SRC TR 88-77

An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications

by

G. Harhalakis, R. Nagi, and J.M. Proth

AN EFFICIENT HEURISTIC IN MANUFACTURING CELL FORMATION FOR GROUP TECHNOLOGY APPLICATIONS

G. Harhalakis[1], R. Nagi[1], J.M. Proth[2]

[1]Department of Mechanical Engineering & Systems Research Center,
University of Maryland, College Park, MD 20742, U.S.A.
[2]Institut National de Recherche en Informatique et en Automatique
INRIA-LORRAINE, Technopole de NANCY-BRABOIS,
Campus Scientifique - Bd des Aiguilettes -B.P. 239,
54506 VANDOEUVRE LES NANCY CEDEX, France

In this paper, the problem of finding a good decomposition of the manufacturing system into manufacturing cells, that can be assigned to part families is addressed. A simple twofold heuristic algorithm capable of minimizing the inter-cell material movement, and addressing industrial applications of realistic dimensions is presented. The first step of the proposed heuristic is a bottom-up aggregation procedure to minimize the "Normalized Inter-Class Traffic". The second step is a procedure to attempt further improvement, in which the significance of a machine to a cell is validated.

Key words : Group Technology, Clustering, Inter-class traffic

*Supported in part by the National Science Foundation's Engineering Research Centers Program: NSFD CDR 8803012

0. Introduction

Group Technology (GT) has been recognized as the key to alleviate problems of material-handling, management and control, and productivity of a typical batch-manufacturing system, which has profound implications on the profitability and overall operational efficiency of a manufacturing organization. GT can also simplify the design and process planning of new products by taking advantage of similarities of part design characteristics. The basic idea of GT from the manufacturing viewpoint is the decomposition of the manufacturing system into subsystems, by classifying parts into families and machines into machining cells, based on the similarity of part manufacturing characteristics. Parts that have to undergo similar operations and require the same set of machines for these operations are grouped together, and these machines are grouped into a machine cell, thus forming the subsystems. It is important to note that the design characteristics cannot usually be used to find the manufacturing subsystems because the manufacturing characteristics of seemingly similar parts may be entirely different (and vice-versa).

Finding a completely unencroached partition, wherein each part of a part family (class) remains confined to one machine cell (or alternately, machines of a particular manufacturing cell operate upon parts of only the corresponding class), is in fact an illusion in a typical industrial case. The presence of alternate process plans, duplication of machines, and subcontracting may not always eliminate the problem of "inter-class" transfers, and the situation becomes more aggravated in the case of make-to-order parts.

Many heuristic and non-heuristic methods have been developed by Askin (1987), Chandrasekharan (1987), Garcia (1985, 1986), King (1979, 1980), Kumar (1986), Kusiak (1985, 1987), Mc Cormick (1972). Bottom-up aggregation procedures have been used by McAuley (1972) and Leskowsky (1987), in which aggregation between groups takes place on the basis of similarity coefficients.

Most of the suggested algorithms in the literature are either not amenable to problems of a large size or are computationally prohibitive in typical industrial applications. Furthermore, they do not address the following issues :

• The sequence of operations

A typical matrix formulation clustering approach, tries to confine the operations of a part to the corresponding class of machines regardless of the sequence of operations. It is evident that if an intermediate operation of a part is in an external cell it will necessitate two inter-class transfers, as opposed to an operation that happens to be the first or the last operation. On the other hand, having more than one consecutive operations in the external cell, is just as bad as having a single outside operation, as the material handling effort involved is the same (The management and control effort is also almost the same). Thus, the sequence of operations is of great significance.

• Non-consecutive operations on the same machine

It is not uncommon that the same machine is used more than once in a routing; and if such a part has to visit foreign cells its implications on the material handling are significant. Matrix formulation approaches cannot address this situation because each element of the incidence matrix can hold only one entry.

The proposed heuristic addresses a real-life situation in which a perfect decomposition is not possible (an encroached case). It tries to find machine cells between which the inter-cell traffic is minimum. The procedure is twofold. The first step is a bottom-up aggregation which finds a basic assignment based on the volume (cost) of material flow; at each step of the algorithm two classes between which the "Normalized Inter-Class Traffic" is maximum are grouped, provided the size of the union remains within prescribed limits (A hierarchical clustering). The second step is a refinement procedure in which the basic assignment is tried to be improved if possible. In this step, it is ascertained that each machine is assigned to the cell in which it is the most significant.

The paper is divided into five main sections. In the first section a fast twofold heuristic is presented to define the manufacturing cells. The algorithms are presented in section two. In the third section evaluation criteria for determining the fairness of the solution have been suggested. The fourth section is devoted to an example, and the last section describes an industrial application, to show the efficacy and practical applicability of this method.

