Artificial Intelligence*, 1990, *1*, p. 111–121.

18. B Jaumard, C. C. Ribeiro, *A Column Generation Approach to Cell Formation Problems in Cellular Manufacturing*, Working paper, February 1993.

19. N. Hyer, The Potential of Group Technology for U.S. Manufacturing, *Journal of Operations Management*, 1984, *4*, p. 183-202.

20. N. Hyer, U. Wemmerlöv, Group Technology in the U.S. Manufacturing Industry: A Survey of Current Practices, *International Journal of Production Research*, 1989, *27*, p. 1287-1304.

21. J. R. King, V. Nakornchai, Machine-Component Group Formation in Group Technology: Review and Extension, *International Journal of Production Research*, 1982, *20*, p. 117-133.

22. K. R. Kumar, A. Vannelli, Strategic Subcontracting for Efficient Disaggregated Manufacturing, *International Journal of Production Research*, 1983, *25*, p. 1715-1728.

23. A. Kusiak, S. S. Heragu, Group Technology, *Computers in Industry*, 1987, *9*, p. 83-91.

24. M. Laguna, *Tabu Search Primer*, Research report, University of Colorado at Boulder, Graduate School of Business and Administration, Boulder, 1992.

25. M. Laguna, J. W. Barnes, F. Glover, Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs and Times Using Tabu Search, working paper, 1990.

26. R. Logendran, A Workload Based Model for Minimizing Total Intercell and Intracell Moves in Cellular Manufacturing, *International Journal of Production Research*, 1990, *28*, p. 913-925.

27. R. Logendran, A Model for Duplicating Bottleneck Machines in the Presence of Budgetary Limitations in Cellular Manufacturing, *International Journal of Production Research*, 1992, *30*, p. 683-694.

28. J. Miltenburg, W. Zhang, A Comparative Evaluation of Nine Well-Known Algorithms for Solving the Cell Formation Problem in Group Technology, *Journal of Operations Management*, 1991, *10*, p. 44-72.

29. P. A. Miscimara, *The NLRB and Managerial Discretion: Plant Closings, Relocations, Subcontracting, and Automation*, University of Pennsylvania, Philadelphia, 1983.
30. J. S. Morris, R. J. Tersine, A Simulation Analysis of Factors Influencing the Attractiveness of Group Technology Cellular Layouts, *Management Science*, 1990, *36*, p. 1567-1578.
31. S. C. Porto, C. C. Ribeiro, *A Tabu Search Approach to Task Scheduling on Heterogeneous Processors under Precedence Constraints*, Monografias em Ciência da Computação MCC-03/93, Catholic University of Rio de Janeiro, Department of Computer Science, January 1993, to appear in the *International Journal of High Speed Computing.*
32. R. Rajagopalan, J. L. Batra, Design of Cellular Production Systems: A Graph-Theoretic Approach, *International Journal of Production Research*, 1982, *13*, p. 567-579.
33. D. Rajamani, N. Singh, Y. P. Aneja, Integrated Design of Cellular Manufacturing Systems in the Presence of Alternative Process Plans, *International Journal of Production Research*, 1990, *28*, p. 1541-1554.
34. D. Rajamani, N. Singh, Y. P. Aneja, Selection of Parts and Machines for Cellularization: A Mathematical Programming Approach, *European Journal of Operational Research*, 1992, *62*, p. 47-54.
35. H. Seifoddini, Duplication Process in Machine Cells Formation in Group Technology, *IIE Transactions*, 1989, *21*, p. 382-388.
36. H. Seifoddini, P. M. Wolfe, Application of the Similarity Coefficient Method in Group Technology, *IIE Transactions*, 1986, *18*, p. 271–277.
37. S. M. Shafer, D. F. Rogers, A Goal Programming Approach to the Cell Formation Problem, *Journal of Operations Management*, 1991, *10*, p. 28-43.
38. J. Skorin-Kapov, Tabu Search Applied to the Quadratic Assignment Problem, *ORSA Journal on Computing*, 1990, *2*, p. 33-45.
39. J. Skorin-Kapov, A. J. Vakharia, *Scheduling a Flow-Line Manufacturing Cell: A Tabu Search Approach*, Research report, State University of New York at Stony Brook, New York, 1992.
40. A. J. Vakharia, Y.-L. Chang, A Simulated Annealing Approach to Scheduling a Manufacturing Cell, *Naval Research Logistics Quarterly*, 1990, *37*, p. 559-577.
41. A. Vannelli, K. R. Kumar, A Method for Finding Minimal Bottleneck Cells for Grouping Part-Machine Families, *International Journal of Production Research*, 1986, *24*, p. 387-400.
42. A. Vannelli, K. R. Kumar, Minimal Bottleneck Cell Approach for Generating Part-Machines Families in Cellular Manufacturing, in *Computer Integrated Manufacturing* I. B. Turkara *et al.*, Eds., Springer-Verlag, Berlin, 1988.
43. V. Venugopal and T. T. Narendran, A Genetic Algorithm Approach to the Machine-Component Grouping Problem with Multiple Objectives, *Computers and Industrial Engineering*, 1992, *22*, p. 469-480.
44. V. Venugopal, T. T. Narendran, Cell Formation in Manufacturing Systems through Simulated Annealing: An Experimental Evaluation, *Journal of the Operational Research Society*, 1992.
45. J. C. Wei, N. Gaither, A Capacity Constrained Multiobjective Cell Formation Method, *Journal of Manufacturing Systems*, 1990, *9*, p. 222-232.
46. U. Wemmerlöv, N. L. Hyer, Cellular Manufacturing in the U.S. Industry: A Survey of Users, *International Journal of Production Research*, 1989, *27*, p. 1511–1530.
47. M. Widmer, A. Hertz, A New Approach for Solving the Flow Shop Sequencing Problem, *European Journal of Operational Research*, 1989, *41*, p. 186-193.