Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 151 (2003) 434–446
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

A simulated annealing approach for manufacturing cell formation with multiple identical machines

Ana R. Xambre *, Pedro M. Vilarinho

Departamento de Economia, Gestão e Engenharia Industrial, Universidade de Aveiro, Campus de Santiago, 3810-193 Aveiro, Portugal

Abstract

One of the crucial steps in the design of a cellular manufacturing system is the cell formation problem, which involves grouping the parts into part families and machines into manufacturing cells, so that parts with similar processing requirements are manufactured within the same cell. When modelling this problem it is usually assumed that the specific machine in which each operation is carried out is known.

In a cellular manufacturing system where multiple, functionally identical, machines are available, a new degree of freedom can be introduced into this problem—the allocation of the operations to specific machines.

In this paper a mathematical programming model for the cell formation problem with multiple identical machines, which minimises the intercellular flow, is presented. Due to the combinatorial nature of this problem a simulated annealing algorithm was developed to solve it.

Keywords: Manufacturing; Cellular manufacturing systems; Simulated annealing

1. Introduction

In a process oriented production system functionally identical machines are grouped together in departments, thus parts requiring processing by more than one machine type need to travel around visiting the relevant departments for its manufacturing. This type of production systems are very flexible and favour a high utilisation rate of the resources, but also induce a significant amount of material handling, high work-in-process inventories, long throughput times and are difficult to control.

In product oriented manufacturing systems machines are placed in production lines dedicated to the manufacturing of a specific product. In this type of systems material handling costs are low, work-in-process inventories are low, throughput times are short, production control is easy, but they are not flexible. Production lines are the most effective and efficient way of organising the manufacturing resources, but they can only be used for a low variety and high volume product mix.

Cellular manufacturing has emerged in the last two decades as an innovative manufacturing strategy that collects the advantages of both product and process oriented systems for a high

* Corresponding author.
E-mail addresses: raquelx@egi.ua.pt (A.R. Xambre), pvil@mail.ua.pt (P.M. Vilarinho).

0377-2217/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0377-2217(02)00836-6

variety and medium volume product mix (Burbidge, 1992). In a cellular manufacturing system, functionally diverse machines are grouped in cells, each of which is dedicated to the production of a part family, composed of different parts with similar processing requirements.

One of the crucial steps in the design of a cellular manufacturing system is the cell formation problem, which involves grouping the parts into part families and machines into manufacturing cells, so that parts with similar processing requirements are completely manufactured within the same cell. However, completely independent cells are usually difficult to generate in practice, since some parts may need to be processed in more than one machine cell, leading to the existence of some intercellular flow. Thus, the major goal for the cell formation problem is to find the grouping of machines into cells that minimises intercellular flow.

The cell formation problem has been the topic of a considerable amount of research, due to its importance in the design of cellular manufacturing systems. Burbidge, 1963 production flow analysis was one of the first procedures to solve the cell formation problem, although not in an algorithmic approach. His procedure uses the machine-part incidence matrix (a binary matrix used to indicate whether a machine is used to process a part or not) and rearranges it into a block diagonal form (BDF), in which blocks of machine-part combinations are grouped into cells along the diagonal of the matrix. The elements in the BDF matrix that are outside the diagonal cells are termed exceptional elements and denote the existence of intercellular flow. In the procedure proposed by Burbridge, the exceptional elements are then dealt with by using methods such as machine duplication, subcontracting or route change.

A few methods only attempt to find out the part families, yielding only a partial solution to the cell formation problem, because upon identification of the part families, the machines required to completely process the parts within a cell need to be identified. Generally these methods are based on classification and coding schemes and group the parts according to its process similarities by determining similarity coefficients (see, for example, Tam, 1990; Jeon et al., 1998). A p-median formulation to form part families is given by Kusiak, 1985.

Other approaches try to determine not only the part families, but also the machine groups. The majority of these methods are based on the machine-part incidence matrix and can be divided into hierarchical and non-hierarchical clustering (Shargal et al., 1995; Seifoddini and Hsu, 1994; Srinivasan, 1994), mathematical programming and graph theoretic methods (Deutsch et al., 1998; Atmani et al., 1995; Adil et al., 1993; Boctor, 1991) and novel methods like simulated annealing, genetic algorithms and neural networks (Su and Hsu, 1998; Adil et al., 1997; Chen et al., 1995; Venugopal and Narendran, 1994). Singh, 1993, Selim et al., 1998 and Venugopal, 1999 present in-depth reviews of the different methods for the cell formation problem.