1. Setting the problem

Consider a set $P = \{p_1, p_2, \ldots, p_n\}$ of *n* part types and a set $M = \{m_1, m_2, \ldots, m_m\}$ of *m* machine types.

Let u_i be the weight of part p_i . The weight of a part may be the volume of production or the number of batches, and/or a cost factor. The cost factor can be a combination of the following :

• Material handling cost : depending upon the size, shape, weight, etc. of the part. (Based the need of different types of material handling equipment like forklifts, cranes,...).

• Cost of the part : In order to minimize the total "dollar" value of the work-in-process (WIP). (The material movement is generally faster within a cell, rather than between different cells. Thus, a costly part critical to the WIP, should be confined to a single cell.)

Let r_k denote the sequence of machines as in the routing of part p_k.

Let $C = \{c_1, c_2, \ldots, c_w\}$ be a partition of the set M into w subsets or classes.

For p_k, we define x^k_{ij} , as the number of times any machine belonging to c_j is the immediate successor of any machine belonging to c_i.

The traffic t_{ij} between two distinct classes c_i and c_j is defined as follows :

$$t_{ij} = \sum_{k=1}^{n} u_k (x^k_{ij} + x^k_{ji}); \qquad (1)$$

We denote by T_{ij}, the Normalized Inter-Class Traffic between c_i and c_j as :

$$T_{ij} = t_{ij} / (n_i + n_j) ; \qquad (2)$$

where

$n_i = \mathrm{card}(c_i)$

$n_j = \mathrm{card}(c_j)$

We finally define N as the maximal number of machines in a class.

The problem consists in finding the partition $C = \{c_1, c_2, \ldots, c_w\}$ of classes to minimize :

$$\text{MIN} : \quad \sum_{i=2}^{w} \sum_{j=1}^{i-1} T_{ij} \qquad (3)$$

$$\text{S.T.} \quad n_k \leq N; \quad k = 1, \ldots, w$$

Note that the traffic between classes is a symmetric matrix. Thus we can deal with the lower half only.

For example, let us consider three part types, and three machine classes in a production system. Class 1 contains two machines (M1 and M3), classes 2 and 3 are composed of one machine each (M2, and M4 respectively) as indicated in figure 1. For simplicity the weights of the parts are equal to unity each : $u_1=u_2=u_3=1$.

The sequence of machines is :

$r_1 = < m_1 , m_2 , m_4 , m_1 , m_2 > ; r_2 = < m_1 , m_3 > ; r_3 = < m_4 , m_1 , m_4 >$.

Then

$t_{12} = u_1 (x^1_{12} + x^1_{21}) + u_2 (x^2_{12} + x^2_{21}) + u_3 (x^3_{12} + x^3_{21})$ or

$t_{12} = 1 (2 + 0) + 1 (0 + 0) + 1 (0 + 0) = 2.$

$t_{23} = u_1 (x^1_{23} + x^1_{32}) + u_2 (x^2_{23} + x^2_{32}) + u_3 (x^3_{23} + x^3_{32})$ or

$t_{23} = 1 (1 + 0) + 1 (0 + 0) + 1 (0 + 0) = 1.$

$t_{13} = u_1 (x^1_{13} + x^1_{31}) + u_2 (x^2_{13} + x^2_{31}) + u_3 (x^3_{13} + x^3_{31})$ or

