TECHNICAL RESEARCH REPORT

UNIVERSITY OF MARYLAND
COLLEGE PARK
1856

SYSTEMS RESEARCH CENTER

VE RI TAS
HARVARD

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Multiple Routings and Capacity Consideration in Group Technology Applications

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

TR 89-70

Multiple Routings and Capacity Considerations in Group Technology Applications

Rakesh Nagi[1], Jean-Marie Proth[2], George Harhalakis[1]

[1]Department of Mechanical Engineering & Systems Research Center,
University of Maryland, College Park, MD 20742, U.S.A.
[2]INRIA-LORRAINE, B.P. 239, 54506 VANDOEUVRE Cedex, France,
& Systems Research Center, University of Maryland, U.S.A.

This paper addresses the problem of manufacturing cell formation, given multiple part routings, and multiple functionally similar work-centers. The suggested choice of part routings favors the decomposition of the manufacturing system into manufacturing cells in a way that minimizes part traffic, along with satisfying the part demand and work-center capacity constraints. The proposed heuristic, iteratively solves two independent problems : (i) routing selection, and (ii) cell formation. The common objective is to minimize the inter-cell traffic in the system. The first problem is formulated as a linear-programming problem, while the latter is approached by an existing bottom-up aggregation procedure, known as Inter-Cell Traffic Minimization Method (ICTMM), enhanced appropriately. Applications of the proposed system include : (i) the design of a manufacturing facility with respect to machine layout, (ii) selection of part routings for changing product mixes, and (iii) assignment of new parts to part families, given the initial layout.

Key words : Group Technology, Manufacturing Cell, Part Families, Clustering, Inter-cell traffic.

1 Introduction

From a manufacturing viewpoint, the Group Technology (GT) concept is to partition a manufacturing system into cells, and part types to be manufactured into part families, based on the similarity of part manufacturing characteristics. The objective is to decrease the production cost by reducing material handling and transportation costs, work-in-process, and by simplifying production management.

The formation of manufacturing cells and part families for GT applications has been considered in the literature [1 - 15], with pre-specified routings. Heuristic and non-heuristic methods have been developed, in which the decomposition of the system is performed on the basis of a given set of part routings (one for each part), that are assumed to be known and unalterable. However, in practical applications, a particular part production process can be chosen from a set of alternative processes. Among the set of feasible process plans initially developed by some process planning function, clearly some are sub-optimal in terms of manufacturing costs, and can be ignored right away. However, some comparable plans may still be available. With reference to the concept of Group Technology, one might even settle to choose a sub-optimal plan in order to confine a part to a specific cell. This may be preferred to reduce queueing times, transportation times and costs, work-in-process, and to increase productivity.

Another reason for alternate process plans is the existence of functionally similar work-centers. In a functional layout, this is not a consideration, because the parts can be routed to any such available work-center. For example, a part 'p' requires a turning operation. Our turning facility consists of two identical lathes. The part 'p' could use either lathe when it is routed to the turning facility. On the other hand, in a GT environment, manufacturing cells usually consist of functionally dissimilar work-centers; the two lathes are likely to be placed in different cells. In this case, we would prefer to route 'p' to the lathe in its corresponding cell. Thus, there is a need to identify each work-center as a specific one. This unique identification of functionally similar work-centers leads to the existence of multiple routings.

In addition, most of the suggested approaches in the literature tend to disregard the capacity of work-centers. Furthermore, in the choice of part routings and formation of manufacturing cells, capacity constraints of the work-centers should be taken into account.

In this paper, we address the manufacturing cell formation for the GT problem with multiple part routings and multiple functionally similar work-centers. We also consider finite capacity for work-centers. The approach is amenable to large dimension problems, and computationally inexpensive. Furthermore, it is capable of addressing : (i) an encroached partition, i.e., where a perfect decomposition is impossible if each part (or each part family) remains confined to its corresponding cell, (ii) non-consecutive part operations on the same work-center, and (iii) the sequence of operations.

The proposed heuristic determines manufacturing cells to minimize the material flow within the shop. The choice of part routings is such that : (i) it is the most favorable to the decomposition of the manufacturing system from a part traffic viewpoint, (ii) the production demand of each part is satisfied, and (iii) the capacity constraint of each work-center is respected.

The proposed algorithm solves two problems : (i) the selection of routings, and (ii) the formation of manufacturing cells. The procedure is iterative, until a set of working routings and manufacturing cells are obtained with minimal inter-cell traffic. The first problem is formulated as a linear programming problem, while the second is solved by the Inter-Cell Traffic Minimization Method (ICTMM) presented in [5], or by a parametrized algorithm presented in [6]. ICTMM is a bottom-up aggregation procedure which aggregates work-centers to cells, based on the "Normalized Inter-Cell Traffic", and then validates the assignment of work-centers. The aggregation-validation procedure is continued until the cells are formed and no further reduction in traffic is possible under the prescribed cell size constraint.

The paper is organized as follows. In section 2, the notations and problem formulation are presented. Section 3 is devoted to the proposed algorithm. The Inter-Cell Traffic Minimization Method is presented in section 3. Proof of convergence, and determination of part families is also included in this section. Criteria for evaluating the solution are defined in section 4. Section 5 is devoted to a small example. Finally, our conclusions along with recommendations for further work form the contents of the last section.

2 Problem formulation

We consider a set $M = \{m_1, m_2, \ldots, m_m\}$ of m work-centers in a given manufacturing system. Each work-center is recognized as unique, i.e., functionally similar work-centers are referred to by a different identification. Let work-center m_j, have a finite availability of e_j units of time in a given horizon H; $j = 1,2,\ldots,m$. It is appropriate to mention here that e_j is the average capacity of work-center m_j, that can be calculated on the basis of its average down-time and maintenance schedule.

We also consider a set $P = \{p_1, p_2, \ldots, p_n\}$ of n part types to be manufactured. Let N_i be the production volume required for part type p_i in the chosen horizon H; $i = 1,2,\ldots,n$. Each part type has associated with it a set of alternative routings. Let the set $R_i = \{r^1{}_i, r^2{}_i, \ldots, r^{q(i)}{}_i\}$ of q(i) alternative routings be associated with part type p_i; $i = 1,2,\ldots,n$.