A major drawback of the methods that use the machine-part incidence matrix is that they do not allow an accurate calculation of the intercellular flow, because they do not take into account neither the operation sequence of the parts nor the flow volume between the operations. A few methods appeared recently that use either operation sequence or flow matrix information to solve the cell formation problem (for example, Sofianopoulou, 1997; Del Valle et al., 1994; Okogbaa et al., 1992).

Most models for the cell formation problem assume that the machine in which each operation will be carried out was previously defined. However, in most real cellular manufacturing systems multiple, functionally identical, machines are available, allowing for the operation allocation to be seen as another degree of freedom to the problem. In fact, switching the machine in which an operation is performed can foster a reduction in the intercellular flow. Wu and Salvendy, 1999 reviewed some of the existing approaches to solve the manufacturing cell formation problem with multiple identical machines (i.e., the problem that simultaneously tackles the traditional cell formation and the operations allocation problems) and presented a graphical model together with an efficient merging-and-breaking heuristic to solve the problem.

The model and heuristic presented in the following sections solve the cell formation problem with multiple identical machines, the goal being the minimisation of the intercellular flow, by explicitly considering the flow volume between the operations and machine capacity requirements. The approach suggested differs from previous work in two aspects. First, it allows for both operations and machines to switch cells, after an initial solution has been set, thus exploring a higher number of potential solutions. Second, it takes into account the possibility of an operation being split between two identical machines (in practice this means that two different batches of the same part can take different routes). Operation splitting can occur in real world systems when a set of identical machines is used near full capacity or when two or more functionally identical machines need to be placed in the same cell to produce large batches of similar parts.

2. Mathematical programming model

The proposed model for the manufacturing cell formation problem with multiple identical machines is based on the following set of basic assumptions:

(i) the production throughput is stable over time,
(ii) the overall production throughput can be adequately represented by a limited set of parts produced in predetermined quantities within a given planning horizon,
(iii) the operation sequence for each part is known and, hence, the flow between each pair of operations (i, j), f_{ij}, is known, and represents the total number of material handling trips between operations i and j required to move the volume of parts that use these operations consecutively,
(iv) each of the O operations can be performed on different parts,
(v) the production will be processed through equipment classified into a set of machine types $(t = 1, \ldots, T)$,
(vi) each operation can be performed on a set of functional identical machines $(m = 1, \ldots, M_t)$,
(vii) the usage rate of each operation i is expressed as the percentage of the capacity available on each machine of type t and is denoted by α_{it},
(viii) the maximum number of machines allowed per cell $(c = 1, \ldots, C)$ is MC. It should be noted that both upper bounds on the number of cells (C) and on the number of machines per cell (MC) are defined by the decision-maker.

Denoting the controllable variables as follows,

$$x_{mtc} = \begin{cases} 1, & \text{if the } m\text{th machine of type } t \text{ is allocated to cell } c, \\ 0, & \text{otherwise,} \end{cases}$$

y_{imt} = percentage of operation i processed on the mth machine of type t,

the cell formation problem with multiple identical machines can be modelled as follows:

$$\text{Min } Z = \sum_{i=1}^{O} \sum_{j=1}^{O} \sum_{t,u=1}^{T} \sum_{m,n=1}^{M_t} f_{ij} \cdot y_{imt} \cdot y_{jnu} \cdot \left(1 - \sum_{c=1}^{C} x_{mtc} \cdot x_{nuc}\right) \tag{1}$$

subject to

$$\sum_{i=1}^{O} y_{imt} \cdot \alpha_{it} \leqslant 1 \quad (m = 1, \ldots, M_t;\ t = 1, \ldots, T), \tag{2}$$

$$\sum_{t=1}^{T} \sum_{m=1}^{M_t} x_{mtc} \leqslant \text{MC} \quad (c = 1, \ldots, C), \tag{3}$$

$$\sum_{m=1}^{M_t} y_{imt} = 1 \quad (i = 1, \ldots, O;\ t = 1, \ldots, T), \tag{4}$$

$$\sum_{c=1}^{C} x_{mtc} = 1 \quad (m = 1, \ldots, M_t;\ t = 1, \ldots, T), \tag{5}$$

$$x_{mtc} \in \{0, 1\}, \qquad y_{imt} \in [0, 1]. \tag{6}$$

The objective function (1) minimises the intercellular flow. The constraints can be interpreted as follows:

(2) constraints insuring that the capacity of each machine is not exceeded,
(3) constraints limiting the number of machines per cell,

(4) constraints insuring that each operation is totally allocated to one or more machines,
(5) constraints imposing that each machine is allocated to one and only one cell,
(6) constraints defining the domain of each variable.

