Journal of the Korean Institute of Industrial Engineers
Vol. 26, No. 1, pp. 1-8, March 2000.

Characterizing the Independent Cells by Increasing Grouping Efficiency

Moon-Soo Cho[1] · Byung-Hee Chung[2]

[1]Soongsil University, Department of Industrial / [2]Information Systems Engineering

그룹핑 효율증대를 위한 독립적 셀의 특성화

조문수[1] · 정병희[2]

We extended a minimum spanning tree algorithm (Cho *et al.*, 1997) by characterizing the mutually independent cells with maximizing the grouping efficiency referring to few propositions developed by Shu, 1990 in cellular manufacturing system. Each row of the machine-part incidence matrix is regarded as a node in a graph, and a distance function is defined for every pair of nodes. It shows that there are K mutually independent cells in the cellular manufacturing system if only if there are K-1 arcs of length 1 in the minimum spanning tree of the graph, and gives an effective policy for sub-cell formation from larger cells.

1. Introduction

Group Technology (GT) has been recognized as one of the key factors to improve productivity of manufacturing systems. Despite numerous economic benefits and operational advantages offered by the GT concepts, its real potential has not been fully explored. A number of factors, including vulnerability to machine breakdown, under-utilization of resources and eventual unbalanced workload distribution in a multi-cell plant in manufacturing system, pose some problems when using GT concept.

These problems mainly stem from somewhat standard principles of GT, such as the avoidance of interaction between the machine cells and part families, and tendency to setting up permanent idealistic cells which do not exist any bottleneck situations, etc.

For detailed review of GT and its advantages, see Fazakerlay (1974), Ham *et al.* (1985), and Gallager and Knight (1986). A recent survey on the US firms that adopted GT showed that most of these forms have made substantial improvement in various areas which include throughput time, work in process, material handling, fixture for cell parts, setup time, space, part quality, inventory, labor cost, job satisfaction, etc.

One of the frequently used representations of the GT problem is a machine-part incidence matrix $[a_{ij}]$ which consists of '1' (empty) entries, where an entry 1 (empty) indicates that machine i is used (not used) to process part j. Typically, when an initial machine-part incidence matrix $[a_{ij}]$ is constructed, clusters of machines and parts are not visible. A clustering algorithm allows transforming the initial incidence matrix into a structured (possible block diagonal) form. To illustrate the clustering approach to GT, consider the machine-part incidence matrix (1).

Rearranging rows and columns in matrix (1) results in matrix (2). Two machine cells or clusters MC-1 = {2,4} and MC-2 = {1,3}, and two corresponding part families PF-1={1,3} and PF-2 ={2,4,5} are visible in matrix (2).

Clustering of a binary incidence matrix may result in the form of mutually separable clusters or partially separable clusters.

Mutually separable clusters are shown in matrix (2), while partially clusters are presented in matrix (3).

$$[a_{ij}] = \begin{array}{c|ccccc|l} & \multicolumn{5}{c}{\text{Part Number}} & \\ & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & & 1 & & 1 & 1 & \\ 2 & 1 & & 1 & & & \text{Machine} \\ 3 & & 1 & & 1 & & \text{Number} \\ 4 & 1 & & 1 & & & \end{array} \qquad (1)$$

$$\begin{array}{cc|cc|ccc|} & & \multicolumn{2}{c}{\text{PF-1}} & \multicolumn{3}{c}{\text{PF-2}} \\ & & 1 & 3 & 2 & 4 & 5 \\ \hline \text{MC-1} & 2 & 1 & 1 & & & \\ & 4 & 1 & 1 & & & \\ \hline \text{MC-2} & 1 & & & 1 & 1 & 1 \\ & 3 & & & 1 & 1 & 1 \\ \hline \end{array} \qquad (2)$$

Matrix (3) cannot be separated into two disjoint clusters because of part 5 from matrix (3) results in the decomposition of matrix (3) into two separable machine cells, MC-1={1,2} and MC-2={3,4} and two part families PF-1={1,2} and PF-2={3,4}. The two clusters are called partially separable clusters and the overlapping part is called a bottleneck part i.e. the part that is processed on machines belonging to more than one machine cell.