A routing $r^k{}_i$; $i = 1,2,\ldots,n$; $k = 1,2,\ldots,q(i)$, is defined by a sequence of work-centers to be visited. For $i = 1,2,\ldots,n$; $k = 1,2,\ldots,q(i)$, the routing $r^k{}_i = \{m^k{}_i(1), m^k{}_i(2), \ldots, m^k{}_i(s^k{}_i)\}$, is a sequence of $s^k{}_i$ work-centers. $m^k{}_i(y) \in M$ is the work-center required for the y-th operation of part type p_i using routing $r^k{}_i$; $y = 1,2,\ldots,s^k{}_i$. We also associate with each routing $r^k{}_i$, the corresponding processing times $\tau^k{}_i(m_x) \in R^+$. $\tau^k{}_i(m_x)$ is the processing time of one part of p_i at work-center m_x. Note :

(1) if a work-center m_x is not used in $r^k{}_i$, then, $\tau^k{}_i(m_x) = 0$.

(2) if a work-center m_x is required for two or more non-consecutive operations in $r^k{}_i$, then, $\tau^k{}_i(m_x)$ represents the summation of the individual processing times.

(3) Batch sizes and set-up times are not considered.

Let u_h be a weight associated with part type p_h; $h = 1,2,\ldots,n$. The weight of a part may be a combination of the material handling and the part costs, to express the relative importance of a part.

Finally, let MN denote the maximal number of work-centers permissible in a cell.

The information presented thus far is "acquired". In the following paragraphs, we will introduce the notation pertaining to the problem formulation.

Let $n^k{}_i$ be the production volume of part type p_i using routing $r^k{}_i$; $i = 1,2,\ldots,n$; $k = 1,2,\ldots,q(i)$.

Let $C = \{c_1, c_2, \ldots, c_w\}$ be a partition of the set M of work-centers into w subsets or cells. We define $x^k{}_h(i,j)$ as the number of times any work-center belonging to the cell c_j is the immediate successor of any work-center belonging to the cell c_i in the routing sequence $r^k{}_h$; $h = 1,2,\ldots,n$; $k = 1,2,\ldots,q(h)$.

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

$$t_{ij} = \sum_{h=1}^{n} \sum_{k=1}^{q(h)} u_h \times n^k{}_h \left(x^k{}_h(i,j) + x^k{}_h(j,i) \right) \tag{1}$$

Equation (1), representing the traffic between two cells, is the total exchange of parts between them, factored by the quantity and the weight of each part type.

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

$$T_{ij} = \frac{t_{ij}}{g_i + g_j} \tag{2}$$

where

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

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

Let $\quad z^k{}_h(i,j) = u_h \left(x^k{}_h(i,j) + x^k{}_h(j,i) \right)$

Then, (1) can be written as :

$$t_{ij} = \sum_{h=1}^{n} \sum_{k=1}^{q(h)} z^k{}_h(i,j) \times n^k{}_h \tag{3}$$

The problem consists of finding the partition $C = \{c_1, c_2, \ldots, c_w\}$ of cells, and the vector of decision variables $\mathbf{n} = [n^k{}_i]$; $i = 1,2,\ldots,n$; $k = 1,2,\ldots,q(i)$, in order to minimize :

$$\sum_{i=2}^{w} \sum_{j=1}^{i-1} t_{ij} \tag{4}$$

Subject to :

$$g_k \leq MN; \qquad k = 1,2,\ldots,w \tag{5}$$

$$\sum_{k=1}^{q(i)} n^k{}_i = N_i; \qquad i = 1,2,\ldots,n \tag{6}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{q(i)} n^k{}_i \, \tau^k{}_i(m_j) \leq e_j; \qquad j = 1,2,\ldots,m \tag{7}$$

Using (3), (4) can also be written as :

$$\sum_{h=1}^{n} \sum_{k=1}^{q(h)} \sum_{i=2}^{w} \sum_{j=1}^{i-1} z^k{}_h(i,j) \times n^k{}_h \tag{4'}$$

The objective function (4) represents the Total Inter-Cell Traffic in the system. The constraints (5) represent the cell size constraint. The constraints (6) represent the production volume requirement for each part type. Finally, (7) represents the capacity constraints of each work-center.

3 The Algorithm

In this section, we present the basis and the details of the proposed algorithm. Proof of convergence is also provided.

3.1 Basis of the Algorithm

The problem formulated in section 2 can be viewed as a linear programming problem with $\mathbf{n}$ as the vector of decision variables. The complexity however is that the coefficients $\Sigma_i \Sigma_j \, z^k{}_h(i,j)$, of the objective function (4'), and the satisfaction of constraint (5), are in fact dependent on the partition $C = \{c_1, c_2, \ldots, c_w\}$. Thus, we try to address this complexity by an iterative heuristic algorithm. We decouple the problem to two problems $\mathcal{P}1$ and $\mathcal{P}2$, that are solved in succession to minimize (4). The problem $\mathcal{P}1$ is : given an initial partition C (i.e., $\Sigma_i \Sigma_j \, z^k{}_l(i,j)$) for which (5) is satisfied, solve the linear program (4) subject to (6) and (7). The result of solving $\mathcal{P}1$ is a new vector $\mathbf{n}$, i.e., the selection of routings. The problem $\mathcal{P}2$ is : given a vector $\mathbf{n}$ for which (6) and (7) are satisfied, minimize (4) subject to (5). The result of solving $\mathcal{P}2$ is a new partition C^1, i.e., the formation of manufacturing cells. Problems $\mathcal{P}1$ and $\mathcal{P}2$ are solved iteratively until convergence is achieved.

$\mathcal{P}1$ can be solved by any of the available linear programming tools like Simplex method, etc. $\mathcal{P}2$ is a combinatorial problem, thus, we can use the heuristics suggested in [5],[6] to obtain a good if not optimal solution. Especially [5] presents the Inter-Cell Traffic Minimization Method (ICTMM), which solves $\mathcal{P}2$ very fast and with reasonably good results. It can therefore be used iteratively without much computational expense.