$t_{13} = 1 (0 + 1) + 1 (0 + 0) + 1 (1 + 1) = 3.$

$T_{12} = t_{12} / (n_1 + n_2) = 2 / (2 + 1) = 0.66.$

$T_{23} = t_{23} / (n_2 + n_3) = 1 / (1 + 1) = 0.5.$

$T_{13} = t_{13} / (n_1 + n_3) = 3 / (2 + 1) = 1.$

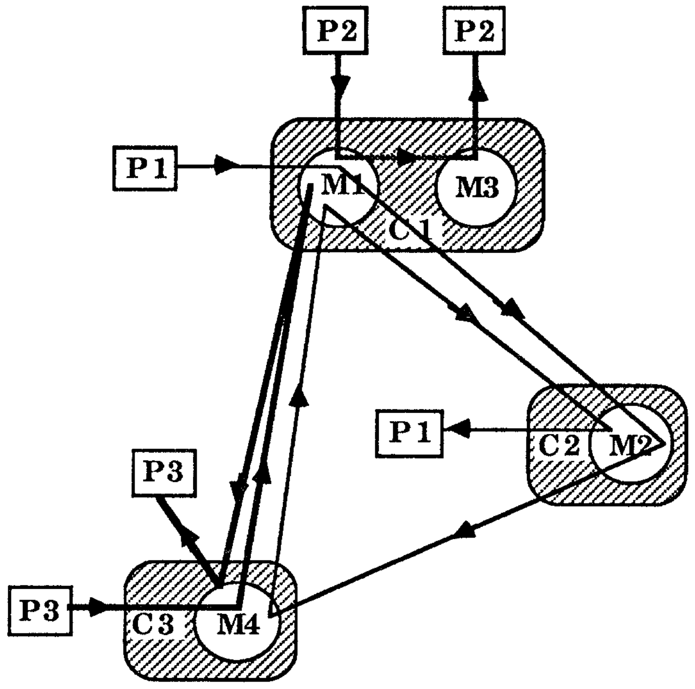

Figure 1 : A Production System showing traffic between machine classes

2 The Algorithm

We propose a twofold heuristic procedure to reach a good, if not optimal solution.

2.1 Basis of the algorithm

2.1.1 The bottom-up aggregation procedure

At the beginning of the algorithm each machine is placed in a separate class. At each step of the minimization procedure, the Normalized Traffic for all the feasible aggregations are calculated. Feasible aggregations are those which if allowed, the aggregate will not have machines more than the defined limit. The two classes between which the Normalized Traffic is maximum are aggregated into a single class. There is thus a reduction in number of classes by unity. It is obvious that the total Inter-class traffic in the system will have decreased by the value that is equal to that between these two classes concerned.

The traffic between classes is now revised by the following rules :

The traffic between classes is now revised by the following rules :
1) The traffic between two classes that were unaffected remains the same.
2) The traffic between an unaffected class and the new aggregate is the summation of the traffic between that unaffected class and the components of the aggregated class.

The procedure is continued until it is either not possible to have any feasible aggregation or the traffic between each of the existing classes is zero (perfect decomposition).

2.1.2. The local refinement procedure

In the beginning of the refinement procedure we have a basic partition of the system into machine-cells. We try to maximize the total intra-class traffic in the system, or in other words, we try to convert the "inter-class traffic" into "intra-class traffic". For this purpose, the significance of each machine to the cell it has been assigned to has to validated. Reflecting back to the basic assignment procedure: machines are grouped significantly, but once a machine is assigned to a particular class, it cannot be reassigned to a newly aggregated class, even if it is more suitable in that one. Thus in the refinement procedure, at each step of the algorithm one machine is considered as a separate external entity, and its traffic with each of the existing classes is evaluated. This machine is then assigned to the class with which its interaction is the most significant. Usually the machine is reassigned to the same class as in the basic assignment, but it is important to note that reassignments are possible. The process is repeated for each machine. Reassignments will also impact on the natural size of the cells.

2.2. The detailed algorithm

In this section the algorithms for the suggested twofold heuristic are presented.

2.2.1. The bottom-up aggregation procedure

The pseudo-code for the bottom-up aggregation procedure for the formation of basic machine cells is presented in this section.

```
repeat
    For $i$ = 2 to w
        For $j$ = 1 to (i - 1)
            find max($T_{ij}$)    s.t. $n_i + n_j \leq N$;
        end;
    end;
    if(max = 0)
        either change N or quit ;
    Group i and j for which $T_{ij}$ is maximum (Note : i > j)
    Begin
        For l = 1 to (j - 1)
            $t_{jl} = t_{jl} + t_{il}$
        end;
        For l = (j + 1) to (i - 1)
            $t_{lj} = t_{lj} + t_{il}$
        end;
        For l = (i + 1) to w
            $t_{lj} = t_{lj} + t_{li}$ ;
            For r = i to l
                $t_{lr} = t_{l(r-1)}$
            end;
        end;
        For l = i to (w - 1)
            For r = 1 to i
                $t_{lr} = t_{(l+1)r}$
            end;
        end;
    End;
    $n_j = n_j + n_i$ ;
    w = w - 1;
    For $l$ = i to w
        $n_l = n_{(l+1)}$
    end;
until ($T_{ij}$ = 0)
```

In the algorithm, a lower triangular "Traffic" matrix is operated upon. The best feasible aggregation is identified, and operations are performed according to the rules specified in section 2.1.1. If no feasible aggregation is possible, the user can either change the value of 'N' or exit the algorithm.

2.2.2. The local refinement procedure

The pseudo-code for the local refinement procedure for the improvement of the basic machine cells is presented in this section.

```
For $i = 1$ to $m$
    determine $c_j$ , the class $m_i$ is assigned in the basic solution;
    For $l = 1$ to $w$
        find $G^l_i$
    end;
    Let $G^k_i = MAX\{ G^l_i \}$
    if( $G^k_i > G^j_i$ )
        reassign $m_i$ from $c_j$ to $c_k$ ;
end.
```

Where G^k_i is the traffic of machine m_i with class c_k.