The proposed model is unusually non-linear and it cannot be solved to optimality even for small instances of the problem. Thus, a simulated annealing procedure, presented in the next sections, was developed to tackle the problem.

3. Simulated annealing for the cell formation problem with multiple identical machines

3.1. General simulated annealing algorithm

Kirkpatrick et al. (1983) initially presented the simulated annealing algorithm, which attempts to solve hard combinatorial optimisation problems through controlled randomisation. Since then the algorithm has been applied to many optimisation problems in a wide variety of areas, including cell formation problems (Su and Hsu, 1998; Adil et al., 1997; Sofianopoulou, 1997; Chen et al., 1995).

The pseudo-code for the general procedure for implementing the simulated annealing algorithm is presented in Table 1. The algorithm evolves from an initial solution S_0 for the problem. In the inner cycle of the algorithm, repeated while $n \leqslant L$, a neighbouring solution S_n of the current solution S is generated. If S_n is better than S ($\Delta \leqslant 0$) then the generated solution replaces the current solution, otherwise the solution is accepted with a certain probability ($p = \mathrm{e}^{-\Delta/T}$). The value of the temperature T decreases in each iteration of the outer cycle of the algorithm, which diminishes the probability of accepting as current solution worst solutions. Obviously, during the algorithm the best solution found (S^*) is always kept and the generation of neighbouring solutions obliges that two consecutive solutions must be different ($S_n \neq S_{n-1}$).

The most important characteristic of this algorithm is the possibility of accepting worst solutions, which can allow it to escape from local minima. Nonetheless, the performance of the algorithm depends on the definition of several control parameters:

(i) The initial temperature (T_0) should be high enough that in the first iteration of the algorithm the probability of accepting worst solutions is, at least, of 80% (Kirkpatrick et al., 1983).
(ii) The most commonly used temperature reducing function is geometric: $T_i = a_i \cdot T_{i-1}$ ($a_i < 1$ and constant). Typically, $0.7 \leqslant a_i \leqslant 0.95$.
(iii) The length of each temperature level (L) determines the number of solutions generated at a certain temperature (T).
(iv) The stopping criterion defines when the system has attained a desired energy level.

Table 1
Simulated annealing algorithm in pseudo-code

Select an initial temperature $T_0 > 0$;
Select an initial solution, S_0, and make it the current solution, S, and the current best solution, S^*;
repeat
  set repetition counter $n = 1$;
  repeat
    generates solution S_n in the neighbourhood of S;
    calculate $\Delta = f(S_n) - f(S)$;
    if ($\Delta \leqslant 0$) *then* $S = S_n$;
      else $S = S_n$ with the probability of $p = \mathrm{e}^{-\Delta/T}$;
    if ($f(S_n) < f(S^*)$) *then* $S^* = S_n$;
    $n = n + 1$;
  until $n >$ number of repetitions allowed at each temperature level (L);
  reduce the temperature T;
until stop criterion is true.

Some of the most common criteria are based on

- the total number of solutions generated;
- the temperature at which the desired energy level is attained (freezing temperature);
- the acceptance ratio (ratio between the number of solutions accepted and the number of solutions generated).

Naturally each of these control parameters must be refined according to the specific problem on hand. Two other important issues that need to be defined when adapting this general algorithm to a specific problem are the procedures to generate both the initial solution and the neighbouring solutions. The details of the proposed implementation of the simulated annealing algorithm to the cell formation problem with multiple identical machines are presented in the next section.