Here is the outline of the proposed method;

Initialize :
$C^0 = \{c_1, c_2, \ldots, c_w\}$ is a starting partition for which (5) is satisfied. (see Section 3.4)
i = 0; iteration number.
Repeat
    i = i + 1
    Solve $\mathcal{P}1$ to obtain $\mathbf{n}^i$ (using a linear programming method)
    Solve $\mathcal{P}2$ to obtain C^i (see Section 3.2)
Until $\mathbf{n}^i = \mathbf{n}^{i-1}$ & $C^i = C^{i-1}$

3.2 Inter-Cell Traffic Minimization Method

We propose an enhancement to the twofold heuristic algorithm presented in [5]. As explained in [5], the first phase consisted of a bottom-up aggregation procedure to determine the basic manufacturing cells (a hierarchical clustering). The second phase consisted of a local improvement by validating the significance of a machine to its assigned cell. In the proposed enhancement, we include a validation step following each aggregation of the first phase. The enhanced algorithm is presented below.

At the beginning of the algorithm, each work-center is placed in a separate cell. At each step of the minimization procedure, the Normalized Traffic for each feasible aggregation is calculated. A feasible aggregation is one in which two cells merge to form an aggregate that respects constraint (5). The two cells, between which the Normalized Traffic is maximum, are aggregated into a single cell. The number of cells in the system is subsequently reduced by one. It is obvious that the total inter-cell traffic in the system will have decreased by the value corresponding to the traffic between these two cells.

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

(1) The traffic between two unaffected cells remains the same.

(2) The traffic between an unaffected cell and the new aggregate is the summation of the traffic between that unaffected cell and the components of the aggregate.

Each aggregation is followed by a validation step that ascertains the significance of each work-center to its assigned cell. This step, leading to the occasional reassignments of work-centers, is essential because despite the fact that work-centers are grouped significantly, the bottom-up aggregation is a hierarchical clustering. Once a work-center is assigned to a particular cell, it cannot be withdrawn even if it is more suitable for one of the aggregates formed later.

In the validation step, we consider cells which have two or more work-centers, and arrange them in decreasing order of their cardinals. A work-center belonging to the first cell is considered as a separate external entity; its traffic with each cell is evaluated. This work-center is then assigned to the cell with which its interaction is the most significant. A reassignment is permitted only if constraint (5) is respected. The process is repeated for each work-center of this cell, and the work-centers of all other cells.

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

3.3 The detailed ICTMM algorithm

In this section the algorithm for the enhanced ICTMM is presented.

Initialize :
$C = C^0 = \{c_1, c_2, \ldots, c_w\}$, where :
$w = m$;
$c_i = \{m_i\}$, $g_i = 1$; $\quad i = 1,2,\ldots,w$
Find $t°_{ab} = t_{ab}$; $\forall$ a,b $(a > b)$
Note : $t°_{ab}$ denotes the traffic between machines m_a and m_b.

Repeat
    Let $E = \{(s,r) \mid g_s + g_r \leq MN,\ s = 2,\ldots,w,\ r = 1,\ldots,s-1\}$
    If $E \neq \phi$, compute (i,j) such that

$$T_{ij} = \max_{(s,r) \in E} T_{sr}$$

$c_j = c_j \cup c_i$
$g_j = g_j + g_i$
$c_{k-1} = c_k$; $\forall\ k > i$
$g_{k-1} = g_k$; $\forall\ k > i$
$w = w - 1$
Note the new partition is : $C = \{c_1, c_2, \ldots, c_w\}$
        Find t_{ab}; $\forall$ a,b $(a > b)$
        VALIDATE(C)
    Else
        either change MN or STOP.
Until $\Sigma_{i,j} T_{ij} = 0$

Procedure VALIDATE(C)
We consider $C_s = \{c_{s1}, c_{s2}, \ldots, c_{sz}\} \subseteq C$ where $g_{s1} \geq g_{s2} \geq \ldots \geq g_{sz} \geq 2$
For s = s1,s2,…,sz
    For each $m_i \in c_s$
        • Compute

$$I(m_i, c_r) = \sum_{m_k \in c_r} t°_{ik}$$

for $r \in G$, where :

$G = \{r \mid r \in \{s1, s2, \ldots, sz\};\ r \neq s, \text{ and } g_r < MN\}$

- Find $v \in G$ such that

$$I(m_i, c_v) = \underset{r \in G}{\mathrm{Max}}\ I(m_i, c_r)$$

- Compute
  $I(m_i, c_s - \{m_i\})$
- If $I(m_i, c_v) > I(m_i, c_s - \{m_i\})$ set :
  $c_s = c_s - \{m_i\}$
  $c_r = c_r \cup \{m_i\}$

Continue

Continue

End.

3.4 Convergence

RESULT : The algorithm presented in section 3.1 converges.
PROOF :
a) After the first time $\mathcal{P}1$ is solved, we have a feasible solution of $\mathcal{P}1$, i.e., a feasible selection of routings. At the beginning of solving $\mathcal{P}1$ in any subsequent iteration, the value of the objective function (4) corresponds to a feasible solution of $\mathcal{P}1$. Thus, the optimal solution of $\mathcal{P}1$ will be either the feasible solution we have at this step, or a better feasible solution, i.e., one with a reduced value of the objective function (4).

Thus, the objective function (4) is either reduced or at least remains constant after solving $\mathcal{P}1$. As mentioned above, this result holds after the first time $\mathcal{P}1$ is solved.

b) Solving $\mathcal{P}2$ consists of finding a partition of M which minimizes the inter-cell traffic, given the selection of routings, i.e., the vector **n**. At the beginning of solving $\mathcal{P}2$, a feasible partition of M as well as the corresponding value of the objective function are known for the given selection of routings, **n**. Thus the optimal partition of M leads to a value of the objective function (4) which is less than or equal to the previous one.

Thus, the objective function (4) is non increasing in value, and we obtain convergence at a local if not global optimum.

Q.E.D.

3.5 The initial partition

The algorithm leads to a solution that is dependent on the chosen initial partition C^0. Since the algorithm improves the partitions iteratively (hill-climbing strategy), it is only possible to converge to a local optimum. Convergence is usually obtained in less than three iterations.

It is appropriate to try a few random initial partitions to decide on the best one. However, in this section we also suggest a relatively simple, but good starting partition. This starting partition led to the optimum in most cases that were tried.

