A Heuristic Approach Solving the Bottleneck Machine Problem in Group Technology Manufacturing Systems

예외적 원소의 재배치를 고려한 그룹테크놀로지 접근방법

Moonsoo Cho, Byunghee Chung and Changwon Yoon
(조 문 수, 정 병 희, 윤 창 원)

요 약 : 기존에 있는 그룹테크놀로지의 방법들은 기계와 부품간의 공정유사성이나 그밖의 다른 정의의 유사성을 갖고 한 가지 정도의 알고리즘이나 수학적인 모델로써 해를 구했으나 그 해들은 실질적으로 현대 산업의 생산시스템이나 자동화에서 풀어나가야 하는 기계들의 병목현상을 풀어나가는 직접적인 접근방법을 제시하지 못했다. 따라서 본 연구에서는 새로운 기계간 유사성의 개발과 그래프이론을 응용한 알고리즘 그리고 수학적인 모델을 동시에 응용해서 병목현상을 초래하는 예외적 원소의 재배치를 효과적이고 쉬운방법으로 제시하였다.

Keywords : group technology, minimum spanning tree, similarity measure, bottleneck machine, grouping efficiency, graph theoretic algorithm

I. 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. The basic approach to solve the GT problem is to classify the parts into families according to the similarity of operations, and the machines are grouped into machine cells according to the parts that they have to process, since the formation of machine cells and part families is the first step towards the design of cellular manufacturing systems.

For detailed review of GT and its advantages, see Fazakerlay [1], Ham et al. [2], and Gallagher and Knight [3]. However, a common theme among all these objective function is to group the machines and parts in such a way that the intercell flow of parts is as little as possible.

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).

$$\begin{array}{c} \text{Part Number} \\ \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \end{array}$$

$$[a_{ij}] = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \left[\begin{array}{ccccc} & 1 & & 1 & 1 \\ 1 & & 1 & & \\ & 1 & & 1 & \\ 1 & & 1 & & \end{array} \right] \text{Machine Number} \qquad (1)$$

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).

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

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.

접수일자 : 1997. 3. 28., 수정완료 : 1997. 8. 11.
조문수, 정병희 : 숭실대학교 산업공학과
윤창원 : 쌍용정보통신 SI Division. Consultant

		PF-1: 1	PF-1: 3	PF-2: 2	PF-2: 4	PF-2: 5
MC-1	2	1	1			1
MC-1	4	1	1			
MC-2	1			1	1	
MC-2	3			1	1	1

(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. Therefore, the new approach solving the bottleneck part is proposed in a different way such that, if the bottleneck part should be assigned to a certain machine cell, the bottleneck part should be assigned to an appropriate machine cell which may increase the grouping efficiency based on the new similarity coefficient between bottleneck part and the 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 as 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.

Machine Number \ Part Number	1	2	3	4	5	6
1	1	1				
2	1	1				
3	1	1	1		1	1
4			1	1	1	1
5			1		1	1

(4)

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).

To solve the matrix formulation of the group technology problem, the following heuristic approaches have been developed: similarity coefficient methods [4], sorting-based algorithms [5][6], bond energy algorithms [7][8][9], cost-based methods [10][11], extended cluster identification algorithm [12], within-cell utilization based heuristic [13], non-hierarchical clustering algorithm [14].

		PF-1: 1	PF-1: 2	PF-2: 3	PF-2: 4	PF-2: 5	PF-2: 6
MC-1	1	1	1				
MC-1	2	1	1				
MC-1	3(1)	1	1				
MC-2	3(2)			1		1	1
MC-2	4			1	1	1	1
MC-2	5			1		1	1

(5)

In the following, a new similarity measure is defined and incorporated in a proposed heuristic algorithm such as $\sum_{j=1}^{n} \delta(a_{ij}, a_{i'j})$, where $\delta(a_{ij}, a_{i'j}) = 1$, if $a_{ij} = 1$ and $a_{i'j} = 1$; 0, otherwise in a machine or part vectors as part vector k = $[a_{1k}, a_{2k}, \ldots, a_{mk}]^T$, and part vector l = $[a_{1l}, a_{2l}, \ldots, a_{ml}]^T$. The value of the similarity measure 1 indicates that the part vector l is a subset of part vector k or vice versa.