3.2. Implementation of the proposed simulated annealing algorithm

The proposed simulated annealing procedure for the cell formation problem with multiple identical machines requires the following data obtained from the production process details and from the demand over the planning horizon:

(i) interoperation flow matrix—derived from the operation sequence and demand for each part;
(ii) type of machine required by each operation;
(iii) usage rate of the machine type per operation—the ratio between the processing time to carry out the operation, in all the parts that require it, and the time capacity of a machine of the type involved, over the planning horizon.

The decision-maker also needs to define the maximum number of machines allowed per cell (MC) and maximum number of cells (C). These parameters can change according to the decision-maker strategy in regard to the specific problem he is trying to solve and, by changing them, he can generate different solutions to the problem.

There are two stages involved in obtaining the initial solution. In the first stage, operations are allocated to machines by decreasing order of their usage rate. In the second stage machines are grouped into cells taking into account the operations they process and the flow between the operations. The machine grouping procedure starts by determining a set of core machines (one for each cell) that will support the rest of the machine grouping. Note that the notation used in this section is the same that was used in Section 2.

(i) Operations allocation

For each machine type, $t = 1, \dots, T$:

Step 1: Calculate the number of units required: $M_t = \lceil \sum_{i=1}^{O} \alpha_{it} \rceil$;
Step 2: Sort the operations by decreasing order of α_{it};
Step 3: Following the order defined on the previous step, allocate the operations sequentially to the machine that, among those with enough capacity to process the operation, has more capacity available;
Step 4: If one operation cannot be allocated because there is not a machine available with enough usable capacity to process it, the operation is split and allocated to two, or more machines, starting with the machine which has more capacity available.

(ii) Machine grouping

Step 5: Choose the machine with the highest total flow with the other machines and assign it to a cell;
Step 6: Choose the machine with the lowest flow with the machines already assigned to cells and assign it to a new cell. In case of a tie choose, among the machines not yet assigned to cells, the one with the highest total flow with all machines;
Step 7: Repeat the previous step until all the C cells have one core machine assigned to them;
Step 8: Each of the remaining machines is assigned to the cell with which it has the highest average flow. The average flow between a machine and a cell is defined as

the ratio between the flow that goes through the machine to all the machines assigned to the cell and the number of machines in the cell. In case of a tie a machine is chosen randomly.

The procedure to generate neighbouring solutions selects operations that generate intercellular flow and tries to change them to another cell, attempting, in this way, to reduce the overall intercellular flow. This step is repeated until it is no longer possible to switch these operations to another cell, either because the machines of the relevant type placed in the other cells have no capacity available or because there are no machines of the requested type in any other cell. When this situation is reached, the procedure attempts to change the machines that generate intercellular flow to different cell. The steps involved in this procedure are as follows:

Step 1: Select the pair of operations that generate the highest intercellular flow;

Step 2: Select the operation, from the pair selected in the previous step, which is allocated to the machine with less available capacity;

Step 3: Select the cell containing the machines with which the selected operation has the highest flow. If, in this cell, there is a machine of the type required by the operation and with enough capacity to process it, change the operation to that cell. If there is more than one machine that meets these requirements, choose the one with more available capacity. If it is not possible to carry out this exchange, repeat this step with the remaining operation from Step 2;

Step 4: Until a neighbouring solution is found, repeat Steps 2 and 3, selecting the pair of operations in Step 2 by decreasing order of the intercellular flow it generates;

Step 5: If all the pairs of operations that generate intercellular flow have been tested and, even though it was not possible to generate a neighbouring solution go to Step 6;

Step 6: Select the machine that generates the higher intercellular flow;

Step 7: Change the machine selected in the previous step to the cell containing the machines with which the selected machine has the highest flow, if this change does not violate the maximum number of machines per cell (MC) constraint;

Step 8: If Step 7 fails, repeat it until a neighbouring solution is found selecting the machine by decreasing order of the flow it generates;

Step 9: If all the machines that generate intercellular flow have been tested and, even though it was not possible to generate a neighbouring solution go to Step 10;

Step 10: Generate a 'neighbouring solution' applying the same procedure used to generate the initial solution, but by choosing randomly the first core machine.

The solutions are evaluated using the flow inefficiency criteria defined by Heragu and Gupta, 1994, which is the ratio between the overall intercellular flow and the total flow.

The following control parameters were used to implement the algorithm:

(i) Initial temperature (T_0): The computational experience showed that the flow inefficiency never changed by more than 10% between two neighbouring solutions. So, for an initial temperature of 50 it is guaranteed that at least 80% of the inferior solutions are accepted.