$$\begin{array}{cc|cc|cc|c|} & & \multicolumn{2}{c}{\text{PF-1}} & \multicolumn{2}{c}{\text{PF-2}} & \\ & & 1 & 3 & 2 & 4 & 5 \\ \hline \text{MC-1} & 2 & 1 & 1 & & & 1 \\ & 4 & 1 & 1 & & & \\ \hline \text{MC-2} & 1 & & & 1 & 1 & \\ & 3 & & & 1 & 1 & 1 \\ \hline \end{array} \qquad (3)$$

However, ways to handle a bottleneck part is to use an alternative process plan which is frequently available for many parts, to do subcontract, and duplicate the bottleneck part. For example, assuming that an alternative process plan for part 5 in matrix (3) involves machine 2 and 4 would result in two mutually separable machine cells. Alternative process plans are frequently available for many parts. In practical cases, not all components of a part family can always be processed within a single cell. It is obvious that grouping of parts with alternative routes increases the likelihood of generating ideal machine cells.

The components having operations in more than one cell are called exceptional parts, and the machines processing them are referred to as bottleneck machines. The traditional transportation of exceptional parts between cells can be eliminated by assigning a sufficient number of bottleneck machines to appropriate cells. Analogous to the bottleneck part, a bottleneck machine is defined. A bottleneck machine is a machine that processes parts belonging to more than one cell, i.e. it does not allow for decomposition of a machine-part incidence matrix into disjoint submatrices.

For example, machine 3 in matrix (4) does not permit decomposition of that matrix into two machine cells and two part families. A way to handle a bottleneck machine is to allocate the bottleneck machines to the appropriate machine cells by examining the similarity between bottleneck machines and structured machine cells. It results in the efficient facilitation of material flows and easy to implement for controlling the sizes of the machine cells. It may increase the grouping efficiency.

$$\begin{array}{ll|cccccc|} & & \multicolumn{6}{c}{\text{Part Number}} \\ & & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & 1 & 1 & & & & \\ \text{Machine} & 2 & 1 & 1 & & & & \\ \text{Number} & 3 & 1 & 1 & 1 & & 1 & 1 \\ & 4 & & & 1 & 1 & 1 & 1 \\ & 5 & & & 1 & & 1 & 1 \\ \hline \end{array} \qquad (4)$$

$$\begin{array}{cc|cc|cccc|} & & \multicolumn{2}{c}{\text{PF-1}} & \multicolumn{4}{c}{\text{PF-2}} \\ & & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline & 1 & 1 & 1 & & & & \\ \text{MC-1} & 2 & 1 & 1 & & & & \\ & 3(1) & 1 & 1 & & & & \\ \hline & 3(2) & & & 1 & & 1 & 1 \\ \text{MC-2} & 4 & & & 1 & 1 & 1 & 1 \\ & 5 & & & 1 & & 1 & 1 \\ \hline \end{array} \qquad (5)$$

A way to decompose matrix (4) into two disjoint submatrices is to use an additional copy of machine 3. The latter leads to the transformation of matrix (4) into matrix (5). Two machine cells, MC-1 ={1,2,3(1)} and MC-2={3(2),4,5} and two corresponding part families, PF-1={1,2} and PF-2 ={3,4,5,6} are shown in matrix (5).

Different formulations of the GT have been attempted. The quadratic assignment formulation, see Kusiak and Vannelli (1986), and the traveling salesman formulation, see Lenstra (1973), are among the most well-known ones applied integer programming. However, these are difficult to solve and there is no guarantee that optimal solution can be obtained in reasonable amount of CPU time.

Another class of algorithm is the hierarchical ordering method developed by McAully (1972), which requires to choose a threshold value for partitioning the machine cells. Another widely used one is the permutation of rows and columns of the machine-part incidence matrix shown in the conclusion section.

These are short coming in each of these different existing classes of formulations and algorithms. The integer programming is difficult to solve, and the choice of the threshold value in the hierarchical ordering methods is arbitrary. The non-hierarchical approach and the row and column permutation method do not make explicit use of the intrinsic properties between the machines. They implicitly assume that if there exists some natural grouping of these machines, then these grouping will appear eventually in the final solution, even though no proof of this has ever been done.

The ideal grouping for a given manufacturing system will be such that the machines and the parts can be partitioned into mutually independent cells of the suitable sizes with no intercell flow of parts between different machine cells. Though this ideal situation rarely occurs in practical situation, one would like to know what are the characteristics exhibited by these independent cells if they do exist, see Cho *et al.* (1997).