To determine the initial partition, we disregard the machine capacity constraints. Part production volumes are divided equally among the alternate routings. The initial partition is then obtained using ICTMM. This partition leads to good results especially when some parts have single routings, and when the alternate routings of a part are very similar. In most cases, this assumption reflects the true traffic between machines, and consequently leads to reasonable starting cells.

3.6 Part families

We begin with the decomposition of the manufacturing system into manufacturing cells, i.e., the partition $C^* = \{c_1, c_2, \ldots, c_w\}$, and the selected routings, i.e., the vector $\mathbf{n}^*$. The part families can be computed by the approaches suggested in [3],[4],[6]. The objective of this computation is the minimization of the number of operations performed in external cells. We adopt a rather simple approach. A part is assigned to the manufacturing cell in which the majority of its operations are performed. Thus, parts requiring the same manufacturing facilities are grouped in the same family. Note here that the number of part families obtained by this assignment will be equal to or less than the number of manufacturing cells.

4 Evaluation

To determine the "goodness" of a solution, and to compare solutions obtained by different initial partitions, we use the global evaluation criteria suggested in [5].

1) **Global Efficiency** : This is the ratio of the total number of part operations that are performed within their respective cells to the total number of operations in the system. It reflects the effectiveness of the assignment in confining the operations of parts within their respective cells.

2) **Group Efficiency** : This is the 1's complement of the ratio of the total number of external cells actually visited by the parts to the total number of maximum external cells that could be visited by them. It reflects the effectiveness of the assignment in confining the parts to as few external cells as possible.

3) **Group Technology Efficiency** : This is the 1's complement of the ratio of the inter-cell traffic in the system to the maximum possible inter-cell traffic.

The maximum possible inter-cell traffic in the system is :

$$I = \sum_{i=1}^{n} \sum_{j=1}^{q(i)} u_i \times n^j{}_i \times (s^j{}_i - 1) \tag{8}$$

For routing $r^j{}_i$ define :

$$x^j{}_i(k) = \begin{cases} 0 & \text{If operation k, k + 1 are performed in the same cell} \\ 1 & \text{otherwise} \end{cases}$$

The actual inter-cell traffic in the system is :

$$U = \sum_{i=1}^{n} \sum_{j=1}^{q(i)} \sum_{k=1}^{s^j{}_i - 1} u_i \times n^j{}_i \times x^j{}_i(k) \tag{9}$$

$$\text{Group Technology Efficiency} = 1 - \frac{U}{I} \tag{10}$$

This takes into account the sequence in which the operations are performed apart from the cell in which they are performed.

5 An Example

In this section a small example is presented to illustrate the proposed algorithm. There are 20 part types and 20 work-centers in the system.

Work-centers 6 and 7 are similar, i.e., they represent work-centers that can be interchanged. Similarly work-centers, 18, 19 and 20 are similar. Most parts have alternate routings. Figure 1 presents the details of part production volume desired over the chosen horizon of one unit of time (H=1) (column 1), the part routing number (column 2), and the global routing reference number (column 3). Finally, the matrix indicates the work-centers required by each routing (entries indicate the operation number). The routings, indicating the sequence of work-centers and processing times are presented in figure 2. For simplicity, we assume the weight of each part equal to unity. Furthermore, we assume all the machines are considered to be available for the entire period of the chosen horizon ($e_j = H = 1; j = 1,\ldots,m$).

An initial partition of the system to manufacturing cells based on the method proposed in section 3.5 is presented in figure 3. The linear programming problem, $\mathcal{P}1$ is presented in figure 4. The solution of this problem, obtained using the simplex method, leads to the selection of 22 working routings. The new manufacturing cells based on the 22 working routings are determined by solving problem $\mathcal{P}2$ by the enhanced ICTMM. The incidence matrix after the first iteration is presented in figure 5. The evaluation criteria are also calculated and presented in figure 5. Finally, the solution converges at the second iteration; 20 working routings (one for each part) are selected. The final incidence matrix is presented in figure 6. It is worthwhile indicating that the functionally similar work-centers are placed in different cells. Each part family uses an appropriate work-center as long as capacity constraints are respected.

The algorithms, coded in C, were tested on a SUN/Unix platform. Solution to problem $\mathcal{P}1$ was obtained in 22 seconds, while problem $\mathcal{P}2$ was solved in 4 seconds of c.p.u. time. The total c.p.u. time of less than a minute suggests the applicability to problems of larger dimensions. While problem $\mathcal{P}2$ is not constraining, problem $\mathcal{P}1$ imposes a constraint on the dimension of the problem that can be solved. Depending on the memory of the computer (Sun 3/60), efficient linear programming codes can address a problem of 100 machines and 500 routings.

6 Conclusions

A comprehensive algorithm that can address most practical aspects of the Group Technology problem has been presented in this paper. The proposed method selects a suitable set of working part production processes to satisfy part demand under work-center capacity constraints. It decomposes the manufacturing system into manufacturing cells with an objective of minimizing the inter-cell material movement. A simple, yet effective, decomposition of the problem is proposed that leads to a good if not optimal solution. The solution is dependent on the partition chosen initially. However, owing to its speed and quick convergence, several trials with random initial partitions can be made inexpensively. On the other hand, a method for finding a good initial partition is also suggested in the paper. Determination of part families and evaluation criteria are also presented.

The system has been tested with a wide variety of inputs of varying encroachment. Problems having a perfect decomposition can be solved optimally, but higher encroachments leads to deterioration.

The enhanced ICTMM also performs better in most cases than the one suggested in [5]. Although it is slower, it corrects less significant aggregations in a timely fashion, thereby preventing further deterioration of subsequent clusters.

The problem $\mathcal{P}1$ is of special and continual interest to a company after the manufacturing cells have been formed on the shop-floor. $\mathcal{P}1$ can be solved to determine a good set of working routings when either (i) the product mix changes, or (ii) new parts are introduced in the system, or (iii) work-center capacities change. The simplex method of solving $\mathcal{P}1$ can provide additional information about remaining capacity on work-centers (slack-variables), that could prove helpful in subsequent planning.

The major recommendations for further work include the incorporation of set-up times and batch sizes explicitly in the capacity constraints, and consideration of transportation times of material handling systems with finite capacities. Also, we recommend the employment of a linear programming method that will solve the problem even in an infeasible case, and that will indicate over-loading of work-centers rather than resulting in no solution at all. This is of special importance in the design stage of the manufacturing system, because : (i) the part production volumes considered are estimates and are subject to change, and (ii) it will help the designer in procuring new resources. Finally, the computational efficiency of the ICTMM can also be increased by avoiding redundant computations at the validation stage.