The comparison of the existing similarity measures with zero-one similarity measure is shown in Table 1, since the similarity coefficient is one of the methods used in GT applications. Compared to other methods, similarity measure incorporates more flexibility into the machine component grouping process and more easily lends itself to the computer application. Most of the GT algorithms employ a binary machine-part incidence matrix and do not use similarity measures. Many existing similarity measure (Table 1) are suitable for problems that do not involve bottleneck parts or machines.

In order to develop clustering heuristics, a new similarity measure is proposed. The proposed similarity measure is especially used for measuring the similarity between the bottleneck machines and the unstructured machine cells which are not belonging any bottleneck machines.

The purpose of developing a new similarity measure is to allocate or assign the bottleneck machines or bottleneck parts to the appropriate machine cells or part families based upon the grouping efficiency (GE), while the existing similarity measures generally apply the machine-part incidence matrix.

To measure the quality of clusters, the following grouping efficiency measure has been introduced as GE = $qn1 + (1-q)n_2$, where n_1 = number of entries "1" in the diagonal blocks / total number of elements in the diagonal blocks, n2 = 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.

Table 1. Similarity measures.

Coefficient	Reference	Formula	Form
Minkowski	[15]	$\sum_{k=1}^{n} \lvert a_{ik} \; a_{jk} \rvert \, p]^{1/p}$	Integer
McAuley	[16]	$\sum_{k=1}^{n} \delta'(a_{ik},a_{jk}) / \sum_{k=1}^{n} \delta''(a_{ik},a_{jk})$	$0 \le s \le 1$
Jaccard	[17]	$n_{ij} / (n_i + n_j \; n_{ij})$	$0 \le s \le 1$
Dutta et al.	[18]	$n_i + n_j - 2n_{ij}$	Integer
Dice Sorensen	[19]	$2n_{ij} / (n_i + n_j)$	$0 \le s \le 1$
Dot product	[20]	$n_{ij} / (n_i + n_j)$	$0 \le s \le 1$
Chu and Lee	[21]	$n_{ij}^2 / (n_i + n_j \; n_{ij})$	Integer
Kusiak and Cho	[22]	$\begin{cases} 1, & \text{if } a_{ik} \ge a_{il} \text{ or } a_{ik} \le a_{il}, \text{ for all } i \\ 0, & \text{otherwise} \end{cases}$	0 or 1
Cho et al.	This paper	$\sum_{j=1}^{n} \delta(a_{ik},a_{jk})$	Integer

$\delta'(a_{ik}, a_{jk})$ = 1, if $a_{ik} = a_{jk}$ = 1; 0, otherwise.

$\delta''(a_{ik}, a_{jk})$ =0, if $a_{ik} = a_{jk}$ = 0; 1, otherwise.

n_i = number of machines visited by part i.

n_{ij} = number of machines visited by part i and j.

d $(a_{ij}, a_{i'j})$ = 1, if a_{ij} = 1 and $a_{i'j}$ = 1; 0, otherwise.

II. Graph theoretic algorithm

The diagonal blocks of the machine-part incidence matrix are formed by a similarity of distance function between two rows of the machine-part incidence matrix. A minimum spanning tree (MST) with respect to this distance function will be found. The machines are to be grouped by deleting the longest arcs in the MST [23]. A weight function is then defined for the columns and is used to assign a column to a machine group according to the highest weight.

Let M = {1, 2, ..., m} be the index set of rows, and N = {1, 2, ..., n} be the index set of columns in a machine-part incidence matrix. Let S and T be two subsets of M, the symmetric difference of S and T is defined to be $S \Delta T = (S-T) \cup (T-S) = (S \cup T)-(S \cap T)$. The number of elements in S is denoted by | S |, and let M_j = {j∈N: a_{ij} = 1}, i = 1, 2, .., m ; N_j = {i∈M: a_{ij} = 1}, j = 1, 2, .., n. M_j is the set of parts to be processed by machine i and N_j is the set of machines to be visited by part j. For any two rows, i,h∈M, the distance between i and h is defined to be d(i,h) = $|M_j \Delta M_h|$ / $|M_j$ $_\cup M_h|$. Notice that this distance function satisfies that 0≤ d(i,h)≤1 for any i,h∈M.