Therefore, the efficient way to identify these independent cells and effective policy to guide to partition the larger cells into smaller cells are proposed in this paper.

2. Definition and Notation

A clustering algorithm allows transforming the initial incidence matrix into a structured (possible block diagonal) form. Thus the grouping of machines and parts into families is equivalent to rearrange the columns and rows of A so that there are as many non-zero entries being grouped into clusters as possible. The following notations are used:

m = number of rows in A, *i.e.*, the number of machines
n = number of columns in A, *i.e.*, the number of parts
$M = \{1, 2, \ldots, m\}$, index set of rows
$N = \{1, 2, \ldots, n\}$, index set of columns
$M_i = \{j \in N\colon a_{ij} = 1\}$, (the parts that will visit machine i)
$N_j = \{i \in M\colon a_{ij} = 1\}$, (the machines that will process part j)
$|S|$ = number of elements in S
K = a positive integer number, where $1 \leq K \leq m$, and $1 \leq K \leq n$

Let R_i, $i=1,2,\ldots,K$ be a partition of M and C_i, $i=1,2,\ldots,K$ be a partition of N, which is called R_i is machine cells and C_i is part families. If the rows and the columns of A is arranged according to the order they appear in R_i and C_i respectively, it calls (R_i, C_i) $i=1,2,\ldots,K$, a diagonal block decomposition of A, and (R_i, C_i) is called the ith diagonal block or the ith cell, see Shu (1987).

Let e_1 = total number of 1's in the diagonal blocks, e_0 = total number of 1's not in the diagonal block, and

$$d_1 = \sum_{i=1}^{K} |R_i| \, |C_i|, \quad d_0 = mn - d_1$$

The cells (R_i, C_i) i=1,2,...,K are to be mutually independent (or the diagonal blocks are mutually separable) if e_0 is zero. If a manufacturing system can be decomposed into mutually independent cells, then there will be no material flow between different cells.

For a given mutually separable diagonal block decomposition of A, say (R_i, C_i), i=1,2,...,K; if K is the maximum number possible, *i.e.*, any further decomposition will result in non-zero entries off the diagonal blocks, Shu (1987) denotes (R_i, C_i), $i=1,2, \ldots, K$, the maximum mutually separable diagonal block decomposition of A, or simply maximum separable decomposition.

Let S, T be two sets, the symmetric difference of S, T is defined as follows:

$$S \triangle T = (S - T) \cup (T - S) = (S \cup T) - (S \cap T).$$

Then the distance between two rows i,j M can be defined by $d(i,j) = |M_i \triangle M_j| \;/\; |M_i \cup M_j|$.

For a given diagonal block decomposition, (R_i, C_i),

$i=1,2,\ldots,K$, the distance between two different row sets (machine cells) R_h, R_k of M is defined as follows:

$$d(R_h,\ R_k)\ =\ \min\ \{d(i,j)\ :\ i \in R_h,\ j \in R_k\}.$$

Notice that if $R_h \cap R_k \neq \phi$, then $d(R_h,\ R_k) = 0$. A weight function for each column with respect to the machine cell is defined:

$$\mathrm{W}(j,\ R_i)\ =\ \sum_{h \in R_i} ah_j\ /\ \sum_{h=1}^{m} ah_j,$$

$i=1,2,\ldots,K$, $j=1,2,\ldots,n$. Thus a weight function is the function of nonzero entries of column j in machine group R_i.

3. Propositions

The minimum spanning tree (MST) algorithm will decompose A into diagonal blocks, and characterize the mutually independent cells using the spanning tree obtained by the MST algorithm in Cho *et al.* (1997), and the following propositions by Shu, 1990 are needed.

Proposition 1. Let $(R_i,\ C_i)$, $i=1,2,\ldots,K$, be a mutually separable diagonal block decomposition of A (not necessary maximum), then:

(a) $d(R_i,\ R_j)\ =\ 1$ for all $i \neq j$.

(b) If the decomposition is maximum mutually separable, then for any $h \in R_i$ ($i=1,2,\ldots,K$), there exists $k \in R_i$ such that $d(h,\ k)\ <\ 1$. In particular, if S_1, S_2 is a partition of R_i, then $d\ (S_1,\ S_2)\ <\ 1$.

