IE Interfaces
Vol. 12, No. 4, pp. 532-539, December 1999.

# Minimizing the Number of Inter-Cell Movement of Parts with Consideration of a Machine-Cell Size

Chang-Kyu Park
University of Ulsan, College of Business Administration

# 제한된 기계군의 크기하에서 부품의 이동을 최소로 하는 GT 기법

박 창 규

The first step to design a cellular manufacturing system is to make part-families and machine-cells. This process is called the machine-part grouping. This paper considers a machine-cell size when grouping machine-cells. By considering a machine-cell size, an unrealistically big size of machine-cell which may be caused by the chaining effect can be avoid. A heuristic algorithm which minimizes the number of inter-cell movement of parts considering a machine-cell size is presented. The effectiveness and performance of the proposed heuristic algorithm are compared with those of several heuristic algorithms previously reported. The comparison shows that the proposed heuristic algorithm is efficient and reliable in minimizing the number of inter-cell movement of parts and also prevents the chaining effect.

## 1. Introduction

Group technology (GT) is a philosophy that implies the notion of recognizing and exploiting similarities in three different ways:

(1) By performing like activities together
(2) By standardizing similar tasks
(3) By efficiently storing and retrieving information about recurring problems

The primary advantage of GT implementation is that a large manufacturing system to produce a set of parts can be decomposed into smaller subsystems of part-families based on similarities in design attributes and manufacturing features. The decomposition based on similarities of design attributes, manufacturing features, and functions leads to improved productivity in various functional areas of an organization(Singh, 1996). GT has also resulted in many benefits such as reduced set-up times, reduced material handling and work-in-process, and improved output quality(Boe and Cheng, 1991; Chen and Guerrero, 1994).

Cellular manufacturing is an application of GT to the manufacturing. In other words, the manufacturing system can be re-organized into many cells through GT. Each cell has a group of dissimilar machines so that a family of parts can be processed as much as possible in the given cell. The first step to design cells in the cellular manufacturing system is to identify part-families and to form machine-cells where the parts can be processed(Boe and Cheng, 1991). This grouping process is called the machine-part grouping.

This paper considers a machine-cell size when grouping machine-cells. By considering a machine-cell size, an unrealistically big size of machine-cell which may be caused by the chaining effect can be avoid. A heuristic algorithm which minimizes the number of inter-cell movement of parts considering a machine-cell size is presented.

Many methods to solve the machine-part grouping problem have been reported. Kusiak and Chow (1988) classified the existing algorithms for GT into:

(1) Matrix formulation
(2) Mathematical programming formulation
(3) Graph formulation

Srinivasan(1994) divided the matrix formulation into array-based method and clustering method.

Thus, the existing algorithms for GT were classified into:

(1) Array-based method
(2) Clustering method
(3) Mathematical programming-based method
(4) Graph theoretic method

The approach of array-based method is to re-arrange the rows and/or columns of the machine-part incidence matrix to produce a matrix with a block diagonal form(Boe and Cheng, 1991; Burbidge, 1971; King, 1980; Chan and Milner, 1982; Mukhopadhyay *et al.*, 1995). Srinivasan(1994) pointed out that the array-based method has the disadvantage of being dependent on the initial configuration of the machine-part incidence matrix and not being able to provide disjoint part-families and machine-cells for ill-structured machine-part incidence matrices.

Clustering method has been developed based on hierarchical and/or non-hierarchical clustering algorithm(Chen and Guerrero, 1994; McAuley, 1972; McCormick *et al.*, 1972; Carrie, 1973; Mosier and Taube, 1985; Seifoddini and Wolfe, 1986; Chandrasekharan and Rajagopalan, 1986; Seifoddini, 1989; Srinivasan and Narendran, 1991). The clustering method has not been successfully implemented in grouping machine-parts. Boe and Cheng(1991) explained that the clustering method permits a solution matrix with a chequer-board structure which has overlapping blocks. Thus, the visual identification of machine-cells and part-families from such a solution matrix is very difficult and may be impossible for large problems.