(ii) Temperature reduction function: The geometric function with a temperature reduction factor of 0.9 ($T_i = 0.9T_{i-1}$) was used.

(iii) The length of each temperature level (L): A dominant factor on the computational effort associated to the solution of the problem is the number of intercellular flows, which is never greater than the number of operations. So, in order to restrict the computational effort to the first order of the dominant factor, the number of solutions searched at each temperature level was set to $K \cdot O$, where K is a user defined constant ($K = 1$ is the value suggested by default).

(iv) Stopping criteria: Two alternative criteria were set. In the first one, a freezing temperature of 10 is set, which means that 16 temperature levels are used ($T_0 \cdot a_i^{15} = 50(0.9^{15}) = 10.29$). In the second one, it is admitted that, if in five consecutive temperature levels 85% of the generated solutions are rejected, then the probability of replacing the best solution found is very small and the procedure is then terminated.

The procedure just described does not form the part families. So, after a good solution is found it is necessary to allocate the existing parts to the cells. The parts are assigned to cells sequentially by decreasing order on the number of operations performed on them. Each part is assigned to the cell where the largest number of its operations is processed. In case of a tie the part is allocated to the cell that contains fewer parts.

4. Numerical illustration

In this section, a numerical example to demonstrate the proposed simulated annealing procedure is presented. In this example four machine types ($T = 4$) are required to process sixteen operations ($O = 16$). A maximum of four machines per cell ($\mathrm{MC} = 4$) and a maximum of three cells ($C = 3$) were considered. The interoperation flow matrix is shown in Table 2. In Table 3, it is shown, for each operation, the machine type involved in processing it and the respective usage rate.

Table 3
Machine type and usage rate for the operations

Operation	Usage rate α_{it} (%)	Machine type t
1	22.3	4
2	63.8	2
3	18.5	4
4	67.9	3
5	68.9	1
6	38.5	2
7	38.1	2
8	74.8	3
9	66.5	1
10	10.9	2
11	53.2	4
12	66.4	1
13	26.9	4
14	69.7	2
15	16.4	3
16	39.0	3

To obtain an initial solution the procedure starts by allocating the operations to the machines. Machine type 3, for example, performs operations 4, 8, 15 and 16. The percentage of capacity required by these operations is 198.1% ($67.9 + 74.8 + 16.4 + 39$), which means that two units of

Table 2
Interoperation flow matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1								112								
2					224											
3														123		
4											137					
5												190				
6			182													
7								227								
8												56				
9				113												
10							222									
11						85										
12													101			
13					209											
14									237							
15										59						
16									246							

machine type 3 are required. For machine types 1, 2 and 4 the number of units required are 3, 3 and 2, respectively. Proceeding with machine type 3 as an example, operation 8 has the highest usage rate (74.8), so it is allocated to unit 1. Operation 4 has the second highest usage rate (67.9), so it is allocated to unit 2. The next operation to be allocated is number 16, with a usage rate of 39.0. However, the remaining capacity in unit 1 is 26.2% and in unit 2 is 32.1%. This means that operation 16 has to be split between units 1 and 2, filling first the available capacity in unit 2 (because this unit has more capacity available than unit 1). Finally, operation 15 is allocated to unit 1 (the only one with capacity available).

The final operations allocation, for all machine types, is shown in Table 4.

The machine grouping procedure requires an intermachine flow matrix, shown in Table 5, which is calculated using the interoperation flow matrix and the allocation done in the previous stage. In Table 5, row and column headings A,B represent unit B of machine type A, each cell accounts for the flow between the pair of corresponding operations and the last column (Total) indicates the total flow that goes through each machine. It should also be noticed that because of the split of operation 16, the corresponding flow was proportionally divided between machines 3,1 and 3,2 (accordingly to the percentage of this operation that each machine processes).

The core machine for the first cell (cell 1) is the one with the highest flow with every other machine (in the example, unit 2 of machine type 4). The machines with the lowest flow with this machine, and thus candidates to be the core machine in cell 2, are 1,2; 2,2; 3,2 and 4,1; each one with zero flow with machine 4,2. Since there is a tie, the machine with the highest flow with all machines is chosen (unit 2 of machine type 1). Using a similar reasoning, unit 2 of machine type 2 is selected as the core machine for cell 3.