Proof.

(a) Let $h \in R_i$, $k \in R_j$, $i \neq j$. Since $(R_i,\ C_i)$, $(R_j,\ C_j)$ are mutually separable, $M_h \cup M_k \neq \phi$, hence, $M_h \triangle M_k = M_h \cup M_k$, we have $d(h,\ k)=1$ for all $h \in R_i$, $k \in R_j$ which gives $d(R_i\ R_j)=1$.

(b) If the decomposition is maximum mutually separable, then each block $(R_i,\ C_I)$ cannot be further decomposed into mutually separable diagonal blocks. Let $h \in R_i$, the decomposition $\{h\}$, $R_i - \{h\}$ and the corresponding decomposition of the column set C_i will give non-zero entries in the submatrix of $(R_i,\ C_i)$. For the non-zero off diagonal block entries of row h, there exists a column k such that $a_{pk}=1$ for some $p \in R_i - \{h\}$. Otherwise the following will be a mutually separable decomposition of $(R_i,\ C_i)$: $(\{h\},\ C_i\,(h)$, $(R-\{h\},\ C_i - C_i\ (h))$ where $C_i\ (h)=\{j \in C_i\!:\ a_{hj}=1\}$, where non-zero on diagonal block entries of row h, there exists a column j such $h \in R_j - \{C\}$. Therefore $M_h \cap M_p \neq \phi$, thus $d(h,p)<1$. The argument for $d(S_1,S_2) <1$ follows similarity.

Proposition 2. Suppose at certain iteration of the MST algorithm developed by Cho, *et al.*, 1997, $\pi_k = \min\ \{\ \pi_j\ :\ j \in M-V\ \}=1$ can be performed, then, $d(V,\ M\text{-}V)=1$, where a tree of G is a subgraph $(V,\ H)$ of G with no cycles where V are included in a set of nodes, and H is included in a set of arcs in the graph, see Minieka (1978).

Proof. Let $h \in V$ be such that $d(h,\ k)=1$. By choice of k, $\pi_j \geq \pi_k$ for all $j \in M-V$, and $d(i,j) \geq \pi_j \geq \pi_k = 1$ for all $i \in V$, $j \in M-V$. Hence $d(V,\ M-V)=1$.

Proposition 3. Suppose the minimum spanning tree T is decomposed into K components (subtrees) $G_i=(R_i,\ T_i)$, i=1,2,...,K, by deleting K-1 arcs in T, all of which have arc length of 1. Then for each column $k \in N$, we have $\mathrm{w}\ (k,\ R_i)\ =\ 0$ or 1 for i=1,2,...,K; and there exists exactly one R_j such that $w(k,\ R_j)=1$ and $w(k,\ R_h)=0$, $h \neq j$.

Proof. By definition of w, $\sum_{i=1}^{k}\ \mathrm{w}\ (j,\ R_i)=1$ for all $j \in N$. If there exists row set R_p such that $0\ <\ w\ (u,\ R_p)\ <\ 1$ for some $u \in N$, which implies that there exists a row $s \in R_p$ that $a_{su}=1$, then another row set R_r such that $0\ <\ w(u,\ R_r)\ <\ 1$, where r $\neq p$, $r \in \{1,2,\ldots,k\}$ can be found, which implies that there exists a row $t \in R_r$ such that $a_{tu}=1$, hence $M_s \cap M_t \neq \phi$ and $d(s,t) <1$, thus $\mathrm{d}(R_p,\ R_r)<1$. However, this contradicts $d(R_p,\ R_r)=1$ for i $\neq j$ by proposition 1. Therefore, $w\ (\ k,\ R_i\ )=0$ or 1 for all $i \in \{1,\ldots,k\}$. By $\sum_{i=1}^{k}\ w(j,\ R_i\ )=1$ for all $j \in N$, there exists exactly one $i \in \{1,\ldots,k\}$ such that $w\,(k,\ R_j\ )=1$.

Proposition 4. If there exists a maximum of K mutually separable diagonal blocks in the machine-part incidence matrix A, then the MST algorithm correctly identifies these K mutually separable diagonal blocks.