Most mathematical programming-based methods utilize distance measures (or dissimilarity measures), similarity measures, product costs, and so on(Kusiak, 1987; Steudel and Ballaur, 1987; Gunasingh and Lashkari, 1989; Shtub, 1989; Srinivasan *et al.*, 1990). These methods formulate the machine-part grouping into the optimization model.

Graph theoretic method treats the machines and parts as nodes and the processing of parts as arcs connecting these nodes. This method aims at obtaining disconnected sub-graphs from the machine-part graph to identify part-families and machine-cells(Srinivasan and Narendran, 1991; Srinivasan, 1994; Rajagopalan and Batra, 1975; Lee et al., 1982; Vannelli and Kumar, 1986; Vohra *et al.*, 1990; Askin *et al.*, 1991). Three types of graphs can be used:

(1) Bipartite
(2) Transition graph
(3) Boundary

Another classification was given by Chow and Hawaleshka(1993). They classified the above approaches into:

(1) Machine-part grouping approach
(2) Part grouping approach
(3) Machine grouping approach

The machine-part grouping approach includes the simultaneous identification of machine-cells and part-families; the part grouping approach is based on the initial formation of part-families and followed by the assignment of machines to those part-families; and the machine grouping approach attempts to first identify machine-cells and then assign parts to those machine-cells.

This paper is organized as follows: The machine-part grouping problem is described in the next section and followed by a solution methodology. Solution methodology presents a heuristic algorithm which minimizes the number of inter-cell movement of parts considering a machine-cell size. The comparison of the proposed heuristic algorithm with several heuristic algorithms previously reported is also presented.

## 2. Machine-Part Grouping Problem

The machine-part grouping problem is to find machine-cells and part-families such that one or more part-families can be fully processed within a single machine-cell(Seifoddini and Wolfe, 1986). A machine-part incidence matrix is used as the input of the machine-part grouping problem. A machine-part incidence matrix $\{a_{ij}\}$ is an ($M$ x $N$) matrix with zero or one entry:

$$a_{ij}=\begin{cases}1 & \text{if machine } i \text{ processes part } j\\ 0 & \text{otherwise}\end{cases}$$

where $M$ is the total number of machines and $N$ is the total number of parts.

An ideal case of machine-part grouping is to find a grouping of machines such that no part requires more than a single machine-cell for processing (i.e., there is no inter-cell movement of parts). In practice this is very rare(Chen and Guerrero, 1994). The parts processed in more than one machine-cell are called exceptional parts and the machines processing them are referred to as bottleneck machines (Seifoddini and Wolfe, 1986). Exceptional parts and bottleneck machines are the sources of inter-cell movement of parts. In order to obtain the maximum benefits of a cellular manufacturing system, the number of inter-cell movement of parts must be reduced to the minimum(Boe and Cheng, 1991). To deal with them, Burbidge(1977) suggested that exceptional parts can be eliminated by:

(1) Changing the machining methods to use machines already in the machine-cell
(2) Re-designing the parts to eliminate the need for outside operation
(3) Buying the parts instead of making them

The problem created by these exceptional parts or bottleneck machines leads to no clear grouping for machines or parts. This results in variation from algorithm to algorithm and the necessity for subjective judgement or human intervention. Thus, the identification of these conditions becomes crucial to the development of parts-families and machine-cells (Chen and Guerrero, 1994).

With accepting the inevitability of exceptional parts or bottleneck machines, this paper solves the machine-part grouping problem with the following objectives:

(1) Minimize the number of inter-cell movement of parts
(2) Satisfy the limitation of machine-cell size.

It is assumed that the desired number of machine-cells and machine-cell size are known. The next section presents a solution methodology.

## 3. Solution Methodology

In this section, the machine-part grouping problem is solved by considering the limitation of machine-cell size. The solution methodology consists of two steps: first group machine-cells and then assign parts to those machine-cells. A heuristic algorithm that minimizes the number of inter-cell movement of parts considering a machine-cell size is presented.

### 3.1 Heuristic Algorithm

A heuristic algorithm that minimizes the number of inter-cell movement of parts without consideration of a machine-cell capacity was proposed by Chow and Hawaleshka(1993). They used the commonality score defined by Wei and Kern(1989). The commonality score is the weighted sum of the number of common one and zero entries in the pair of rows in a machine-part incidence matrix.