It also satisfies the triangular inequality, i.e., d(i,k)≤ d(i,h)+d(h,k) for any i,h,k∈M. Recall that the Jaccard similarity coefficient between two rows i,h∈M is given by s(i,h) = $|M_i \cap M_h|$ / $|M_i \cup M_h|$. Hence the relation between Jaccard similarity coefficient and proposed distance function is given by the following equation d(i,h) + s(i,h) =1.

Jaccard similarity coefficient is a measure of the relative similarity of the two different machines in terms of the parts they have to process. The distance function is a measure of the relative dissimilarity of the two different machines. Some simple consequence of this definition is as if $M_i = M_h$, i,h ∈ M, then d(i,h) = 0, and if $M_i \cap M_h = \emptyset$, then d(i,h) = 1.

Consider each i∈M as a node in a graph and E = {(i,j): i,j∈M, i≠j} as the sets of arcs in the graph G = (M,E). (i,j) is considered to be the same as (j,i). The weight of arc (i,j) is given by d(i,j). A tree of G is a subgraph F = (V,H) of G with no cycles where V⊂M, H ⊂E. A spanning tree of G is a tree F = (V,H) where V = M.

A minimum spanning tree of G is a spanning tree with minimum total weight [3]. The MST algorithm consists of two phases such as to find minimum spanning tree of G = (M,E) and to decompose M into the required number of groups by deleting some arcs in the minimum spanning tree, to assign parts to different machine cells according to a weight function [24].

Step 0 : (Initialization)

Let V = M. Choose an arbitrary i∈V and set U = {i}. Give i the label [0,0]. Let $_j$ = d(j,i) for all j∈V-U. Label j∈V-U with [$_j$, i]. Let T =∅.

Step 1 : (Labeling procedures)

Choose j_0∈V-U such that $_{j0}$ = min {$_j$: j∈V-U}, and let the label of j_0 be [$_{j0}$, i_0]. In the case of tie, choose the j_0 with minimum $|M_{i0} \cup M_{j0}|$. Replace U by U∪{ j_0 } and T by T∪{(i_0, j_0)}.

Step 2 : (Updating $_j$)

If V-U =∅, go to step 3, otherwise for each jV-U. If $_j$ > d(j, j_0), then set $_j$ = d(j, j_0) and label j with [$_j$, j_0]. Go to step 1.

Step 3 : (Decomposition of the MST)

Arrange the m-1 arcs in T according to descending order of their weights. In case of tie, say d(i,h) = d(j,k), order (i,h) in front of (j,k) if $|M_i \cup M_h| \le |M_j \cup M_k|$. If the required number of groups is K, then delete the first K-1 arcs with the longest weight from the minimum spanning tree T. The resulting graph has K disjoint subtrees, (R_i, T_i), i=1,2,..., K, which has partitioned the m machines into K machine groups. Each R_i will form a machine cell.

Step 4 : (Assign parts to machine cells)

Set C_i =∅, i=1,2,..., K. For each j∈N, find w(j, R_i), i=1,2,..., K, where the weight function for each column w(j, R_i) is defined as $(\sum_{j \in i} a_{hj}) / \sum_{h=1}^{m} a_{hj}$, j∈N, i=1,2,...,K. Namely, w(j, R_i) is the ratio of the number of machines in R_i to be visited by j to the total number of machines to be visited by part j Let K be such that w(j, R_k) = Max {w(j, R_i): i=1,2,...,K}. Assign part j to part family C_k, i.e., update C_k to C_k {j}. In case of tie, break it arbitrarily.

The time required to determine for the distance matrix (d(i,j)) is $0(m(\sum_{i=1}^{m} mi))$, and for the minimum spanning tree is $0(m^2)$, where $m_i=|M_i|$, i=1,2,.., m and $n_i=|N_j|$, j=1,2,..,n. The time to decompose the spanning tree to K machine cells is 0(m log m). An alternate criterion is to choose an arc whose deletion will give the greatest increase in the grouping efficiency. The decomposition is then terminated when the desirable number of groups is obtained, or when there is no further increase in grouping efficiency by deleting arc.