Proof. Let B_i denote the row set of block I, ($i=1,2,..,k$) of these mutually separable diagonal blocks. Without loss of generality, let the initial row h

chosen at the beginning of the algorithm be in B_i, by Proposition 1(b), there exists $k \in B_1\text{-}\{h\}$ such that $d(k,h)<1$. If $j \in M\text{-}B_1$, then $d(j,h)=1$. Also from Proposition 1(b), $d(B_1\text{-}V)<1$ for any proper subset V of B_1 is shown. Thus $|B_1|\text{-}1$ iterations, all nodes adding to V will be from the set B_1 and $\pi_k<1$ for each $k \in B_1\text{-}V$.

When $V=B_1$, the next incoming node will be $k \in M\text{-}B_1$ with $\pi_k=1$ by Proposition 2. Let the label on this k be $[\pi_k, p]$, and one arc (k,p) of length 1 exists. Without loss of generality, assume $k \in B_2$, then in the next $|B_2|\text{-}1$ iteration, all the nodes j that are added to V will be from the set of B_2, with $\pi_j<1$ if $j \neq k$. Then, there will be exactly K-1 arcs of length of 1 in the MST by using induction. By deleting these K-1 edges, K subtrees $G_j=(R_j, T_j)$, $i=1,2,..K$ can be obtained, and it is not difficult to see that the vertex set R_j, i=1,2,..K are the row set B_j, $i=1,2,..K$, different only in the ordering.

By Proposition 3, define $C_j=\{j \in N: d(j,R_j)=1\}$, the column set N is partitioned into K mutually disjoint sets, and hence (R_j, C_j), $i=1,2,...K$, form K mutually separable diagonal blocks of given matrix.

Proposition 5. The machine-part incidence matrix A has a maximum separable decomposition of K diagonal block if only if there are exactly K-1 arcs of unit length in the minimum spanning tree obtained by the MST algorithm.

4. Grouping Efficiency

Using the grouping efficiency η as an objective function for maximization has the advantage of simultaneously maximizing the utility of the diagonal block entries and minimizing the intercell flows, where q is the weight assigned to the block entry usage and $(1-q)$ will be the weight for the zero entry usage in the off diagonal blocks.

To measure the quality of clusters, the following grouping efficiency measure has been introduced as $GE = qn_1 + (1-q)n_2$, where n_1 = number of entries "1" in the diagonal blocks / total number of elements in the diagonal blocks, n_2 = number of entries "0" in the off-diagonal blocks / total number of elements in the off-diagonal blocks, and q = a weight factor having a value between 0 and 1. This function is non-negative and its range is zero to one.

However, there is no good way of selecting the value of q published in the literature. If the value of GE is one means that the matrix has a perfect block diagonal form, and zero is the opposite case, see Kumer and Chandrasekharan. (1990).

The grouping efficiency for the decomposition (R_i, C_i), $i=1,2,...,K$, is given by $\eta = q\,(e_1/d_1) + (1-q)(1-e_0/d_0)$. The objective is to find a diagonal block decomposition of A which maximize the grouping efficiency. The grouping efficiency objective function was first proposed by Chandrasekharan and Rajagopalan (1987).

Let (R_i, C_i) be a diagonal block decomposition of A, not necessary mutually separable. If the current decomposition has satisfied given requirements, it may stop. Otherwise, it can be further decomposed some of the existing blocks to form smaller blocks. Without loss of generality, let R_1 be too large and we want to further decompose it into smaller machine cells. Therefore, the following proposition states a necessary and sufficient condition of what type of decomposition will give a higher grouping efficiency.

Proposition 6. Suppose (R_1, C_1) is decomposed into two diagonal blocks $(R_i1(j), C_i(j))$, j = 1, 2; then the new grouping $(R_1(j), C_1(j))$, j = 1,2; (R_i,C_i), i = 2, 3, .., K, will have greater grouping efficiency if only if,

$q\{\alpha d_1 - \beta e_1 / d_1 (d_1 - \beta)\} < (1-q)\{\beta e_0\text{-}\alpha d_0 / d_0 (d_0 + \beta\}$,

where $\alpha = \sum_{j \in C1(1)} \mathrm{w}(j, R_i(2))\, |N_j| + \sum_{j \in C1(2)} w(j, R_i1(1))\, |N_j|$,

and $\beta = |R_1(1)|\ |C_1(2)| + |R_1(2)|\ |C_1(1)|$

Proof. The grouping efficiency of the diagonal block decomposition $(R_i,C_i), i=1,2,...K$, is $\eta = q\,(e_1/d_1) + (1\text{-}q)(1\text{-}\ e_0/d_0)$. First observe that a is the number of non-zero entries in the off diagonal blocks of $(R_1(j), C_1(j))$, $j=1,2$ with respect to the block (R_1,C_1).