The heuristic algorithm presented in this paper is different from Chow and Hawaleska's algorithm in that:

(1) It considers a machine-cell size, but Chow and Hawaleshka's algorithm does not. If let a machine-cell size be infinite, then two algorithms generates the same result. Thus, this paper's algorithm is more general form than Chow and Hawaleshka's algorithm.
(2) It prevents the chaining effect, but Chow and Hawaleshka's algorithm does not. Thus, this paper's algorithm does not allow an unrealistically big size of machine-cell.
(3) It uses the similarity and cohesiveness scores as measures to solve the machine-part grouping problem, while Chow and Hawaleshka's algorithm uses the commonality score. Because the chaining effect cannot be prevented using only the commonality score, this paper's algorithm devises the similarity and cohesiveness scores. The similarity score is the number of common one in a pair of machine-cells and the cohesiveness score represents a degree of similarity of a machine to the considered machine group.

First of all, to facilitate the understanding of this paper's algorithm, the following concepts are presented. It is important to note that each machine is regarded as an individual machine-cell at the beginning and each iteration reduces one machine-cell by combining a pair of machine-cells.

(1) The machine-cells, $MC_i$, are defined as

$$MC_i = [mc_{i1}, mc_{i2}, \cdots, mc_{iN}],\ i = 1, 2, \cdots, M$$

where

$$mc_{ij} = \begin{cases} 1 & \text{if part } j \text{ visits machine-cell } i \\ 0 & \text{otherwise} \end{cases}$$

(2) The similarity of a pair of machine-cells (e.g., machine-cells k and p), $S_{kp}$, is defined by

$$S_{kp} = \sum_{j=1}^{N} mc_{kj}\ mc_{pj}$$

In other words, the similarity score is the number of common one in a pair of machine-cells.

(3) The difference of a pair of machine-cells (e.g., machine-cells k and p), $D_{kp}$, is defined by

$$D_{kp} = \sum_{j=1}^{N} mc_{kj} + \sum_{j=1}^{N} mc_{pj} - 2^{*} S_{kp}$$

(4) The cohesiveness of machine $i$ to the considered machine group, $C_i$, is defined as

$$C_i = \sum_{j \neq 1} S_{ij} \quad \forall j \in \text{machine group}$$

In other words, the cohesiveness score represents a degree of similarity of a machine to the considered machine group.

(5) Chaining is checked when combining a pair of machine-cells, one of which has the full machine-cell size limit and other of which has only one machine (it is called as a candidate machine). <Figure 1> shows a simple example where the limitation of machine-cell size is 3. Machine-cells, $MC_1$, $MC_2$, and $MC_3$ are already formed as a machine-cell (dash-lined box) and $MC_4$ is a candidate machine to join them. Numbers on arcs represent similarity scores and numbers on nodes represent cohesiveness scores.

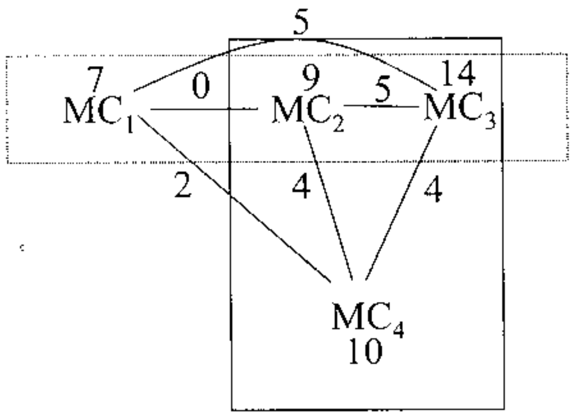


Figure 1. Chaining check.

If the cohesiveness score of candidate machine is greater than any cohesiveness score of a machine in the machine-cell, the candidate machine joins into the machine-cell and extracts the machine that has the lowest cohesiveness score. In this example, $MC_2$, $MC_3$, and $MC_4$ produce a new machine-cell(bold-lined box) and $MC_1$ becomes a new single machine-cell.