Acknowledgements

--

References

[1] Askin, R., Subramanian, S.B. (1987) : "A cost-based heuristic for group technology configuration", International Journal of Production Research, 25(1), pp. 101-113.

[2] 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.

[3] 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.

[4] 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.

[5] Harhalakis, G., Nagi, R., Proth, J.M. (1989) : "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications", accepted by the International Journal of Production Research.

[6] Harhalakis, G., Hilger, J., Nagi, R., Proth, J.M. (1989) : "Formation of Manufacturing Cell : An Algorithm for Minimizing the Inter-Cell Traffic", submitted for publication to the journal of Applied Stochastic Models and Data Analysis.

[7] King, J.R. (1979) : "Machine-Component Group Formation in Group Technology", OMEGA The International Journal of Management Science, 8(2), pp. 193-199.

[8] King J.R. (1980) : "Machine-Component grouping using ROC algorithm", International Journal of Production Research, 18(2), pp. 213-231.

[9] 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.

[10] Kusiak, A. (1985) : "The Part Families problem in Flexible Manufacturing Systems", Annals of Operations research, 3, pp. 279-300.

[11] Kusiak, A. (1987) : "The Generalized Group Technology Concept", International Journal of Production Research, 25(4), pp. 561-569.

[12] Kusiak, A., Herangu, S.S. (1987) : "Group Technology", Computers in Industry, 9, pp. 83-91.

[13] 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.

[14] McAuley, J. (1972) : "Machine Grouping for Efficient Production", The Production Engineer, Feb., pp. 53-57.

[15] Mc Cormick W.T., Schweitzer P.J., White T.E. (1972) : "Problem decomposition and data reorganization by a cluster technique", Operations Research, 20.

prod vol	Part	Rnum	Ref num	Machines																			
				01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
2.00	1	1	1	.	.	.	.	.	3	.	.	2	.	.	1	.	.	.	.	.	.	.	.
2.00	1	2	2	.	.	.	.	.	.	3	.	2	.	.	1	.	.	.	.	.	.	.	.
1.00	2	1	3	.	2	.	.	.	.	4	.	.	.	.	.	.	.	.	1	.	.	3	.
1.00	2	2	4	.	2	.	.	.	4	.	.	.	.	.	.	.	.	.	1	.	.	3	.
1.00	2	3	5	.	2	.	.	.	.	4	.	.	.	.	.	.	.	.	1	.	3	.	.
1.00	2	4	6	.	2	.	.	.	4	.	.	.	.	.	.	.	.	.	1	.	3	.	.
1.00	2	5	7	.	2	.	.	.	.	4	.	.	.	.	.	.	.	.	1	.	.	.	3
1.00	2	6	8	.	2	.	.	.	4	.	.	.	.	.	.	.	.	.	1	.	.	.	3
3.00	3	1	9	.	.	.	2	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.
2.00	4	1	10	.	3	.	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	4	2	11	.	3	.	.	2	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	5	1	12	.	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	2	1	.	.
2.00	5	2	13	.	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	2	.	1	.
2.00	5	3	14	.	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	2	.	.	1
2.00	6	1	15	3	.	.	.	.	1	.	.	.	.	.	2	.	.	.	.	.	.	.	.
2.00	6	2	16	3	.	.	.	.	.	1	.	.	.	.	2	.	.	.	.	.	.	.	.
2.00	7	1	17	.	.	.	.	1	.	4	.	.	.	.	.	.	.	.	2	.	.	3	.
2.00	7	2	18	.	.	.	.	1	.	4	.	.	.	.	.	.	.	.	2	.	3	.	.
2.00	7	3	19	.	.	.	.	1	.	4	.	.	.	.	.	.	.	.	2	.	.	.	3
2.00	7	4	20	.	.	.	.	1	4	.	.	.	.	.	.	.	.	.	2	.	.	3	.
2.00	7	5	21	.	.	.	.	1	4	.	.	.	.	.	.	.	.	.	2	.	3	.	.
2.00	7	6	22	.	.	.	.	1	4	.	.	.	.	.	.	.	.	.	2	.	.	.	3
4.00	8	1	23	.	.	3	.	.	.	.	2	.	.	1	.	.	.	.	.	.	4	.	.
4.00	8	2	24	.	.	3	.	.	.	.	2	.	.	1	.	.	.	.	.	.	.	4	.
4.00	8	3	25	.	.	3	.	.	.	.	2	.	.	1	.	.	.	.	.	.	.	.	4
2.00	9	1	26	.	.	.	.	.	.	.	.	.	3	.	.	.	2	.	.	1	.	.	.
2.00	10	1	27	1	.	.	.	.	4	.	.	2	.	.	3	.	.	.	.	.	.	.	.
2.00	10	2	28	1	.	.	.	.	.	4	.	2	.	.	3	.	.	.	.	.	.	.	.
2.00	11	1	29	.	.	.	.	.	.	.	.	.	1	.	.	.	3	.	.	4	2	.	.
2.00	11	2	30	.	.	.	.	.	.	.	.	.	1	.	.	.	3	.	.	4	.	2	.
2.00	11	3	31	.	.	.	.	.	.	.	.	.	1	.	.	.	3	.	.	4	.	.	2
2.00	12	1	32	1	.	.	.	.	3	.	.	.	.	.	2	.	.	.	.	.	.	.	.
2.00	12	2	33	1	.	.	.	.	.	3	.	.	.	.	2	.	.	.	.	.	.	.	.
3.00	13	1	34	.	.	.	2	.	.	.	.	.	.	.	.	1	.	3	.	.	.	.	.
2.00	14	1	35	2	.	.	.	.	.	.	.	1	.	.	3	.	.	.	.	.	.	4	.
2.00	14	2	36	2	.	.	.	.	.	.	.	1	.	.	3	.	.	.	.	.	4	.	.
2.00	14	3	37	2	.	.	.	.	.	.	.	1	.	.	3	.	.	.	.	.	.	.	4
3.00	15	1	38	.	.	2	.	.	.	.	1	.	.	3	.	.	.	.	.	.	4	.	.
3.00	15	2	39	.	.	2	.	.	.	.	1	.	.	3	.	.	.	.	.	.	.	4	.
3.00	15	3	40	.	.	2	.	.	.	.	1	.	.	3	.	.	.	.	.	.	.	.	4
2.00	16	1	41	.	3	.	.	.	.	2	.	.	.	.	.	.	.	.	1	.	.	.	.
2.00	16	2	42	.	3	.	.	.	2	.	.	.	.	.	.	.	.	.	1	.	.	.	.
3.00	17	1	43	.	1	.	.	3	4	.	.	.	.	.	.	.	.	.	2	.	.	.	.
3.00	17	2	44	.	1	.	.	3	.	4	.	.	.	.	.	.	.	.	2	.	.	.	.
3.00	18	1	45	.	.	1	.	.	.	.	2	.	.	.	.	.	.	.	.	.	3	.	.
3.00	18	2	46	.	.	1	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	3	.
3.00	18	3	47	.	.	1	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	.	3
2.00	19	1	48	.	.	.	.	.	.	.	.	.	2	.	.	.	3	.	.	.	1	.	.
2.00	19	2	49	.	.	.	.	.	.	.	.	.	2	.	.	.	3	.	.	.	.	1	.
2.00	19	3	50	.	.	.	.	.	.	.	.	.	2	.	.	.	3	.	.	.	.	.	1
3.00	20	1	51	.	.	.	1	.	.	.	.	.	.	.	.	2	.	3	.	.	.	.	.