Once the core machine for each cell has been selected, the remaining machines are allocated to the cell with which they have their highest average flow. In Table 6 it is showed the average flow between each of the non-allocated machines and the cells. Note that, in this case this average flow is equal to the corresponding absolute flow, because each cell still has only one machine assigned to it.

Machine 2 of machine type 3 has the highest average flow with cell 2, so it is allocated to that cell. To determine the second non-allocated machine to be assigned to a cell the figures in Table 5 need to be recomputed, and the new values are shown in Table 7. As a result, machine 1 of machine type 1 is allocated to cell 3.

This step is repeated until all machines are allocated to a cell. The final result of this stage, and thus the initial solution for the simulated annealing procedure, is shown in Table 8.

To compute a neighbouring solution, the operation intercellular flow needs to be computed. For that purpose, the interoperation flow matrix is rewritten in Table 9, in the block diagonal form. The third row in this table (Capacity) shows the capacity available in each machine.

Table 4
Operations allocation

Machine type t	1			2					
Machine m	1	2	3	1	2	3	3	2	
Operation	**5**	**9**	**12**	**14**	**2**	**6**	**7**	**10**	
Usage rate α_{it} (%)	68.9	66.5	66.4	69.7	63.8	38.5	38.1	10.9	
Remaining capacity	31.1	33.5	33.6	30.4	36.2	61.5	23.4	25.3	
Machine type t	3					4			
Machine m	1	2	2	1	1	1	2	2	2
Operation	**8**	**4**	**16**	**16**	**15**	**11**	**13**	**1**	**3**
Usage rate α_{it} (%)	74.8	67.9	39.0	39.0	16.4	53.2	26.9	22.3	18.5
Remaining capacity	25.2	32.1	0	18.3	1.9	46.8	73.1	50.9	32.4

Table 5
Intermachine flow matrix

	1,1	**1,2**	**1,3**	**2,1**	**2,2**	**2,3**	**3,1**	**3,2**	**4,1**	**4,2**	**Total**
1,1			190		224					209	623
1,2				237			43	316			596
1,3	190						56			101	347
2,1		237								123	361
2,2	224					222	59				505
2,3					222		227		85	182	715
3,1		43	56		59	227				112	497
3,2		316							137		453
4,1						85		137			221
4,2	209		101	123		182	112				727

Table 6
Average flow between each of the non-allocated machines and the cells, for the first iteration

Non-allocated machine	Cell		
	1	2	3
1,1	209	0	224
1,3	101	0	0
2,1	123	237	0
2,3	182	0	222
3,1	112	43	59
3,2	0	316	0
4,1	0	0	0

Table 7
Average flow between each of the non-allocated machines and the cells, for the second iteration

Non-allocated machine	Cell		
	1	2	3
1,1	209	0	224
1,3	101	0	0
2,1	123	119	0
2,3	182	0	222
3,1	112	22	59
4,1	0	69	0

The pair of operations that generate the highest intercellular flow is (10, 7). In an attempt to nullify this intercellular flow, operation 7 (performed in unit 3 of machine type 2) is selected to switch to cell 3, because machine 2,3 has less capacity available than machine 2,2 (where operation 10 is carried out). However, operation 7 cannot be moved because, although cell 3 has a machine of the required type to process the operation (unit 2 of machine type 2), this machine does not have enough capacity to do so. In fact, machine 2,2 has only 25.3% of its capacity available, while operation 7 requires 38.1% of the machine capacity to be processed (see Table 3). Another attempt is then made, this time by switching operation 10 to cell 1, with success. Operation 10 needs 10.9% of the capacity of a machine of type 2, and unit 3 of this machine type in cell 1 has 23.4% of its capacity available. The exchange is carried out and the neighbouring solution is found.