(6) Combining operation of a pair of machine-cells (e.g., machine-cells $k$ and $p$, and new machine-cell $n$) is performed by

$$\begin{aligned} MC_n &= MC_k \text{ OR } MC_p \\ &= [mc_{n1}, mc_{n2}, \cdots, mc_{nN}] \end{aligned}$$

where

$$mc_{nj} = \begin{cases} 1 & \text{if } mc_{kj} = 1 \text{ in } MC_k \text{ or } mc_{pj} = 1 \text{ in } MC_p \\ 0 & \text{otherwise} \end{cases}$$

(7) The number of inter-cell movement of parts, $NMOVE$, is calculated by (e.g., given $r$ machine-cells)

$$NMOVE = \sum_{j=1}^{N} \left( \sum_{k=1}^{r} mc_{kj} - 1 \right)$$

This algorithm is simple and easy to implement, but quite powerful in obtaining the objective of minimizing the number of inter-cell movement of parts, even for large problems. At the beginning, each machine is regarded as an individual machine-cell. Then, each iteration reduces one machine-cell by combining a pair of machine-cells according to the rule. The algorithm iterates until it produces the desired number of machine-cells. Once the best machine-cells are identified, part-families are grouped. Part-families are formed to include parts which visit a specific machine-cell. If a part visits several machine-cells (i.e., exceptional part), the part is included in the part-family corresponding to the machine-cell where the part uses the most number of machines.

*Algorithm*

Step 1: Compute the similarity for each pair of machine-cells.

Step 2: Select the pair of machine-cells that has the highest similarity score. In the case of ties, compute the difference and select the pair of machine-cells that has the lowest difference score (In the case of ties again, select a pair of machine-cells randomly).

Step 3: IF the total number of machines in the pair of machine-cells does not exceed the machine-cell size limit ($L_k$)
THEN go to step 4
ELSE check the number of machines in each machine-cell.

IF the numbers of machines in both machine-cells are greater than 1
THEN go to step 2 without considering the current pair of machine-cells
ELSE check chaining.

IF the cohesiveness score of candidate machine is greater than any cohesiveness score of a machine in the machine-cell
THEN combine the candidate machine into the machine-cell, extract the machine that has the lowest cohesiveness score, and go to step 1
ELSE go to step 2 without considering the current pair of machine-cells.

Step 4: Combine the pair of machine-cells selected and calculate the number of inter-cell movement of parts.
Step 5: Repeat steps 1, 2, 3, and 4 until the desired number of machine-cells is reached.
Step 6: Group part-families (Part-families are formed to include parts which visit a specific machine-cell. If a part visits several machine-cells, the part is included in part-family corresponding to machine-cell where the part uses the most number of machines.).

## 3.2 Example

To illustrate this algorithm, a (11x22) machine-part incidence matrix from Seifoddini's paper(1989) is considered in <Figure 2>. Suppose that the desired number of machine-cells is 3 and that each machine-cell size limitation is 4. The results of each iteration are shown in <Table 1>.

| Machines \ Parts | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 1 | 1 | | | | | | | | 1 | | | | 1 | 1 | | | | 1 | 1 | 1 |
| B | | | | | 1 | | | 1 | | | 1 | 1 | | | | | | | 1 | | | |
| C | | | | | 1 | 1 | | | | | | 1 | 1 | | | | | | 1 | | | |
| D | 1 | 1 | 1 | | | | | | | | 1 | | | | 1 | 1 | | | | 1 | 1 | 1 |
| E | 1 | 1 | 1 | | | | 1 | | | | | | | | 1 | 1 | | | | 1 | 1 | 1 |
| F | | | | | 1 | | | 1 | | | | 1 | | | | | | | 1 | | | |
| G | | | | 1 | | 1 | | | 1 | | | | | 1 | | | 1 | 1 | | | | |
| H | | | | | 1 | 1 | | | 1 | 1 | | 1 | 1 | 1 | | | 1 | 1 | 1 | | | |
| I | | | | 1 | | | | | 1 | | | | | 1 | | | 1 | 1 | | | | |
| K | 1 | 1 | 1 | | 1 | | 1 | 1 | | | 1 | 1 | | | 1 | 1 | | 1 | | | | |
| L | | | | 1 | | | | | 1 | 1 | | | | | | | 1 | 1 | | | | |