To assign the parts j to machine cells requires $0(\sum_{j=1}^{m} nj)$ time which equals to $0(\sum_{k=1}^{m} mi)$. Hence the overall complexity of the given algorithm is $0(m^2 + m \sum_{i=1}^{m} mi)) = 0(m \sum_{i=1}^{m} mi)$. Based upon the MST, a heuristic algorithm is proposed to handle with the bottleneck machines or called exceptional elements which may increase grouping efficiency.

Step 0 : (Initialization)

Begin with the generalized machine-part incidence matrix $[a_{ij}]^n$.

Step 1 : (Transformation)

Transform the matrix $[a_{ij}]^n$ into the machine similarity matrix $[s_{ij}]^n$.

Step 2 : (Application of MST)

Apply the similarity matrix $[s_{ij}]^n$ to the minimum spanning tree algorithm, which the spanning tree is decomposed into machine cells according to those arcs with the largest weights.

Step 3 : (Graph representation)

Represent the current similarity matrix $[s_{ij}]^{n\ 1}$ with a transition graph, which a part is represented by a node, whereas a machine is represented by an edge for useful in detecting the bottleneck machines or parts.

Step 4 : (Fathoming)

Cluster the nodes the number of C_k -1 by cutting from the smallest arc's weight in the graph.

Step 5 : (Backtracking)

Make the unstructured matrix $[a_{ij}]^{n\ 1}$ with only clustered nodes. If there are some nodes clustered by itself, then exclude them for making unstructured matrix $[a_{ij}]^{n\ 1}$.

Step 6 : (Measuring similarity)

Calculate the similarity measures between unstructured matrix $[a_{ij}]^{n\ 1}$ and any machines clustered by itself.

Step 7 : (Stopping rule)

Apply the similarity matrix $[s_{ij}]^{n\ 2}$ between unstructured matrix $[a_{ij}]^{n\ 1}$ and exceptional elements to the Hungarian method in order to assign any machines clustered by itself which is called exceptional elements to the unstructured matrix $[a_{ij}]^{n\ 1}$ until there is no more machines left, otherwise stop.

One important aspect of the group technology problem is to devise a relevant measure of the final solution. For a given solution by proposed algorithm for efficient handling bottleneck machines, different arrangement of the machines and the parts within a group will give different objective values if some existing algorithms are applied, however, the objective value for the final solution will be unique if some algorithms seeking for the minimum intercell flow or maximum grouping efficiency.

In the proposed algorithm, the spanning tree is decomposed into machine groups according to those arcs with the largest weights. However, sorting the arcs of the MST according to descending order of their weights takes 0(m log m) time. An alternative criterion is to choose an arc whose deletion will give the greatest increase in the group efficiency.

The decomposition is terminated when the desirable number of groups is obtained, or when there is no further increase in grouping efficiency by deleting any arc. This alternative approach takes $0(m \sum_{i=1}^{m} mi)$ time. Hence the overall complexity is still $0(m \sum_{i=1}^{m} mi)$.

The grouping efficiency in general gives a relatively good measure for the solution to the GT problem. However, for some value of q a solution with a higher grouping efficiency need not be a more desirable solution [25]. Therefore, the selection of the value q for which the best solution corresponding to the highest grouping efficiency is not easy undertaking. The way for handling the bottleneck machines should be first considered in determining the group efficiency in real manufacturing systems.

III. Illustrative examples

Consider the following machine-part incidence matrix (6).

$$
\begin{array}{c|ccccccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\
1 & & 1 & 1 & & & & 1 & 1 & 1 & & & 1 & 1 \\
2 & & 1 & & & & & 1 & 1 & & & 1 & 1 & \\
3 & 1 & & & & & 1 & & & & & & & \\
4 & & & 1 & & 1 & & & & & 1 & & & \\
5 & & & 1 & & & & 1 & 1 & 1 & & & 1 & 1 \\
6 & & & & & 1 & & & & & & & & \\
7 & & & & 1 & & & & & & 1 & & & \\
8 & & & & 1 & & & & & & 1 & & & \\
9 & & & & & & 1 & & & & & & &
\end{array} \qquad (6)
$$

The difference between the MST algorithm and the proposed heuristic will be illustrated with this example. Table 2 shows the distance between different rows such as d(i,h) = d(j,k), order (i,h) in front of (j,k) if $|M_i \cup M_h| \le |M_j \cup M_k|$ and iterations of the MST algorithm are shown in Fig 1. The total weight of this minimum spanning tree is 4.435. The average arc weight in the tree is 0.554.