Ref Num,	Sequence of Machine and processing time							
1	12	0.10	9	0.20	6	0.10		
2	12	0.10	9	0.20	7	0.10		
3	16	0.20	2	0.20	19	0.40	7	0.10
4	16	0.20	2	0.20	19	0.40	6	0.10
5	16	0.20	2	0.20	18	0.40	7	0.10
6	16	0.20	2	0.20	18	0.40	6	0.10
7	16	0.20	2	0.20	20	0.40	7	0.10
8	16	0.20	2	0.20	20	0.40	6	0.10
9	15	0.10	4	0.10				
10	6	0.10	5	0.20	2	0.10		
11	7	0.10	5	0.20	2	0.10		
12	18	0.10	17	0.20	10	0.10		
13	19	0.10	17	0.20	10	0.10		
14	20	0.10	17	0.20	10	0.10		
15	6	0.10	12	0.10	1	0.10		
16	7	0.10	12	0.10	1	0.10		
17	5	0.20	16	0.10	19	0.20	7	0.10
18	5	0.20	16	0.10	18	0.20	7	0.10
19	5	0.20	16	0.10	20	0.20	7	0.10
20	5	0.20	16	0.10	19	0.20	6	0.10
21	5	0.20	16	0.10	18	0.20	6	0.10
22	5	0.20	16	0.10	20	0.20	6	0.10
23	11	0.20	8	0.10	3	0.10	18	0.10
24	11	0.20	8	0.10	3	0.10	19	0.10
25	11	0.20	8	0.10	3	0.10	20	0.10
26	17	0.10	14	0.20	10	0.10		
27	1	0.10	9	0.20	12	0.10	6	0.10
28	1	0.10	9	0.20	12	0.10	7	0.10
29	10	0.10	18	0.20	14	0.10	17	0.10
30	10	0.10	19	0.20	14	0.10	17	0.10
31	10	0.10	20	0.20	14	0.10	17	0.10
32	1	0.10	12	0.10	6	0.20		
33	1	0.10	12	0.10	7	0.20		
34	13	0.10	4	0.10	15	0.10		
35	9	0.10	1	0.20	12	0.10	19	0.10
36	9	0.10	1	0.20	12	0.10	18	0.10
37	9	0.10	1	0.20	12	0.10	20	0.10
38	8	0.10	3	0.10	11	0.05	18	0.10
39	8	0.10	3	0.10	11	0.05	19	0.10
40	8	0.10	3	0.10	11	0.05	20	0.10
41	16	0.10	7	0.10	2	0.10		
42	16	0.10	6	0.10	2	0.10		
43	2	0.10	16	0.10	5	0.05	6	0.10
44	2	0.10	16	0.10	5	0.05	7	0.10
45	3	0.10	8	0.10	18	0.10		
46	3	0.10	8	0.10	19	0.10		
47	3	0.10	8	0.10	20	0.10		
48	18	0.10	10	0.20	14	0.10		
49	19	0.10	10	0.20	14	0.10		
50	20	0.10	10	0.20	14	0.10		
51	4	0.10	13	0.20	15	0.10		

```
Objective Function is :
+1.0*x1+2.0*x3+2.0*x4+2.0*x5+2.0*x6+2.0*x7+2.0*x8+1.0*x11+1.0*x12+1.0*x14+1.0*x15+2.0*x17+2.0*x18
+2.0*x19+2.0*x20+2.0*x21+2.0*x22+1.0*x24+1.0*x27+2.0*x29+2.0*x31+1.0*x32+1.0*x35+1.0*x36+1.0*x37
+1.0*x39+2.0*x41+1.0*x44+1.0*x46+1.0*x48+1.0*x50
 Constraint Number 1
0.1*x15+0.1*x16+0.1*x27+0.1*x28+0.1*x32+0.1*x33+0.2*x35+0.2*x36+0.2*x37 <= 1.0
 Constraint Number 2
0.2*x3+0.2*x4+0.2*x5+0.2*x6+0.2*x7+0.2*x8+0.1*x10+0.1*x11+0.1*x41+0.1*x42+0.1*x43+0.1*x44 <= 1.0
 Constraint Number 3
0.1*x23+0.1*x24+0.1*x25+0.1*x38+0.1*x39+0.1*x40+0.1*x45+0.1*x46+0.1*x47 <= 1.0
 Constraint Number 4
0.1*x9+0.1*x34+0.1*x51 <= 1.0
 Constraint Number 5
0.2*x10+0.2*x11+0.2*x17+0.2*x18+0.2*x19+0.2*x20+0.2*x21+0.2*x22+0.1*x43+0.1*x44 <= 1.0
 Constraint Number 6
0.1*x1+0.1*x4+0.1*x6+0.1*x8+0.1*x10+0.1*x15+0.1*x20+0.1*x21+0.1*x22+0.1*x27+0.2*x32+0.1*x42+0.1*x43 <= 1.0
 Constraint Number 7
0.1*x2+0.1*x3+0.1*x5+0.1*x7+0.1*x11+0.1*x16+0.1*x17+0.1*x18+0.1*x19+0.1*x28+0.2*x33+0.1*x41+0.1*x44 <= 1.0
 Constraint Number 8
0.1*x23+0.1*x24+0.1*x25+0.1*x38+0.1*x39+0.1*x40+0.1*x45+0.1*x46+0.1*x47 <= 1.0
 Constraint Number 9
0.2*x1+0.2*x2+0.2*x27+0.2*x28+0.1*x35+0.1*x36+0.1*x37 <= 1.0
 Constraint Number 10
0.1*x12+0.1*x13+0.1*x14+0.1*x26+0.1*x29+0.1*x30+0.1*x31+0.2*x48+0.2*x49+0.2*x50 <= 1.0
 Constraint Number 11
0.2*x23+0.2*x24+0.2*x25+0.1*x38+0.1*x39+0.1*x40 <= 1.0
 Constraint Number 12
0.1*x1+0.1*x2+0.1*x15+0.1*x16+0.1*x27+0.1*x28+0.1*x32+0.1*x33+0.1*x35+0.1*x36+0.1*x37 <= 1.0
 Constraint Number 13
0.1*x34+0.2*x51 <= 1.0
 Constraint Number 14
0.2*x26+0.1*x29+0.1*x30+0.1*x31+0.1*x48+0.1*x49+0.1*x50 <= 1.0
 Constraint Number 15
0.1*x9+0.1*x34+0.1*x51 <= 1.0
 Constraint Number 16
0.2*x3+0.2*x4+0.2*x5+0.2*x6+0.2*x7+0.2*x8+0.1*x17+0.1*x18+0.1*x19+0.1*x20+0.1*x21+0.1*x22
+0.1*x41+0.1*x42+0.1*x43+0.1*x44 <= 1.0
 Constraint Number 17
0.2*x12+0.2*x13+0.2*x14+0.1*x26+0.1*x29+0.1*x30+0.1*x31 <= 1.0
 Constraint Number 18
0.4*x5+0.4*x6+0.1*x12+0.2*x18+0.2*x21+0.1*x23+0.2*x29+0.1*x36+0.1*x38+0.1*x45+0.1*x48 <= 1.0
 Constraint Number 19
0.4*x3+0.4*x4+0.1*x13+0.2*x17+0.2*x20+0.1*x24+0.2*x30+0.1*x35+0.1*x39+0.1*x46+0.1*x49 <= 1.0
 Constraint Number 20
0.4*x7+0.4*x8+0.1*x14+0.2*x19+0.2*x22+0.1*x25+0.2*x31+0.1*x37+0.1*x40+0.1*x47+0.1*x50 <= 1.0
 Constraint Number 21
1.0*x1+1.0*x2 = 2.0
 Constraint Number 22
1.0*x3+1.0*x4+1.0*x5+1.0*x6+1.0*x7+1.0*x8 = 1.0
 Constraint Number 23
1.0*x9 = 3.0
 Constraint Number 24
1.0*x10+1.0*x11 = 2.0
 Constraint Number 25
1.0*x12+1.0*x13+1.0*x14 = 2.0
 Constraint Number 26
1.0*x15+1.0*x16 = 2.0
 Constraint Number 27
1.0*x17+1.0*x18+1.0*x19+1.0*x20+1.0*x21+1.0*x22 = 2.0
 Constraint Number 28
1.0*x23+1.0*x24+1.0*x25 = 4.0
 Constraint Number 29
1.0*x26 = 2.0
 Constraint Number 30
1.0*x27+1.0*x28 = 2.0
 Constraint Number 31
1.0*x29+1.0*x30+1.0*x31 = 2.0
 Constraint Number 32
1.0*x32+1.0*x33 = 2.0
 Constraint Number 33
1.0*x34 = 3.0
 Constraint Number 34
1.0*x35+1.0*x36+1.0*x37 = 2.0
 Constraint Number 35
1.0*x38+1.0*x39+1.0*x40 = 3.0
 Constraint Number 36
1.0*x41+1.0*x42 = 2.0
 Constraint Number 37
1.0*x43+1.0*x44 = 3.0
```

```
 Constraint Number 38
1.0*x45+1.0*x46+1.0*x47 = 3.0
 Constraint Number 39
1.0*x48+1.0*x49+1.0*x50 = 2.0
 Constraint Number 40
1.0*x51 = 3.0
```