Figure 2. Machine-part incidence matrix

At the beginning (iteration 0), each machine is regarded as an individual machine-cell. Each iteration reduces one machine-cell by combining a pair of machine-cells. For example, the similarity scores in iteration 0 are shown in <Figure 3>. From <Figure 3>, machine-cells A and D are combined because the pair of machine-cells A and D has the highest similarity score of 9. The iteration of algorithm continues until the algorithm reduces the number of machine-cells to the desired number of machine-cells of 3.

Table 1. The result of each iteration

| Itration | Machine Group | NMOVE |
|---|---|---|
| 0 | (A)(B)(C)(D)(E)(F)(G)(H)(I)(K)(L) | 56 |
| 1 | (AD)(B)(C)(E)(F)(G)(H)(I)(K)(L) | 47 |
| 2 | (ADE)(B)(C)(F)(G)(H)(I)(K)(L) | 39 |
| 3 | (ADEK)(B)(C)(F)(G)(H)(I)(L) | 32 |
| 4 | (ADEK)(B)(C)(F)(GI)(H)(L) | 27 |
| 5 | (ADEK)(B)(CH)(F)(GI)(L) | 22 |
| 6 | (ADEK)(B)(CGHI)(F)(L) | 17 |
| 7 | (ADEK)(B)(C)(F)(GHIL) | 17 |
| 8 | (ADEK)(BF)(C)(GHIL) | 13 |
| 9 | (ADEK)(BCF)(GHIL) | 10 |

*NMOVE* = The number of inter-cell movement of parts

| | A | B | C | D | E | F | G | H | I | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | | | | |
| B | 1 | | | | | | | | | | |
| C | 0 | 3 | | | | | | | | | |
| D | 9 | 1 | 0 | | | | | | | | |
| E | 8 | 0 | 0 | 8 | | | | | | | |
| F | 0 | 4 | 3 | 0 | 0 | | | | | | |
| G | 0 | 0 | 1 | 0 | 0 | 0 | | | | | |
| H | 0 | 3 | 5 | 0 | 0 | 3 | 5 | | | | |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 4 | | | |
| K | 6 | 4 | 2 | 6 | 6 | 3 | 1 | 3 | 1 | | |
| L | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 1 | |

Figure 3. Similarity scores.

In iteration 6, machine-cell L becomes a candidate machine to join into the machine-cell, CGHI. Chaining is checked in this iteration. The cohesiveness scores of machines, C, G, H, and I, are calculated in 6, 15, 18, and 13, respectively. Then, the cohesiveness score of machine L is calculated in 12. Since the cohesiveness score of machine L is greater than that of machine C, machine L joins into the machine-cell and extracts machine C from the machine-cell.

Finally, iteration 9 results in machine-cell groups of (ADEK), (BCF), and (GHIL). The number of inter-cell movement of parts is 10. Part-families are formed as:

Machine-cell ADEK— Parts 1,2,3,7,15,16,20,21,22
Machine-cell BCF — Parts 5,8,12,13,19

Machine-cell GHIL — Parts 4,6,9,10,14,17,18

## 4. Comparison and Discussion

In order to show the effectiveness and performance of the proposed algorithm, several examples were solved and the results were compared with those previously reported. The selected examples from the published papers are given as follows:

(1) 16 x 43 (Average Linkage Clustering Algorithm(Seifoddini and Wolfe, 1986))
(2) 8 x 20 (Ideal Seed Non-hierarchical Clustering Algorithm(Chandrasekharan and Rajagopalan, 1986))
(3) 24 x 40 (GROUPABILITY(Chandrasekharan and Rajagopalan, 1989))
(4) 11 x 22 (Average Linkage Clustering Algorithm(Seifoddini, 1989))
(5) 16 x 30 (Assignment Model(Srinivasan *et al.*, 1990))
(6) 8 x 20, 20 x 35 (Close Neighbour Algorithm(Boe and Cheng, 1991))
(7) 8 x 20, 24 x 40 (GRAFICS(Srinivasan and Narendran, 1991))
(8) 5 x 15 (General Search Algorithm(Chen and Guerrero, 1994))
(9) 24 x 40 (Minimum Spanning Tree(Srinivasan,