For the new diagonal block decomposition, $(R_1(j), C_1(j))$, $j=1,2$; $(R_i,C_i), i=1,2,...K$, the number of non-zero entries inside the diagonal blocks is $e_1\text{-}\ a$, and that outside the diagonal blocks is e_0+a. Thus the new grouping efficiency can be defined as follows:
$\eta' = q((e_1-\alpha)/(d_1-\beta)) + (1-q)(1-(e_0+\alpha)/(d_0+\beta))$. $\eta' > \eta$ if and only if $q((e_1-\alpha)/(d_1-\beta)) + (1-q)(1-(e_0+\alpha)/(d_0+\beta)) > q(e_1/d_1) + (1-q)(1-e_0/d_0)$, simplifying those equations, $q\{\alpha d_1 - \beta e_1 / d_i$

$(d_1 - \beta)\} < (1-q)\{\beta e_0 - \alpha d_0 / d_0 (d_0 + \beta\}$ can be defined.

If (R_p, C_p) is a diagonal block, recall that (R_p, T_p) is a subtree, there are exactly $|R_p| - 1$ arcs in (R_p, T_p). By deleting any one arc in $(R_p\ T_p)$, R_p is partitioned into two disjoint sets, say $R_p(j)$, $j = 1, 2$. For each part $k \in C_p$, recall that the weight of k with respect to these machine cells is given by $W(k, R_p(j)) = \sum_{i \in R_p(j)} a_{ik} / |N_k|$, $j = 1, 2$.

Part k is assigned to the subgroups which gives the greater weight. The following formally states the subtree partition procedure. Let Rp be the machine cell that one want to test and the corresponding subtree be given by (R_p, T_p). Then the subtree partition procedure will be as follows:

Step 1. Use proposition 6 to identify the arc in the subtree (R_p, T_p) which gives the greatest increase in grouping efficiency. If no arc is found, return: else go to step 2.

Step 2. Let $R_p(1)$, $R_p(2)$ be the machine cell obtained after the deletion of the arc identified in step 1. For each part $k \in C_p$, determine the weight of k with respect to the two machine cells $R_p(j)$, $j=1,2$. If $w(k, R_p(1)) \geq w(k, R_p(2))$, assign k to $C_p(1)$, else assign k to $C_p(2)$, and return.

The revised MST algorithm increasing for the grouping efficiency will be follows:

Step 1. Use the MST algorithm developed by Cho, *et. al.*, 1997 to find the minimum spanning tree.

Step 2. Identify the set of arcs with length 1 in the spanning tree obtained in step 1. Delete all these arcs and form the machine cells and part families as in step 3 of the original MST algorithm. Notice that the cells identify in this step are mutually independent.

Step 3. If all machine cells are of the desirable sizes or no subtree partition which increase the grouping efficiency can be identified, stop; otherwise use the subtree partition procedure to decompose the larger cells into smaller cells.

5. Illustrative Example

We will apply the revised MST algorithm to solve an example in Burbidge (1963), which original matrix is shown in <Figure 1>. Applying the proposed MST algorithm to solve the problem, the minimum spanning tree of the problem is given in <Figure 1>, and the length of the arcs in the spanning tree is given in <Table 1>.

Table 1. Arc length of the arcs in the minimum spanning tree in <Figure. 1>

Arc	Length	Arc	Length
(1, 11)	1.000	(15, 12)	0.375
(1, 8)	0.444	(8, 7)	0.444
(1, 20)	0.889	(8, 3)	0.222
(11, 16)	0.111	(8, 17)	0.000
(16, 15)	0.125	(3, 18)	0.923
(3, 19)	0.125	(20, 9)	0.333
(18, 2)	0.222	(9, 10)	0.167
(2, 4)	0.333	(10, 6)	0.143
(2, 14)	0.000	(6, 5)	0.429
(4, 13)	0.333		