use		01	07	09	12	02	05	06	16	03	08	11	18	20	04	13	15	10	14	17	19
1.00	1	.	.	2	1	.	.	3	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	2	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	3	3	.	.	2	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	4	3	1	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	5	1	.	2	3	.	.	4	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	6	1	4	2	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	7	1	.	.	2	.	.	3	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	8	1	3	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0.67	9	2	.	1	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4
0.67	10	2	.	1	3	.	.	.	.	.	.	.	4	.	.	.	.	.	.	.	.
0.67	11	2	.	1	3	.	.	.	.	.	.	.	.	4	.	.	.	.	.	.	.
1.00	12	.	.	.	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.
1.00	13	.	1	.	.	3	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0.33	14	.	4	.	.	.	1	.	2	.	.	.	.	.	.	.	.	.	.	.	3
0.33	15	.	4	.	.	.	1	.	2	.	.	.	3	.	.	.	.	.	.	.	.
0.33	16	.	4	.	.	.	1	.	2	.	.	.	.	3	.	.	.	.	.	.	.
0.33	17	.	.	.	.	.	1	4	2	.	.	.	.	.	.	.	.	.	.	.	3
0.33	18	.	.	.	.	.	1	4	2	.	.	.	3	.	.	.	.	.	.	.	.
0.33	19	.	.	.	.	.	1	4	2	.	.	.	.	3	.	.	.	.	.	.	.
1.00	20	.	2	.	.	3	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.
1.00	21	.	.	.	.	3	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.
0.17	22	.	4	.	.	2	.	.	1	.	.	.	.	.	.	.	.	.	.	.	3
0.17	23	.	.	.	.	2	.	4	1	.	.	.	.	.	.	.	.	.	.	.	3
0.17	24	.	4	.	.	2	.	.	1	.	.	.	3	.	.	.	.	.	.	.	.
0.17	25	.	.	.	.	2	.	4	1	.	.	.	3	.	.	.	.	.	.	.	.
0.17	26	.	4	.	.	2	.	.	1	.	.	.	.	3	.	.	.	.	.	.	.
0.17	27	.	.	.	.	2	.	4	1	.	.	.	.	3	.	.	.	.	.	.	.
1.50	28	.	.	.	.	1	3	4	2	.	.	.	.	.	.	.	.	.	.	.	.
1.50	29	.	4	.	.	1	3	.	2	.	.	.	.	.	.	.	.	.	.	.	.
1.00	30	.	.	.	.	.	.	.	.	2	1	3	4	.	:	.	.	.	.	.	.
1.00	31	.	.	.	.	.	.	.	.	2	1	3	.	.	.	.	.	.	.	.	4
1.00	32	.	.	.	.	.	.	.	.	2	1	3	.	4	.	.	.	.	.	.	.
1.00	33	.	.	.	.	.	.	.	.	1	2	.	3	.	.	.	.	.	.	.	.
1.00	34	.	.	.	.	.	.	.	.	1	2	.	.	.	.	.	.	.	.	.	3
1.00	35	.	.	.	.	.	.	.	.	1	2	.	.	3	.	.	.	.	.	.	.
1.33	36	.	.	.	.	.	.	.	.	3	2	1	4	.	.	.	.	.	.	.	.
1.33	37	.	.	.	.	.	.	.	.	3	2	1	.	.	.	.	.	.	.	.	4
1.33	38	.	.	.	.	.	.	.	.	3	2	1	.	4	.	.	.	.	.	.	.
3.00	49	.	.	.	.	.	.	.	.	.	.	.	.	.	2	1	3	.	.	.	.
3.00	50	.	.	.	.	.	.	.	.	.	.	.	.	.	1	2	3	.	.	.	.
3.00	51	.	.	.	.	.	.	.	.	.	.	.	.	.	2	.	1	.	.	.	.
0.67	39	.	.	.	.	.	.	.	.	.	.	.	2	.	.	.	.	1	3	4	.
0.67	40	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	3	4	2
0.67	41	.	.	.	.	.	.	.	.	.	.	.	.	2	.	.	.	1	3	4	.
0.67	42	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	3	.	2	.
0.67	43	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	.	2	1
0.67	44	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	3	.	2	.
0.67	45	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	2	3	.	.
0.67	46	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	3	.	1
0.67	47	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	2	3	.	.
2.00	48	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	2	1	.