Table 2. Distance matrix (d(i,j)) based on MST.

	1	2	3	4	5	6	7	8	9
1	0.000	0.500	1.000	0.889	0.143	1.000	1.000	1.000	1.000
2	0.500	0.000	1.000	1.000	0.625	1.000	1.000	1.000	1.000
3	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000	0.500
4	0.889	1.000	1.000	0.000	0.875	0.667	0.667	0.750	1.000
5	0.143	0.625	1.000	0.875	0.000	1.000	1.000	1.000	1.000
6	1.000	1.000	1.000	0.667	1.000	0.000	0.000	1.000	1.000
7	1.000	1.000	1.000	0.750	1.000	1.000	1.000	0.000	1.000
8	1.000	1.000	1.000	0.750	1.000	1.000	1.000	0.000	1.000
9	1.000	1.000	0.500	1.000	1.000	1.000	1.000	1.000	0.000

Assume that three machine cells are proposed, then the two arcs in the MST with the largest weight arc (1), (3) and (5), (4) are deleted from the tree. Hence, there are three machine cells as R_1={1,2,5}, R_2={3,9}, and

R_3= {4,6,7,8}.

In order to assign the parts into the machine cells, the weight function for each column w(j, R_i), j=1,2,...,12 and i=1,2,3 are shown in Table 3. Thus according to the matrix (w(j, R_i)), the parts {2,3,7,8,9,11,12,13}=C_1 are assigned to R_1, parts {1,6}=C_2 are assigned to R_2, and parts {4,5,10}=C_3 are assigned to machine cell R_3 as shown in rearranged matrix (7).

However the only MST algorithm can give the basic GT solution as machine groups and part families respectively without considering the bottleneck situations if any exists.

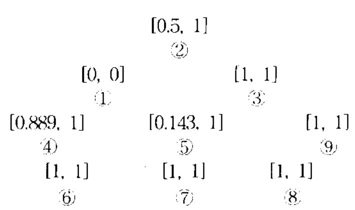
Table 3. Weighted matrix (w(j, R_i)).

	1	2	3	4	5	6	7	8	9	10	11	12	13
R1	0	1	2/3	0	0	0	1	1	1	0	1	1	1
R2	1	0	0	0	0	1	0	0	0	0	0	0	0
R3	0	0	1/3	1	1	0	0	0	0	1	0	0	0

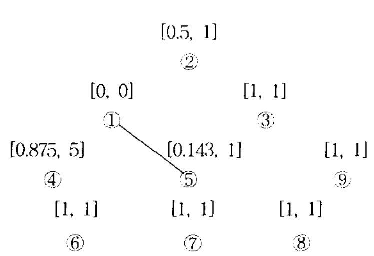
In this case, if the proposed new similarity measure as shown in matrix (7) is applied and solved by the proposed heuristic, then the deleted arcs will be machine 1 to 4 and machine 1 to 3. Therefore, three machine groups are formed as identified as matrix (8), which can be compared to the results of the proposed heuristic algorithm. Fig 2. Shows the procedure of clustering the nodes as the number of c-1 by cutting from the smallest arc's weight.

	1	2	3	4	5	6	7	8	9
1		5	0	1	5	0	0	0	0
2			0	0	3	0	0	0	0
3				0	0	0	0	0	1
4					1	1	1	1	0
5						0	0	0	0
6							0	0	0
7								2	0
8									0
9									

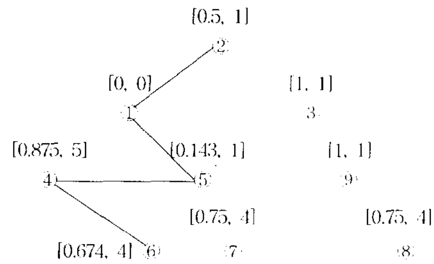
(7)