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1	1		1																	1			1		1										
2		1					1			1		1	1					1						1			1				1				
3	1	1	1												1		1												1		1				
4		1					1					1	1											1			1								
5								1						1		1																		1	
6								1						1		1			1			1				1								1	
7	1		1		1										1		1			1			1												
8	1		1		1										1		1			1			1		1				1						
9								1						1					1			1				1									
10								1						1		1			1			1				1									
11				1		1			1		1										1							1		1		1			1
12				1		1			1		1										1												1		
13		1										1	1											1											
14		1					1			1		1	1				1							1			1				1				
15				1		1			1		1										1							1		1					
16				1		1			1		1										1							1		1		1			
17	1		1		1										1		1			1			1		1				1						
18		1								1		1	1					1						1							1				
19				1					1		1										1							1		1		1			
20								1						1					1				1			1									

Figure 1. Machine- part incidence matrix.

The average arc length of this spanning tree is 0.346. There are 7 arcs in the spanning tree with arc lengths greater than 0.346. However, only 3 of these 7 arcs are substantially greater than the average arc length, they are (1, 11), (3, 18), (1, 20).

There is only one arc in the minimum spanning tree with arc length equal to 1, hence by the Proposition 4 and 5, there are only two mutually independent cells in the manufacturing system.

If we take q = 0.2, then the grouping efficiency attained by the grouping will be 0.864. Using the Proposition 6, there are all together 7 arcs whose deletion from the current subtree will give new grouping with higher grouping efficiency. The arc which gives the highest is (3, 18). Deleting (3, 18) from the subtree, there are three machine cells and part families, and the new grouping efficiency is 0.904.

Applying the Proposition 6 again, there are 6 arcs whose deletion will give a higher grouping efficiency than 0.904. The one which gives the highest is (1, 20). The grouping efficiency of this decomposition is 0.950. There are four machine cells and part families and $e_0 = 2$.

Applying the Proposition 6 again, we find no more arc in the subtree whose deletion will give rise to a new grouping with higher grouping efficiency. Hence the decomposition given in <Figure 2> is the final matrix.

	1	3	5	15	17	20	23	25	29	4	6	9	11	21	28	30	32	33	35	2	7	10	12	13	18	24	27	31	8	14	16	19	22	26	34
1	1	1				1	1	1																											
3	1	1	1	1	1				1																			1							
7	1	1	1	1	1	1	1																												
8	1	1	1	1	1	1	1	1	1																										
17	1	1	1	1	1	1	1	1	1																										
11										1	1	1	1	1	1	1	1		1																
12										1	1	1	1	1				1																	
15										1	1	1	1	1	1	1																			
16										1	1	1	1	1	1	1	1																		
19										1		1	1	1	1	1	1																		
2																				1	1	1	1	1	1	1	1	1							
4																				1	1		1	1		1	1								
13																				1			1	1		1									
14																				1	1	1	1	1	1	1	1	1							
18																				1		1	1	1	1	1		1							
5																													1	1	1				1
6																													1	1	1	1	1	1	1
9																													1	1		1	1	1	
10																													1	1	1	1	1	1	
20							1																						1	1		1		1	

Figure 2. Final matrix with $q = 0.2$, = 0.950, $e_0 = 2$.

A first glance at Proposition 6 will suggest that it can be the right tools in hand, since it gives the criterion to look for the kind of decomposition of a diagonal block that will increase the grouping efficiency. However, it is not too useful because there is an exponential number of ways to decompose the diagonal block into two different diagonal blocks. Therefore only those partitions as indicated by the subtrees was considered in the proposed algorithm.

6. Conclusion

In this paper, a heuristic approach for solving the group technology problem is proposed with a minimum spanning tree algorithm. The independent cell has been characterized completely in manufacturing systems.

Necessary and sufficient conditions for subtree decomposition which may increase grouping efficiency by characterizing the independent cells is also discussed with propositions by Shu, 1990.

If the number of machines in machine group is too large, it can be chosen to delete a suitable arc from the subtree associated with machines in machine group to form smaller machine groups.

To solve the matrix formulation of the group technology problem, Seifoddini and Wolf (1986), and Kusiak and Cho (1992) developed similarity coefficient methods, and King (1980) and Chan and Milner (1982) developed sorting-based algorithms. The bond energy algorithms have been developed by McCormick, *et al.* (1972), Slagle, *et al.* (1975), and Bhat and Haupt (1976), and cost-based methods have been developed by Askin and Subramanian (1987), and Kusiak and Chow (1987).