If the initial traffic between machines (assumed here to be in different groups) is defined as in (1);
and we also define

$$z_i^k = \begin{cases} 1 & \text{if machine } m_i \text{ belongs to the class } c_k \\ 0 & \text{otherwise} \end{cases} \qquad i = 1,\dots, m; \; k = 1,\dots, w$$

The criterion is defined as :

$$\mathrm{MAX}\left\{ G_i^k = \sum_{j=1}^{m} z_j^k \times t_{ij} \right\} \qquad k = 1,\dots, w$$

for every i; i=1,...,m

3. Evaluation

In order to keep the problem as general as possible, without getting into the specifics of cost/distance, costs of inter-class movements, and details of duplication of machines which are peculiar to a particular problem, we have introduced the following evaluation criteria. The notations used are the same as in section 1.

1) **Global Efficiency :** It is the ratio of the total number of operations that are performed within the classes to the total number of operations in the system.

$$\text{Global Efficiency} = \frac{\sum_{i=1}^{n} s_i}{\sum_{i=1}^{n} r_i}$$

where s_i is the number of operations in r_i that are performed in the class corresponding to p_i.

This is a global measure of the number of operations that are being performed on products within the classes they are assigned to. It reflects the effectiveness of the assignment to confine the operations of products within their respective classes as far as possible. The higher the Global Efficiency, the better the assignment.

2) **Group Efficiency :** It is the ratio of the difference between the total number of maximum external classes that could be visited and the total number of external classes actually visited by the products to the total number of maximum external classes that could be visited by them.

The maximum number of external classes that could be visited is :

$$E_w = \sum_{i=1}^{n} \mathrm{MIN}\{(q_i - 1), (w - 1)\}$$

where q_i is the number of different machines belonging to r_i.

For part p_i define $x_{ik} = \begin{cases} 1 & \text{if part } p_i \text{ visits class } k \\ 0 & \text{otherwise} \end{cases}$; $k = 1, \ldots, w$

The total number of classes actually visited by the products :

$$A_w = \sum_{i=1}^{n} \sum_{k=1}^{w} (x_{ik} - 1)$$

$$\text{Group Efficiency} = \frac{E_w - A_w}{E_w}$$

This efficiency provides equal penalty if there is one or more than one operations that are performed on a product in the same foreign class. This is a more abstract measure, but it reflects the effectiveness of the assignment to confine the products to as few foreign classes as possible. As the manufacturing control and scheduling complexity in a cell is related to the number of part types concerned, it is desired to have only a few foreign parts visiting a cell. However, if a foreign part is visiting the cell, then the complexity is not increased in accordance to the number of operations to be performed. For example in the Global efficiency if a

part has two external operations, then the penalty is irrespective of the fact that they are performed in the same foreign class or in different ones, whereas the Group efficiency tries to distinguish between the two. The higher the Group Efficiency, the better the assignment is.
Note that $c > 1$, i.e. there should be at least one foreign class in the system.

3) **Group Technology Efficiency :** It is the ratio of the difference in the maximum number of inter-class travels possible and the number of inter-class travels actually required by the system to the maximum number of inter-class travels possible.

The maximum number of inter-class travels possible in the system is :

$$I = \sum_{i=1}^{n} (r_i - 1)$$

For part p_i define $x_{ik} = \begin{cases} 0 & \text{if operations } k, k+1 \text{ are performed in the same class} \\ 1 & \text{otherwise} \end{cases}$

$k = 1, \dots, r_i - 1;\ i = 1, \dots, n$

The number of inter-class travels actually required by the system is :

$$U = \sum_{i=1}^{n} \sum_{k=1}^{r_i - 1} x_{ik}$$

$$\text{Group Tech Efficiency} = \frac{I - U}{I}$$

This is a very powerful criterion, which takes into account the sequence in which the operations are performed apart from the class in which they are performed. This efficiency provides a penalty of 1 for each inter-class travel. The higher the Group Technology Efficiency, the better the assignment is.

COMMENTS :

1. The three criteria are not entirely independent
2. Efficiency in the case of an encroached case decreases as the desired number of classes is increased.
3. All three are good criteria for choice between two assignments.

4. An Example

In this section an example of small size is presented to illustrate the aggregation method proposed. There are 20 part types and 20 machine types. For simplicity non-consecutive operations on the same machine are not considered (this is not however restrictive); also the weights of the parts is assumed to be unity. The maximal number of machines per class is set to 5.