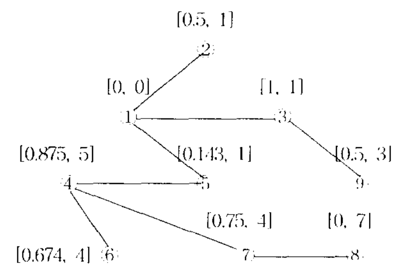
a) Beginning of MST
(U={1}, T=∅, minimum is π_5)



b) After one iteration
(Add node 5 to U. U={2,5} and update π_j for J V-U. T={(1,5)})



c) After 4 iterations
(U={(1,5,2,4,6)}, T={(1,5), (1,2), (5,4), (4,6)}.)



d) MST solution
T={(1,5), (5,4), (4,6), (4,7), (7,8), (2,3), (3,9)}

Fig. 1. Few iterations in MST.

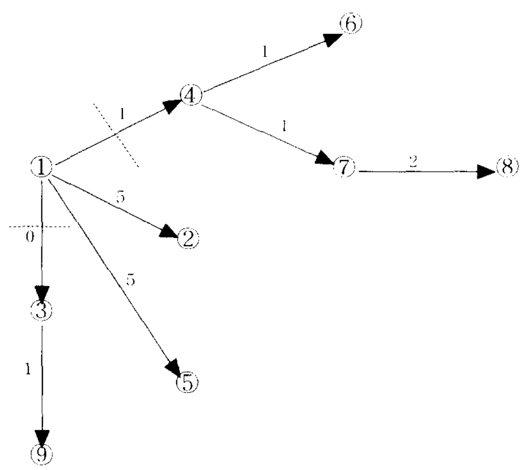


Fig. 2. Clustering procedures by cutting arc's weight matrix (7).

	2	3	7	8	9	11	12	13	1	6	4	5	10
2	1		1	1		1	1						
1	1	1	1	1	1		1						
5		1	1	1	1								
3									1	1			
9										1			
8											1		1
7											1		1
4		1										1	1
6												1	

(8)

Hence, consider the following machine-part incidence matrix (9) for illustrating the proposed heuristic in detail.

Step 0 : Begin with the generalized machine-part incidence matrix (9).

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

(9)

Step 1 : Transform the matrix $[a_{ij}]^n$ into the machine similarity matrix (10).

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

(10)

Step 2 : Apply the similarity matrix $[s_{ij}]^n$ to the minimum spanning tree algorithm. Let V = M. Choose an arbitrary $i \in V$ and set U = {i}, and replace new similarity measures as d(j,i) for all $j \in V-U$. Arrange the m-1 arcs in T according to descending order of their weights. If the required number of groups is K, then delete the first K-1 arcs with the smallest weight from the minimum spanning tree T. The resulting graph has K disjoint subtrees, which has partitioned the m machines into K machine groups.

Step 3 : Represent the current similarity matrix $[s_{ij}]^{n-1}$ with a transition graph.

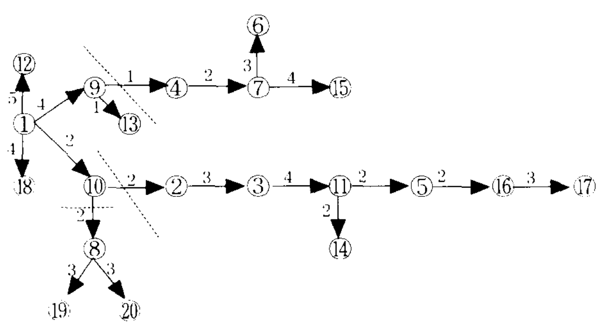


Fig. 3. Clustering procedures by cutting arc's weight matrix (10).

Step 4 : Cluster the nodes as the number of c-1 by cutting from the smallest arc's weight in the below graph as Figure 3 shows.

Step 5 : Make the unstructured matrix $[a_{ij}]^{n-1}$ with only clustered nodes. If there are some nodes clustered by itself, then exclude them for making unstructured $[a_{ij}]^{n-1}$. This step results unstructured machine groups as MC-1={1,9,10,12,18}, MC-2={4,6,7,15}, MC-3={2,3,5,11,17}, MC-4={8,19,20}, and the machine 13 and 14 are clustered by itself only, which are need to be excluded for making the unstructured matrix.