Table 2. Comparison with other heuristic algorithms

| Example | Result (Machine Group) | NMOVE | L = Infinite | | | L = The specific Number | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | R | Result (Machine Group) | NMOVE | R | L | Result (Machine Group) | NMOVE |
| (1) | (1,2,9,16)(3,14)(4,5,6,8,15)(7,10)(11,12,13) | 24 | 5 | (1)(2,4,5,6,7,8,9,10,11,12,5,16)(3)(13)(14) | 10 | 5 | 5 | (1,2,9,16)(3,14)(4,5,6,8,15)(7,10)(11,12,13) | 24 |
| (2) | (1,3)(2,4,7,8)(5,6) | 9 | 3 | (1,3)(2,4,7,8)(5,6) | 9 | 3 | 5 | (1,3)(2,4,7,8)(5,6) | 9 |
| (3) | (1,13,21,22)(2,5,11,19)(3,20)(4,16)(6,8,12,15,18)(7,14,23,24)(9,10,17) | 0 | 7 | (1,13,21,22)(2,5,11,19)(3,20)(4,16)(6,8,12,15,18)(7,14,23,24)(9,10,17) | 0 | 7 | 5 | (1,13,21,22)(2,5,11,19)(3,20)(4,16)(6,8,12,15,18)(7,14,23,24)(9,10,17) | 0 |
| (4) | (ADEK)(BCFH)(GIL) | 10 | 3 | (ADEK)(BF)(CGHIL) | 8 | 3 | 4 | (ADEK)(BCFH)(GIL) | 10 |
| (5) | (1,4,7,8,11,12)(2,13)(3,6,9,15)(5,10,14,16) | 18 | 4 | (1,3,4,6,7,8,9,11,12,15)(2)(5,10,14,16)(13) | 14 | 4 | 6 | (1,4,7,8,11,12)(2)(3,6,9,13,15)(5,10,14,16) | 17 |
| (6) | (1,3)(2,4,7,8)(5,6) | 9 | 3 | (1,3)(2,4,7,8)(5,6) | 9 | 3 | 4 | (1,3)(2,4,7,8)(5,6) | 9 |
| | (1,7,11,12,15,16,19)(2,4,13,14,18)(3,8,17)(5,6,9,10,20) | 30 | 4 | (1,2,3,4,5,6,7,8,9,10,11,14,15,16,17,18,19)(12)(13)(20) | 14 | 4 | 7 | (1,3,7,8,11,16,17)(2,4,13,14,18)(5,6,9,10,20)(12,15,19) | 26 |
| (7) | (1,3)(2,4,7,8)(5,6) | 9 | 3 | (1,3)(2,4,7,8)(5,6) | 9 | 3 | 4 | (1,3)(2,4,7,8)(5,6) | 9 |
| | (1,10.17)(2,19,23)(3,20)(4,14)(5,9,15,16)(6,8,22)(7,24)(11,13,21)(12,18) | 59 | 9 | (1,2,5,6,8,9,10,12,13,15,16,18,19,20,21,23)(3)(4)(7)(11)(14)(17)(22)(24) | 33 | 9 | 4 | (1,10,17)(2,5,9,19)(3,11,12,18)(4,7,14,24)(6,8,15,16)(13,23)(20)(21)(22) | 52 |
| (8) | (ABCDE)(F) | 5 | 2 | (ABCDE)(F) | 5 | 2 | 5 | (ABCDE)(F) | 5 |
| (9) | (1,17)(2,9,23)(3,11,21)(4,14,20)(5,9,13)(6,8,22)(7,24)(10,12,18)(15,16) | 58 | 9 | (1,2,5,6,8,9,10,12,13,15,16,18,19,20,21,23)(3)(4)(7)(11)(14)(17)(22)(24) | 33 | 9 | 4 | (1,10,17)(2,5,9,19)(3,11,12,18)(4,7,14,24)(6,8,15,16)(13,23)(20)(21)(22) | 52 |
| (10) | (1,7,10)(2,3,4,8)(5,6,9) | 0 | 3 | (1,7,10)(2,3,4.8)(5,6,9) | 0 | 3 | 4 | (1,7,10)(2,3,4,8)(5,6,9) | 0 |