The initial incidence matrix, with the entries indicating the operation number is shown in figure 2. It is a fairly encroached case; that is if we make an attempt to find completely uninteracting classes, the only solution is one class !

```
    || 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2|
    || 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0|
-----------------------------------------------
   1|| 2 . . . . . . . 3 . . 1 . . . . . 4 . 5|
   2|| . 3 2 . . . . . . . 1 . . . . . . . . .|
   3|| . . . . . . . 1 . . . . . . . . . . 3 2|
   4|| . 3 1 . . . . . . 4 2 . . . . . . . . .|
   5|| . . . 1 . 3 4 . . . . . . . 2 . . . . .|
   6|| . . . . 5 . . . . . 1 . . 2 . 3 4 . . .|
   7|| . . . . 1 . . . . . . . . . . 2 3 . . .|
   8|| . . . 5 . . 3 . 4 . . . 2 . 1 . . . . .|
   9|| 4 . . . . . . . 2 . 3 5 . . . . . 1 . .|
  10|| . . . . . . . 3 . . . . . . . . . . 1 2|
  11|| . . 3 . . . . . . . 1 . . 2 . . . . . .|
  12|| 5 . . . 3 . . . 1 . . 4 . . . . . 2 . .|
  13|| . . . . . 1 2 . . . . . . . 3 . 4 . . .|
  14|| 3 4 . . . . . 1 . 2 . . . . . . . . . .|
  15|| . . . . . . . . . . . . 1 2 . 3 4 . . .|
  16|| . . . . . 3 2 . . . . . . . 1 . . . 4 .|
  17|| 2 . . . . . . . 1 . . 3 . . . . . . . .|
  18|| . . . . . . . 1 . 4 . . . . . . . . 2 3|
  19|| . 2 1 . 4 . . . . . 3 . . . . . . . . .|
  20|| 3 . . . . . . . . 2 . 4 . . . . . 1 . .|
-----------------------------------------------
```

Figure 2 : Initial Incidence Matrix

The initial traffic between machines (assigned to different classes) is shown in figure 3(a). The leftmost column indicates the number of machines in that class. From figure 3(a) it is observed that the maximum traffic is between classes 12 and 1 (Normalized traffic is one half of the traffic because all classes are of equal size, that is unity) , the total inter-class traffic is 59 units. Thus, the first grouping is of machine 12 to machine 1, and subsequently the size of class 1 becomes 2. The number of classes has been reduced from 20 to 19, and the total inter-class traffic is 54 units. The aggregation steps are presented in figure 3(b).

```
 1 : 0
 1 : 1 0
 1 : 0 2 0
 1 : 0 0 0 0
 1 : 0 0 0 0 0
 1 : 0 0 0 0 0 0
 1 : 0 0 0 0 0 3 0
 1 : 0 0 0 0 0 0 0 0
 1 : 2 0 0 1 0 0 1 0 0
 1 : 2 1 0 0 0 0 0 1 0 0
 1 : 1 2 2 0 1 0 0 0 1 0 0
 1 : 5 0 0 0 1 0 0 0 0 0 0 0
 1 : 0 0 0 0 0 0 1 0 0 0 0 0 0
 1 : 0 0 1 0 0 0 0 0 0 0 2 0 1 0
 1 : 0 0 0 1 0 1 2 0 0 0 0 0 1 0 0
 1 : 0 0 0 0 1 0 0 0 0 0 0 0 0 2 0 0
 1 : 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 3 0
 1 : 0 0 0 0 1 0 0 0 3 1 0 0 0 0 0 0 0 0
 1 : 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
 1 : 0 0 0 0 0 0 0 2 0 1 0 0 0 0 0 0 0 1 3 0
TOTAL INTER-CLASS TRAFFIC = 59
Grouping between classes : 12 1
```

Figure 3(a) : Initial traffic between machines

Iter No.	c	Total Inter-Class Traffic	Grouping Between
1	20	59	12 & 1
2	19	54	7 & 6
3	18	51	15 & 14
4	17	48	15 & 8
5	16	45	16 &15
6	15	42	3 & 2
7	14	40	9 & 2
8	13	36	11 & 5
9	12	33	12 & 6
10	11	30	10 & 2
11	10	27	8 & 1
12	9	25	9 & 4
13	8	23	7 & 1
14	7	20	7 & 5
15	6	18	5 & 3
	5	17	