Table 8
Cells composition

Cell	Machines			
1	4,2	2,3	3,1	
2	1,2	3,2	2,1	4,1
3	2,2	1,1	1,3	

Table 9
BDF of the interoperation flow matrix for the initial solution

Cell	1								2					3			
Machine	**4,2**			**2,3**		**3,1**			**1,2**	**3,2**		**2,1**	**4,1**	**2,2**		**1,1**	**1,3**
Capacity	**32.4**			**23.4**		**1.9**			**33.5**	**0**		**30.4**	**46.8**	**25.3**		**31.1**	**33.6**
Operation	**1**	**3**	**13**	**6**	**7**	**8**	**15**	**16**	**9**	**4**	**16**	**14**	**11**	**2**	**10**	**5**	**12**
1						112											
3												123					
13																209	
6		182															
7						227											
8																	56
15															59		
16									43								
9										113							
4													137				
16									203								
14									237								
11				85													
2																224	
10					222												
5																	190
12			101														

In computing neighbouring solutions a situation can be reached where all the intercellular flow generating pairs of operations have been tested and it is not possible to switch any of the operations to another cell. When such is the case, the procedure computes the machine intercellular flow of the current solution and tries to reduce the overall intercellular flow by changing a machine to a different cell (by decreasing order of the intercellular flow generated and without violating the maximum number of machines per cell constraint). In Table 10 the machine intercellular flow for one such situation is showed.

Machines 2,3 and 4,1 generate the highest intercellular flow. Machine 2,3 cannot be switched to cell 2, because that would violate the maximum number of machines per cell constraint, but machine 4,1 can be changed to cell 1. This task is carried out and results in a new neighbouring solution. Note that the next neighbouring solution to be found will attempt again to change operations rather than machines.

The best solution found for the example problem, by applying the proposed simulated annealing procedure, is shown in Table 11. The flow inefficiency for this solution is 8.4% and the CPU time required to run this example was 5 seconds.

Table 10
BDF of the intermachine flow matrix

Cell	1		2				3			
Machines	**2,3**	**3,1**	**1,2**	**2,1**	**3,2**	**4,1**	**1,1**	**1,3**	**2,2**	**4,2**
2,3		286				267				
3,1	286		44					56		112
1,2		44		237	315					
2,1			237			123				
3,2			315			137				
4,1	267			123	137					
1,1								190	224	209
1,3		56					190			101
2,2							224			
4,2		112					209	101		

Table 11
Best solution found for the example problem

	Cell											
	1					2						
Machine	3,1			2,3		4,1			2,1		3,2	1,2
Operation	8	15	16	10	15	1	3	11	6	14	16	9
% Operation	100	100	18	100	100	100	100	100	100	100	82	100
	3											
Machine	1,1	1,3	2,2	4,2								
Operation	5	12	2	13								
% Operation	100	100	100	100								

5. Computational experience

The relatively few research papers that deal with the cell formation problem with multiple identical machines aiming at minimising the intercellular flow limit the possible comparison of the performance of the proposed heuristic with similar methods. Nevertheless, the examples presented by Wu and Salvendy, 1999 were adapted in order for the proposed heuristic to be applied to them. These three test problems consider respectively 6, 13 and 22 parts, 8, 13 and 22 machine types and 10, 18 and 23 machines. The best results derived from the application of the heuristic were the same for each of the three test problems analysed, and the computational effort associated was within acceptable limits (<10 seconds).

To further validate the proposed heuristic, a set of data concerning part of an auto parts facility, already configured as a cellular manufacturing system, was gathered. This study involved a section of a facility with 38 units of 21 different types of machines, which carry out 140 operations over 15 distinct parts. The process organisation is hybrid, having manufacturing cells side by side with process oriented departments. The actual layout of the section includes three cells to process the parts, a bottleneck machine, through which all the parts need to be routed before going to a finishing cell, where the final operations are performed over all parts. The result obtained from the application of the proposed heuristic led to a result better than the actual cellular configuration, using the flow inefficiency as the measure of the quality of the solution. Unfortunately, the result obtained is not expected to produce any practical results, as we have only dealt with part of the actual facility and the bottleneck machine is also shared by other cells other than the ones analysed. Nevertheless, this experience was useful to show that the heuristic performed rather well for a large-scale problem. Supplementary data is being collected to apply the heuristic for the complete facility.

6. Conclusion

In this paper a mathematical programming model for the cell formation problem with multiple, functionally identical, machines has been developed. The model goal is the minimisation of the intercellular flow and, for that purpose, takes into account the flow volume between the operations. Subsequently a simulated annealing procedure has been proposed to solve the model, which allows the user to control both the size and the number of cells to be generated. The procedure was tested with a set of problems presented by Wu and Salvendy, 1999 and also applied to a real world case. The computational results indicate the efficiency of the present model, even for large sized problems.