*NMOVE* = The number of inter-cell movement of parts

*R* = The number of machine-cells

*L* = The limitation of machine-cell size

1994))
(10) 10 x 10 (Methods of Moments(Mukhopadhyay *et al.*, 1995))

<Table 2> shows the comparison of results produced by heuristic algorithms previously reported and by the proposed algorithm. The number of machine-cells, $R$, is set to the same number of each example. For the sake of simplicity, the limitation of each machine-cell size is set to $L$. The proposed algorithm solved the examples in two cases:
(1) there is no limitations of machine-cell size (i.e., $L = \infty$), in which Chow and Hawaleshka's algorithm generates the same result and
(2) machine-cell size is set to the same number of each example (i.e., $L$ = the specific number).

The machine-part incidence matrix may result in the following two categories:
(1) Mutually separable matrix
(2) Partially separable matrix

In the case that the machine-part incidence matrix can be separated mutually, all existing heuristic algorithms produce perfect machine-cells and part-families grouping (i.e., there is no inter-cell movement of parts). This kind of machine-part grouping problem has a clear solution. However, in the case that the machine-part incidence matrix is separated partially (i.e., exceptional parts and bottleneck machines exist), there is no clear solution. As Chen and Gerrero(1994) mentioned, the results vary from algorithm to algorithm. Each algorithm generates the solution according to its specific objective.

The proposed algorithm aims at minimizing the number of inter-cell movement of parts considering a machine-cell capacity. Compared with heuristic algorithms previously reported, the proposed algorithm always produced less or equal number of inter-cell movement of parts. Even in the case of having the same result, the use of proposed algorithm may save computational time because it is simple and straight-forward.

In <Table 2>, the machine groups generated without consideration of a machine-cell size have less or equal number of inter-cell movement of parts than those with consideration of a machine-cell size. However, without considering a machine-cell size, the algorithm may produce an unrealistically big size of machine-cell (e.g., examples (1), (5), (6), (7), and (9) in <Table 2>). Sometimes, a big size of machine-cell is generated due to the chaining effect. This problem is resolved by considering a machine-cell size.

## 5. Conclusions

This paper considered a machine-cell size when grouping machine-cells. As a way of grouping machine-cells, a heuristic algorithm was presented. The proposed algorithm was compared with heuristic algorithms previously reported. The comparison showed that the algorithm proposed in this paper is efficient and reliable in minimizing the number of inter-cell movement of parts.

In the case of no consideration of a machine-cell size, the algorithm may produce an unrealistic solution: one big size of machine-cell includes most machines and the rest of machine-cells contain one machine. This problem (sometimes, the chaining effect is involved) can be avoid by considering a machine-cell capacity.

## References

Askin, R. G., Cresswell, S. H., Goldberg, J. B. and Vakharia, A. J. (1991), Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing, *International Journal of Production Research*, 29, 1081-1100.

Boe, W. J. and Cheng, C. H. (19910, A close neighbour algorithm for designing cellular manufacturing systems, *International Journal of Production Research*, 29, 2097-2116.

Burbidge, J. L. (1971), Production flow analysis, *Production Engineer*, 50(4/5), 139-152.

Burbidge, J. L. (1977), A manual method of production flow analysis, *Production Engineer*, 56(10), 34-38.

Carrie, A. S. (1973), Numerical taxonomy applied to group technology and plant layout, *International Journal of Production Research*, 11, 339-416.

Chan, H. M. and Milner, D. A. (1982), Direct clustering algorithm for group formation in cellular manufacturing, *Journal of Manufacturing Systems*, 1, 65-74.

Chandrasekharan, M. P. and Rajagopalan, R. (1986), An ideal seed non-hierarchical clustering algorithm for cellular manufacturing, *International Journal of Production Research*, 24, 451-464.