Figure 3(b) : Aggregation steps

An intermediate situation at the 11th iteration leading to the 12th is shown in figure 3(c). It can be observed that even though the highest traffic '2' is found 7 times, the Normalized Traffic is the highest for between two classes $T_{8,1} = T_{10,4} =$ 0.66 units owing to the large size of the other classes. We choose to group the first one i.e. group class 8 to 1. This brings us to the 12th iteration in which we group class 9 (which was class 10 in the 11th iteration) to 4. In this way, we are able to make significant groupings in a step-wise manner. The final number of classes obtained is 5, and the traffic matrix at the end of the aggregation is shown in figure 3(d). This also helps us in determining the physical proximity of two cells in the plant layout.

```
 2 : 0
 4 : 2 0
 1 : 0 0 0
 1 : 1 1 0 0
 3 : 0 0 1 0 0
 3 : 0 0 0 0 1 0
 2 : 2 1 1 1 1 1 0
 1 : 2 1 0 0 0 2 1 0
 1 : 0 1 0 0 2 0 0 0 0
 2 : 0 2 0 2 1 0 0 0 0 0
TOTAL INTER-CLASS TRAFFIC = 27
Grouping between classes : 8 1
```

```
3 : 0
4 : 3 0
1 : 0 0 0
1 : 1 1 0 0
3 : 0 0 1 0 0
3 : 2 0 0 0 1 0
2 : 3 1 1 1 1 1 0
1 : 0 1 0 0 2 0 0 0
2 : 0 2 0 2 1 0 0 0 0
TOTAL INTER-CLASS TRAFFIC = 25
Grouping between classes : 9 4
```

```
 5 : 0
 4 : 4 0
 5 : 2 1 0
 3 : 2 3 1 0
 3 : 3 0 1 0 0
TOTAL INTER-CLASS TRAFFIC = 17
```

<u>Figure 3(c) :</u>
<u>Intermediate steps - 11th to 12th iteration</u>

<u>Figure 3(d) :</u>
<u>Final traffic between classes</u>

The products are assigned to these machine cells and finally we obtain an assignment as shown in figure 4(a). In this case no improvement to the basic assignment was possible, as each machine is the most significant to the cell it was assigned to. The assignment is evaluated with respect to three efficiencies which have been introduced in section 3.

```
   || 0 0 1 1 1| 0 0 1 1| 0 0 0 1 1| 0 1 1| 0 1 2|
   || 1 9 0 2 8| 2 3 1 4| 4 6 7 3 5| 5 6 7| 8 9 0|
---------------------------------------------------
  1|| 2 3 . 1 4| . . . .| . . . . .| . . .| . . 5|
  9|| 4 2 . 5 1| . . 3 .| . . . . .| . . .| . . .|
 12|| 5 1 . 4 2| . . . .| . . . . .| 3 . .| . . .|
 14|| 3 . 2 . .| 4 . . .| . . . . .| . . .| 1 . .|
 17|| 2 1 . 3 .| . . . .| . . . . .| . . .| . . .|
 20|| 3 . 2 4 1| . . . .| . . . . .| . . .| . . .|
---------------------------------------------------
  2|| . . . . .| 3 2 1 .| . . . . .| . . .| . . .|
  4|| . . 4 . .| 3 1 2 .| . . . . .| . . .| . . .|
 11|| . . . . .| . 3 1 2| . . . . .| . . .| . . .|
 19|| . . . . .| 2 1 3 .| . . . . .| 4 . .| . . .|
---------------------------------------------------
  5|| . . . . .| . . . .| 1 3 4 . 2| . . .| . . .|
  8|| . 4 . . .| . . . .| 5 . 3 2 1| . . .| . . .|
 13|| . . . . .| . . . .| . 1 2 . 3| . . 4| . . .|
 16|| . . . . .| . . . .| . 3 2 . 1| . . .| . 4 .|
---------------------------------------------------
  6|| . . . . .| . . 1 2| . . . . .| 5 3 4| . . .|
  7|| . . . . .| . . . .| . . . . .| 1 2 3| . . .|
 15|| . . . . .| . . . 2| . . . 1 .| . 3 4| . . .|
---------------------------------------------------
  3|| . . . . .| . . . .| . . . . .| . . .| 1 3 2|
 10|| . . . . .| . . . .| . . . . .| . . .| 3 1 2|
 18|| . . 4 . .| . . . .| . . . . .| . . .| 1 2 3|
---------------------------------------------------

Global Efficiency = 81.012657%
Group Efficiency = 76.271187%
G. T. Efficiency = 71.186440%
```

Figure 4(a) : Final incidence matrix (5 classes)

If the initial limit of machines per class is set to 7, a different assignment with 4 machine cells is obtained (figure 4(b)). Thus, the assignment depends on the limit of machines which has to be discovered by a few runs.