The application of the heuristic in an industrial environment showed that future research needs to consider issues like restricting machine to particular cells, duplication of machines and pre-definition of the number of units of each machine type.

References

Adil, G.K., Rajamani, D., Strong, D., 1993. A mathematical model for cell formation considering investment and operational costs. European Journal of Operational Research 69 (3), 330–341.

Adil, G.K., Rajamani, D., Strong, D., 1997. Assignment allocation and simulated annealing algorithms for cell formation. IIE Transactions 29, 53–67.

Atmani, A., Lashkari, R.S., Caron, R.J., 1995. A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing. International Journal of Production Research 33 (1), 1–15.

Boctor, F.F., 1991. A linear formulation of the machine-part cell formation problem. International Journal of Production Research 29 (2), 343–356.

Burbidge, J.L., 1963. Production flow analysis. Production Engineer 42 (12), 742–752.

Burbidge, J.L., 1992. Change to group technology: Process organization is obsolete. International Journal of Production Research 30 (5), 1209–1219.

Chen, C.-L., Cotruvo, N.A., Baek, W., 1995. A simulated annealing solution to the cell formation problem. International Journal of Production Research 33 (9), 2601–2614.

Del Valle, A.G., Balarezo, S., Tejero, J., 1994. A heuristic workload-based model to form cells by minimizing intercellular movements. International Journal of Production Research 32 (10), 2275–2285.

Deutsch, S.J., Freeman, S.F., Helander, M., 1998. Manufacturing cell formation using an improved p-median model. Computers and Industrial Engineering 34 (1), 135–146.

Heragu, S.S., Gupta, Y.P., 1994. A heuristic for designing cellular manufacturing facilities. International Journal of Production Research 32 (1), 125–140.

Jeon, G., Broering, M., Leep, H.R., Parsaei, H.R., Wong, J.P., 1998. Part family formation based on a new similarity coefficient which considers alternative routes during machine failure. Computers and Industrial Engineering 35, 479–482.

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Kusiak, A., 1985. The part families problem in Flexible Manufacturing Systems. Annals of Operational Research 3, 279–300.

Okogbaa, O.G., Chen, M., Changchit, C., Shell, R.L., 1992. Manufacturing system cell formation and evaluation using a new inter-cell flow reduction heuristic. International Journal of Production Research 30 (5), 1101–1118.

Seifoddini, H., Hsu, C.-P., 1994. Comparative study of similarity coefficients and clustering algorithms in cellular manufacturing. Journal of Manufacturing Systems 13 (2), 119–127.

Selim, M.H., Askin, R.G., Vakharia, A.J., 1998. Cell formation in group technology: Review, evaluation and directions for future research. Computers and Industrial Engineering 34 (1), 3–20.

Shargal, M., Shekhar, S., Irani, S.A., 1995. Evaluation of search algorithms and clustering efficiency measures for machine-part matrix clustering. IIE Transactions 27 (1), 43–59.

Singh, N., 1993. Design of cellular manufacturing systems: An invited review. European Journal of Operational Research 69 (3), 284–291.

Sofianopoulou, S., 1997. Application of simulated annealing to a linear model for the formulation of machine cells in group technology. International Journal of Production Research 35 (2), 501–511.

Srinivasan, G., 1994. A clustering algorithm for machine cell formation in group technology using minimum spanning tree. International Journal of Production Research 32 (9), 2149–2158.

Su, C.-T., Hsu, C.-M., 1998. Multi-objective machine-part cell formation through parallel simulated annealing. International Journal of Production Research 36 (8), 2185–2207.

Tam, K.Y., 1990. An operation sequence based similarity coefficient for part families formation. Journal of Manufacturing Systems 9 (1), 55–68.

Venugopal, V., 1999. Soft-computing-based approaches to the group technology problem: A state-of-the-art review. International Journal of Production Research 37 (14), 3335–3357.

Venugopal, V., Narendran, T.T., 1994. Machine-cell formation through neural network models. International Journal of Production Research 32 (9), 2105–2116.

Wu, N., Salvendy, G., 1999. An efficient heuristic for the design of cellular manufacturing systems with multiple identical machines. International Journal of Production Research 37 (15), 3519–3540.