Global Efficiency = 79.444443
Group Efficiency = 71.317833
G. T. Efficiency = 77.121208

use		01	07	09	12	02	05	06	16	20	03	08	11	18	04	13	15	10	14	17	19
2.00	2	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	4	3	1	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	6	1	4	2	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	8	1	3	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	11	2	.	1	3	.	.	.	.	4	.	.	.	.	.	.	.	.	.	.	.
2.00	12	.	.	.	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.
0.50	17	.	.	.	.	.	1	4	2	.	.	.	.	.	.	.	.	.	.	.	3
1.50	19	.	.	.	.	.	1	4	2	3	.	.	.	.	.	.	.	.	.	.	.
2.00	21	.	.	.	.	3	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.
0.25	23	.	.	.	.	2	.	4	1	.	.	.	.	.	.	.	.	.	.	.	3
0.75	25	.	.	.	.	2	.	4	1	.	.	.	.	3	.	.	.	.	.	.	.
3.00	28	.	.	.	.	1	3	4	2	.	.	.	.	.	.	.	.	.	.	.	.
3.00	32	.	.	.	.	.	.	.	.	4	2	1	3	.	.	.	.	.	.	.	.
3.00	33	.	.	.	.	.	.	.	.	.	1	2	.	3	.	.	.	.	.	.	.
4.00	36	.	.	.	.	.	.	.	.	.	3	2	1	4	.	.	.	.	.	.	.
3.00	49	.	.	.	.	.	.	.	.	.	.	.	.	.	2	1	3	.	.	.	.
3.00	50	.	.	.	.	.	.	.	.	.	.	.	.	.	1	2	3	.	.	.	.
3.00	51	.	.	.	.	.	.	.	.	.	.	.	.	.	2	.	1	.	.	.	.
2.00	40	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	3	4	2
2.00	43	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	.	2	1
2.00	46	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	3	.	1
2.00	48	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	2	1	.

Global Efficiency = 93.333336
Group Efficiency = 90.566040
G. T. Efficiency = 92.727272

use		01	07	09	12	02	05	06	16	20	03	08	11	18	04	13	15	10	14	17	19
2.00	2	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	4	3	1	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	6	1	4	2	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	8	1	3	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	9	2	.	1	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4
2.00	12	.	.	.	.	3	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.
2.00	19	.	.	.	.	.	1	4	2	3	.	.	.	.	.	.	.	.	.	.	.
2.00	21	.	.	.	.	3	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.
1.00	27	.	.	.	.	2	.	4	1	3	.	.	.	.	.	.	.	.	.	.	.
3.00	28	.	.	.	.	1	3	4	2	.	.	.	.	.	.	.	.	.	.	.	.
3.00	30	.	.	.	.	.	.	.	.	.	2	1	3	4	.	.	.	.	.	.	.
3.00	33	.	.	.	.	.	.	.	.	.	1	2	.	3	.	.	.	.	.	.	.
4.00	36	.	.	.	.	.	.	.	.	.	3	2	1	4	.	.	.	.	.	.	.
3.00	49	.	.	.	.	.	.	.	.	.	.	.	.	.	2	1	3	.	.	.	.
3.00	50	.	.	.	.	.	.	.	.	.	.	.	.	.	1	2	3	.	.	.	.
3.00	51	.	.	.	.	.	.	.	.	.	.	.	.	.	2	.	1	.	.	.	.
2.00	40	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	3	4	2
2.00	43	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	.	2	1
2.00	46	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	3	.	1
2.00	48	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	3	2	1	.

Global Efficiency = 98.507462
Group Efficiency = 97.872337
G. T. Efficiency = 98.181816