There is another interesting point that is worth mentioning with respect to the assignment of machine 14 in figure 4(a). It seems that the machine 14 is more suitable in the 4th class (with 5, 16, 17) rather than the 2nd class because we would be eliminating two external operations (of part types 6 and 15) and we would add only an external operation (of part type 16) ! It can be noticed however that we would be increasing the total traffic in the system. Part 16 has an intermediate operation on this machine and this will require two inter-class transfers. On the other hand, we would be reducing the transfers of part 15 by only one. The transfers of part type 6 remains unaffected in one or the other case.

```
    || 0 0 1 1 1| 0 0 0 1 1 1 1| 0 0 0 1 1| 0 1 2|
    || 1 9 0 2 8| 2 3 5 1 4 6 7| 4 6 7 3 5| 8 9 0|
-------------------------------------------------
   1|| 2 3 . 1 4| . . . . . . .| . . . . .| . . 5|
   9|| 4 2 . 5 1| . . . 3 . . .| . . . . .| . . .|
  12|| 5 1 . 4 2| . . 3 . . . .| . . . . .| . . .|
  14|| 3 . 2 . .| 4 . . . . . .| . . . . .| 1 . .|
  17|| 2 1 . 3 .| . . . . . . .| . . . . .| . . .|
  20|| 3 . 2 4 1| . . . . . . .| . . . . .| . . .|
-------------------------------------------------
   2|| . . . . .| 3 2 . 1 . . .| . . . . .| . . .|
   4|| . . 4 . .| 3 1 . 2 . . .| . . . . .| . . .|
   6|| . . . . .| . . 5 1 2 3 4| . . . . .| . . .|
   7|| . . . . .| . . 1 . . 2 3| . . . . .| . . .|
  11|| . . . . .| . 3 . 1 2 . .| . . . . .| . . .|
  15|| . . . . .| . . . . 2 3 4| . . . 1 .| . . .|
  19|| . . . . .| 2 1 4 3 . . .| . . . . .| . . .|
-------------------------------------------------
   5|| . . . . .| . . . . . . .| 1 3 4 . 2| . . .|
   8|| . 4 . . .| . . . . . . .| 5 . 3 2 1| . . .|
  13|| . . . . .| . . . . . . 4| . 1 2 . 3| . . .|
  16|| . . . . .| . . . . . . .| . 3 2 . 1| . 4 .|
-------------------------------------------------
   3|| . . . . .| . . . . . . .| . . . . .| 1 3 2|
  10|| . . . . .| . . . . . . .| . . . . .| 3 1 2|
  18|| . . 4 . .| . . . . . . .| . . . . .| 1 2 3|
-------------------------------------------------

Global Efficiency = 86.075951%
Group Efficiency = 79.629631%
G. T. Efficiency = 76.271187%
```

Figure 4(b) : Final incidence matrix (4 classes)

5. Industrial application

The algorithm has been applied to a real-life manufacturing job-shop environment. The performance of the algorithm with respect to the various scenarios has been studied. In this section, the application and results have been discussed.

5.1. The Company

KOP-FLEX, manufactures a wide range of Power Transmission Products - Couplings (cylindrical parts). They have one of the largest machine shops in the U.S. east coast, and the present layout is functional. About 1,200 make-to-stock (MTS) part and 20,000 make-to-order (MTO) parts are manufactured. The shop floor has around 173 workcenters which include lathes, gear-shapers, drills, milling machines, etc.

5.2. Data collection and preparation

The following steps were followed :

- Selection of MTS parts only
- Retrieval of their routings
- Retrieval of the production volume (over time T)
- Elimination of obvious central facilities (e.g. packing, heat treatment)
- Aggregation of multiple consecutive operations on the same machine
- Aggregation of identical machines

Most companies hesitate to explore the possibility of a GT application because of the time and effort involved in collection of part and work-center details. The data collection and preparation (assisted by an MRP II package, and few data formatting programs) was very inexpensive and quick.

5.3. Program runs

Several runs for 1186 part types and 86 machine types were made. Central facilities, and work centers that were used by a large number of parts were placed outside the scope, because they are not significant to any specific cell and it is not desired to assign them to a specific class. In fact these were aggregated machines (of the same type), so they can be disaggregated and placed in the cells where they are used the most. Decision of central placement, duplication or otherwise are company specific.

5.4. Results

The algorithms were coded in C on the SUN/Unix platform.

Owing to the large size of the problem it is difficult to present the results in the form of an incidence matrix in this paper, however we present the results in the form of efficiencies that have been explained in section 3.
Since some of the products had about upto 15 operations each, it is intuitively felt that the limit of the size of the machine cells has to be around this value.

1) If the limit of machines per class is set to 12, and relaxed to 17, the basic assignment resulted in 12 machine cells. The cpu time for the basic algorithm was about 20 seconds. The products were then assigned to these cells and the efficiencies obtained were as follows :

Global Efficiency = 93.49%
Group Efficiency = 87.44%
G.T. Efficiency = 81.13%

In the improvement procedure, reassignments were made for 4 machines, and consequently the efficiencies were improved to :
Global Efficiency = 94.41%
Group Efficiency = 89.68%
G.T. Efficiency = 84.78%
The cpu time for the improvement procedure was 1.3 seconds.