The extended cluster identification algorithm by Kusiak (1990), the within-cell utilization based heuristic by Ballakur and Steudel (1987), and the non-hierarchical clustering algorithm has been developed by Chandrasekharan and Rajagopalan (1987).

However, the proposed algorithm in this paper shows the detailed graph theoretical heuristic approach which results in the strong grouping efficiency. Although the proposed approach cannot be more efficient than the other existing approaches regarding to complexity, it can offer more flexibility than most existing heuristic algorithms for the group technology problem.

Another important feature of this paper is the ease of the control of the size of the machine groups. If the number of machines is too large, it can be chosen to delete appropriate arcs from the subtree associated with machines in machine group to form smaller machine groups. Hence, it is efficient and easy to implement for controlling the sizes of the machine cells by allocating the bottleneck machines into the appropriate machine cell. Thus it offers more flexibility than most existing heuristic algorithms for the group technology problem do.

References

Askin, R. and Subramanian, S. (1987), A cost-based heuristic for group technology configuration, *International Journal of*

Production Research, 25(1), 101-114.

Ballakur, A. and Steudel, H. J. (1987), A within-cell utilization based heuristic for designing cellular manufacturing systems, *International Journal of Production Research*, 25(5) 639-665.

Bhat, M.V. and Haupt, A. (1976), An efficient clustering algorithm, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6, 1, 61-64..

Burbidge, J.L. (1963). Production flow analysis, *Production Engineer*, 42, 742.

Chan, H.M. and Milner, D. A. (1982), Direct clustering algorithm for group formation in cellular manufacture", *Journal of Manufacturing Systems*, 1(1), 63-74.

Chandrasekharan, M. P. and Rajagopalan, R. (1987). ZODIAC - An algorithm for concurrent formation of part-families and machine-cells, *International Journal of Production Research*, 25(6) 835-850.

Cho, M., Chung, B. and Yoon, C. (1997). A heuristic approach solving the bottleneck machine problem in group technology manufacturing systems, *Journal of Control,Automation and Systems Engineering*, 3(5), 503-510.

Fazakerlay, G. M. (1974), Group Technology : Social benefits and social problems, *The Production Engineer*, 383-386.

Gallager, C.C. and Knight, W.A. (1986), *Group Technology : Production Methods in Manufacture*, New York, John Wiley.

Ham, I., Hitomi, K. and Yoshida, T. (1985), *Group Technology : Application to Product Management* , Kluwer-Nijhoff, Boston, MA.

King, J. R. (1980), Machine-component group formation in group technology, *The International Journal of Management Science*, 8(2), 193-199.

Kumer, C. S. and Chandrasekharan, M. P. (1990). Grouping Efficacy : A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, *International Journal of Production Research*, 28(2), 233-243.

Kusiak, A. (1990), *Intelligent Manufacturing Systems*, Englewood Cliffs, NJ: Prentice Hall.

Kusiak, A. and Chow, W. S. (1987), Efficient solving of the group technology Problem, *Journal of Manufacturing Systems*, 6(2), 117-124.

Kusiak, A. and Vannelli, A. (1985), Grouping problem in scheduling flexible manufacturing systems, *Robotica*, 3, 245-252.

Lenstra, J. K. (1973), Clustering a data array and the traveling salesman problem, *Operation Research*, 22, 413-414.

McCormick, W. T., Schweitzer, P. J. and White, T. E. (1972), Problem decomposition and data reorganization by a cluster technique, *Operations Research*, 53, 993-1009.

Minieka, E. (1978), *Optimization Algorithms For Networks and Graphs*, Marcel Dekkar. Inc., New York.

Seifoddini, H. and Wolfe, P. M. (1986), Application of the similarity coefficient method in group technology, *IIE Transactions*, 18(3).

Shu, M. N. (1990), Characterizing the independent cells in group technology, *Working Paper*, University of Southern California.

Shu, M. N. (1987), A new spanning tree algorithm for group technology problems, *Working Paper*, University of Southern California.

Slagle, J. L., Chang, C. L. and Heller, S. R. (1975), A clustering and data reorganization algorithm, *IEEE Transactions, on System, Man, and Cybernetics*, SMC-5, 125-128.