Chandrasekharan, M. P. and Rajagopalan, R. (1989), GROUP-ABILITY: An analysis of the properties of binary data matrices for group technology, *International Journal of Production Research*, 27, 1035-1052.

Chen, H. G. and Guerrero, H. H. (1994), A general search algorithm for cell formation in group technology, *International Journal of Production Research*, 32, 2711-2724.

Chow, W. S. and Hawaleshka, O. (1993), Minimizing inter-cellular part movements in manufacturing cell formation, *International Journal of Production Research*, 31, 2161-2170.

Gunasingh, R. K. and Lashkari, R. S. (1989), Machine grouping problem in cellular manufacturing systems: An integer programming approach, *International Journal of Production Research*, 27, 1465-1473.

King, J. R. (1980), Machine-component group formulation in production flow analysis: An approach using a rank order clustering algorithm, *International Journal of Production Research*, 18, 213-232.

Kusiak, A. (1987), The generalized group technology concept, *International Journal of Production Research*, 25, 561-569.

Kusiak, A. and Chow, W. S. (1988), Decomposition of manufacturing systems, *IEEE Journal of Robotics and Automation*, 4, 457-471.

Lee, J. L., Vogt, W. G. and Mickle, M. H. (1982), Calculation of shortest paths by optimal decomposition, *IEEE Transaction on System, Man, and Cybernetics*, SMC-12, 410-415.

McAuley, J. (1972), Machine grouping for efficient production, *Production Engineer*, 51(2), 53-57.

McCormick, W. T., Schweitzer, P. J. and White, T. W. (1972), Problem decomposition and data reorganization by cluster technique, *Operations Research*, 20, 993-1009.

Mosier, C. and Taube, L. (1985), Weighted similarity measure heuristic for the group technology clustering problem, *Omega*, 13, 577-579.

Mukhopadhyay, S. K., Gopalakrishnan, A. and Kripalani, M. K. (1995), Moments-based clustering techniques for manufacturing cell formation, *International Journal of Production Research*, 33, 1091-1115.

Rajagopalan, R. and Batra, J. L. (1975), Design of cellular production systems - A graph theoretic approach, *International Journal of Production Research*, 13, 567-579.

Seifoddini, H. and Wolfe, P. M. (1986), Application of the similarity coefficient method in group technology, *IIE Transactions*, 18, 271-277.

Seifoddini, H. (1989), Single linkage vs. average linkage clustering in machine cells formation applications, *Computers and Industrial Engineering*, 16, 419-426.

Shtub, A. (1989), Modelling group technology cell formation as a generalized assignment problem, *International Journal of Production Research*, 27, 775-783.

Singh, N. (1991), *Systems Approach to Computer-Integrated Design and Manufacturing*, John Wiley & Sons.

Srinivasan, G., Narendran, T. T. and Mahadevan, B. (1990), An assignment model for the part families problem in group technology, *International Journal of Production Research*, 28, 145-152.

Srinivasan, G. and Narendran, T. T. (1991), GRAFICS - A nonhierarchical clustering algorithm for group technology, *International Journal of Production Research*, 29, 463-478.

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

Steudel, H. J. and Ballakur, A. (1987), A dynamic programming based heuristic for machine grouping in manufacturing cell formation, *Computers and Industrial Engineering*, 12, 215-222.

Vannelli, A. and Kumar, K. R. (1986), A method for finding minimal bottleneck cells for grouping part-machine families, *International Journal of Production Research*, 24, 387-400.

Vohra, T., Chen, D. S., Chang, J. C. and Chen, H. C. (1990), A network approach to cell formation, *International Journal of Production Research*, 28, 2075-2084.

Wei, J. C. and Kern, G. M. (1989), Commonality analysis: A linear cell clustering algorithm for group technology, *International Journal of Production Research*, 12, 2053-2062.

**박창규**
1986년 고려대학교 산업공학과 학사
1990년 한국과학기술원 산업공학과 석사
1997년 University of Missouri-Columbia, 산업공학과 박사
현재: 울산대학교 경영대학 경영학부 조교수