2) If the limit of machines per class is set to 12, and relaxed to 15, the basic assignment resulted in 15 machine cells. The cpu time was about 20 seconds. The efficiencies after assigning the products were :
Global Efficiency = 93.33%
Group Efficiency = 87.11%
G.T. Efficiency = 80.67%

In this case the number of reassignments were large, and the algorithm had to be rerun in order to evaluate to significance in the new reassignment. Consequently the number of classes was decreased to 12. The efficiencies after the reassignment were the same as obtained in the previous assignment.

This demonstrates that the quality of the basic assignment depends on the limits of machines per cell, and on whether it is relaxed. The improvement procedure is able to locally improve the assignment. If the quality of the basic assignment is not good or the limit of machine chosen was not suitable for the problem, in that case several machines are reassigned to different classes and in this process some sets can become empty too. In most cases the natural size and number of classes are obtained after the improvement algorithm.

The run times of 20 seconds and 1.3 seconds for the two steps in a problem of dimension 1186 X 86 (n X m) seems to be attractive.

6. Conclusions

A simple, yet effective, method of grouping machines in order to minimize the inter-class material movement has been found and tested with a wide variety of inputs of varying degrees of encroachment. Respecting the sequence of operations, the inter-class traffic was attempted to be minimized. It was observed that in a low degree of encroachment case it is possible to find optimum results. If the input is highly encroached, the quality of the basic assignment is sensitive to the user defined limit on machines per cell, and to whether it is relaxed during the algorithm. The improvement algorithm is able to refine the basic assignment and leads to the natural size of the cells in most cases, but in some peculiar cases it may result in just a few large cells. Owing to the speed, a few enumerated runs can be made inexpensively. The results seem to be satisfactory in the case of an industrial data which is amenable to the application of GT. In addition, the system makes a direct attempt to minimize the inter-class traffic (cost). It can take into account multiple non-consecutive operations on the same machine, a case which has not been addressed by existing clustering methods.

A final remark : The algorithm is very fast and well adapted to problems of large dimension; the order of complexity is almost insensitive to 'n'.

7. Acknowledgements

This research has been partially supported by the Systems Research Center, University of Maryland, and the Institut National de Recherche en Informatique et en Automatique (INRIA-LORRAINE, France).

8. References

Askin, R., Subramanian, S.B. , 1987, "A cost-based heuristic for group technology configuration", International Journal of Production Research, 25(1), pp. 101-113.
Chandrasekharan, M.P., Rajagopalan, R. , 1987, "ZODIAC - an algorithm for concurrent formation of part-families and machine-cells", International Journal of Production Research, 25(6), pp. 835-850.
Garcia, H., Proth, J.M. , 1985, "Group Technology in Production Management : The Short Horizon Planning Level", Applied Stochastic Models and Data Analysis, 1, pp. 25-34.
Garcia, H., Proth, J.M. , 1986, "A new cross-decomposition algorithm : the GPM. Comparison with the bond energy method", Control and Cybernetics, 15(2), pp. 155-164.
King, J.R. , 1979, "Machine-Component Group Formation in Group Technology", OMEGA The International Journal of Management Science, 8(2), pp. 193-199.
King J.R. , 1980, "Machine-Component grouping using ROC algorithm", International Journal of Production Research, 18(2), pp. 213-231.
Kumar, R.K., Kusiak, A., Vannelli, A. , 1986, "Grouping of Parts and Components in Flexible Manufacturing Systems", European Journal of Operational Research, 24, pp. 387-397.
Kusiak, A. , 1985, "The Part Families problem in Flexible Manufacturing Systems", Annals of Operations research, 3, pp. 279-300.
Kusiak, A. , 1987, "The Generalized Group Technology Concept", International Journal of Production Research, 25(4), pp. 561-569.
Kusiak, A., Herangu, S.S. , 1987, "Group Technology", Computers in Industry, 9, pp. 83-91.
Leskowsky Z., Logan L., Vannelli A. , 1987, "Group Technology Decision Aids in An Expert System for Plant Layout", Modern Production Management Systems, Elsevier Science Publisher, pp. 561-585.
McAuley, J. , 1972, "Machine Grouping for Efficient Production", The Production Engineer, Feb., pp. 53-57.
Mc Cormick W.T., Schweitzer P.J., White T.E. , 1972, "Problem decomposition and data reorganization by a cluster technique", Operations Research, 20.