Step 6 : Calculate the similarity measures between unstructured matrix $[a_{ij}]^{n-1}$ and any machines clustered by itself , machine 13 and 14, as shown below Table 4.

Table 4. Similarity between bottleneck machines and unstructured machine cells.

	MC-1	MC-2	MC-3	MC-4
M13	0	1	1	0
M14	0	0	3	0

Step 7 : Apply the similarity mat6rix $[s_{ij}]^{n-2}$ to the Hungarian method [26], which assigns any machines clustered by itself to the unstructured matrix $[a_{ij}]^{n-1}$. This step results that the machine 14 is clustered in the MC-3, and machine 13 can be clustered in either MC-2 or MC-3 based upon the grouping efficiency.

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

(11)

The final matrix is shown in matrix (11). Four machine cells and four part families are grouped such as: MC-1={1,9,10,12,18}, MC-2={4,6,7,13,15}, MC-3={2,3,5,11,14, 16,17}, and MC-4={8,19,20}, and four part families such as PF-1={1,9,12,14,17,20}, PF-2={5,8,13,16}, PF-3={2,4,6,7,,1 1,15,19}, and PF-4={3,10,18} respectively.

Comparing the grouping efficiency regarding to hand with the bottleneck machines, 40x100 matrix from reference Chandrasekharan, M.P., and Rajagopalan, R., [14] is illustrated. Applying the proposed heuristic to 40x100 matrix will result 10 machine groups with 36 bottleneck machines, while reference Kusiak, A., and Cho, M., [22] indicates 37 bottleneck machines with same number of machine groups. The grouping efficiency is also improved based on the grouping threshold values.

IV. Conclusion

In this paper, a combinatorial approach for solving the group technology problem is discussed with a minimum spanning tree algorithm which does not require the initial seeds, of which objective value is independent of the intra group ordering of machines and parts, and Hungarian method which is easy to implement.

Another important feature of the proposed one is the ease of the control of the size of the machine groups. 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. 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.

A new similarity coefficient measure which can be applied to the machine cells and other manufacturing related areas is proposed. This measure is also applicable for the group technology problem with basic and alternative process plans. It works well for many cases where a block diagonal structure is embedded into the machine-part incidence matrix, and machine cell-part incidence matrix.

A non-hierarchical procedures for solving the GT problem has been discussed, which does not require initial seeds unlike the existing non-hierarchical algorithms do. The proposed algorithm performs well for the most of the GT problems.

This study also showed that the similarity coefficients have a significant effect on the quality of clusters. The grouping efficiency depending on the various threshold value are to be tested. It suggests that the grouping efficient may be varied upon the arcs' weight, which we proposed a efficient ways to do for flexibility in real industries. Further research will be applied the different similarity coefficient or distance function to investigate the difference results.

References

[1] G. M. Fazakerlay, "Group technology : social benefits and social problems," *The Production Engineer*, October, pp. 383-386, 1974.

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

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

[4] H. Seifoddini and P. M. Wolfe, "Application of the similarity coefficient method in group technology," *IIE Transactions*, vol. 18, no. 3, 1986.

[5] J. R. King, "Machine-component group formation in group technology," *The International Journal of Management Science*, vol. 8, no. 2, pp. 193-199, 1980.

[6] H. M. Chan and D. A. Milner, "Direct clustering algorithm for group formation in cellular manufacture," *Journal of Manufacturing Systems*, vol. 1, no.1, pp. 63-74, 1982.

[7] W. T. McCormick, P. J. Schweitzer and T. E. White, "Problem decomposition and data reorganization by a cluster technique," *Operations Research*, vol. 53, pp. 993-1009, 1972.

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

[9] M. V. Bhat and A. Haupt, "An efficient clustering algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6, no. 1, pp. 61-64, 1976.

[10] R. Askin and S. Subramanian, "A cost-based heuristic for group technology configuration," *International Journal of Production Research*, vol. 25, no. 1, pp. 101-114, 1987.

[11] A. Kusiak and W. S. Chow, "Efficient solving of the group technology problem," *Journal of Manufacturing Systems*, vol. 6, no. 2, pp. 117-124, 1987.

[12] A. Kusiak, *Intelligent Manufacturing Systems*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[13] A. Ballakur and H. J. Steudel, A "Within-cell utilization based heuristic for designing cellular manufacturing systems," *International Journal of Production Research*, vol. 25, no. 5, pp. 639-665, 1987.

[14] M. P. Chandrasekharan and R. Rajagopalan, "ZODIAC-An algorithm for concurrent formation of part-families and machine-cells," *International Journal of Production Research*, vol. 25, no. 6, pp. 835-850, 1987.

[15] T. S. Arthanari and Y. Dodge, *Mathematical Programming in Statistics*, New York: Wiley, 1981.

[16] J. McAuley, "Machining grouping for efficient production," *The Production Engineer*, pp. 53-57, February, 1972.

[17] L. Leskowsky, L. Logan and Vannelli, "Group technology decision aids in an expert system for plant layout," *Modern Production Management Systems*, edited by Kusiak, Amsterdam: North Holland, pp. 561-585, 1987.

[18] S. P. Dutta, R. S. Lashkari, G. Nadoli and T. Ravi, "A heuristic procedure for determining manufacturing families from design-based grouping for flexible manufacturing systems," *Computers and Industrial Engineering*, vol. 10, no. 1, pp. 193-201, 1986.

[19] C. H. Chu and P. Pan, "A comparison of hierarchical clustering techniques for manufacturing cellular formation," *Proceedings of the Third International Conference on CAD/CAM, Robotics, and Factory of the Future*, edited by B. Prasad, Springer Verlag, New York, 1988.

[20] C. H. Chu and M. Tsai, "Clustering analysis in manufacturing cellular formation," *OMEGA*, vol. 17, no. 3, pp. 289-295, 1989.

[21] C. H. Chu and W. Lee, "An efficient heuristic for grouping part families / machine cells," *Research Paper*, Iowa State University, Ames, Iowa, 1989.

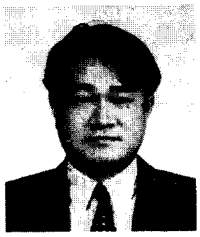
[22] A. Kusiak and M. Cho, "Similarity coefficient algorithms for solving the group technology problem," *International Journal of Production Research*, vol. 30, no. 11, pp. 2646-2666, 1992.

[23] E. Minieka, *Optimization Algorithms For Networks and Graphs*, Marcel Dekkar Inc., New York, 1978.

[24] M. N. Shu, "A new spanning tree algorithm for group technology problems," *Working Paper*, University of Southern California, 1987.

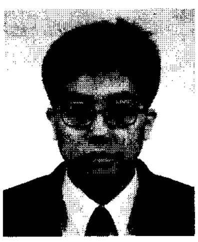
[25] C. S. Kumer and M. P. Chandrasekharan, "Grouping efficacy : a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology," *International Journal of Production Research*, vol. 28, no. 2, pp. 233-243, 1990.

[26] 정성진, 박진우, 김재윤, "FMS의 설계 및 운용을 위한 기계 부품 그룹 형성에 관한 연구" 경영과학, 4권, pp. 76-83, 1987.



조 문 수

1983년 숭실대학교 산업공학과 졸업. 1995년 미국 Western Illinois University, MBA 졸업. 1990년 미국 University of Iowa, 산업공학석사. 1995년 동 대학원 산업공학박사. 1992년 10월 ~ 1996년 1월 미국 PMX사 생산기획조정실장. 1996년 3월 ~ 현재 숭실대학교 산업공학과. 주관심 분야는 생산자동화, 동시공학, 공장자동화.



윤 창 원

1991년 숭실대학교 산업공학과 학사. 1993년 동 대학원 산업공학석사. 1996년 동 대학원 산업공학박사. 1997년 3월 ~ 현재 쌍용정보통신 Consultant. 관심 분야는 정보공학, 사무자동화, 생산자동화.

정 병 희

1977년 서울대학교 산업공학과 학사. 1979년 동 대학원 산업공학석사, 1986년 동 대학원 산업공학박사. 1980년 3월 ~ 현재 숭실대학교 산업공학과 교수. 관심분야는 생산정보시스템, FMS계획 및 운영, Logistics Management.