†원유경* · 원유동**

Two-Phase Approach for Machine-Part Grouping Using Non-binary Production Data-Based Part-Machine Incidence Matrix

Youkyung Won* · Youdong Won**

■ Abstract ■

In this paper an effective two-phase approach adopting modified p-median mathematical model is proposed for grouping machines and parts in cellular manufacturing(CM). Unlike the conventional methods allowing machines and parts to be improperly assigned to cells and families, the proposed approach seeks to find the proper block diagonal solution where all the machines and parts are properly assigned to their most associated cells and families in term of the actual machine processing and part moves. Phase 1 uses the modified p-median formulation adopting new inter-machine similarity coefficient based on the non-binary production data-based part-machine incidence matrix(PMIM) that reflects both the operation sequences and production volumes for the parts to find machine cells. Phase 2 applies iterative reassignment procedure to minimize inter-cell part moves and maximize within-cell machine utilization by reassigning improperly assigned machines and parts to their most associated cells and families. Computational experience with the data sets available on literature shows the proposed approach yields good-quality proper block diagonal solution.

Keywords : Cellular Manufacturing System, Machine-part Grouping, P-median Mathematical Model, Two-phase Approach

논문접수일 : 2007년 03월 08일 논문게재확정일 : 2007년 04월 28일

* 전주대학교 경영학부
** 경남대학교 경영학부
† 교신저자

1. Introduction

Over the past three decades, Group technology (GT) has emerged as an effective management technology for improving the productivity of job shop manufacturing. GT is a method of factory reorganization in which organizational units known as groups of machines complete all the products or parts they make and are equipped with all the processing facilities they need to do so. Cellular manufacturing (CM) is an application of GT principle to manufacturing. It has been well known that the adoption of CM system has led to reduced manufacturing lead times, reduced in-process inventories, reduced set-up costs, improved quality control and flexibility in comparison with the traditional job shop [23, 32].

One of the crucial steps toward designing CM is to create part families and associated machine cells or vice versa. Part family is a collection of parts that have similar operations and require a similar set of machines for the completion of these operations. A set of machines grouped to produce the parts in a specific part family is called the machine cell. The problem of finding part families and their associated machine cells is known as the machine-part grouping (MPG) problem or cell formation (CF) problem in literature.

The fundamental objective of MPG is to find independent machine cells with minimum interaction between cells so that a set of part family having similar or identical processing requirements can be completely produced within a cell. Interaction between cells can be measured with the amount of part moves required outside a specific machine cell so as to complete all the operations of parts. Another main objective of MPG is to accomplish maximum utilization of machines within cells. This paper seeks to find the best configuration of machine cells and part families where inter-cell part moves are minimized and within-cell machine utilization is maximized.

The main input to MPG problem is typically the binary part-machine incidence matrix (PMIM) where each row corresponds to a part and each column to a machine. The elements of binary PMIM can be obtained from the routing information of parts. An $n \times m$ binary PMIM $\mathbf{A}$ $(=[a_{ij}])$ with n part types and m machine types consists of the zero-one element a_{ij} where the element a_{ij} is 1 or 0 depending on whether or not part i requires processing on machine j.

Block diagonalization has been considered as the best approach to form machine cells and part families. In an ideal solution, all the 1's will remain in the diagonal blocks of the matrix and all the zeros in the off-diagonal blocks and these blocks facilitate the grouping of machines and parts into independent cells and families. However, MPG algorithms often fail to find the best block diagonal solution since the algorithms yield the non-optimal block diagonal solutions in which there exist the 1's outside the diagonal (known as exceptional elements) and zeros in the diagonal blocks (known as voids). A 1 lying outside the diagonal blocks indicates processing of a part outside its cell, requiring inter-cell movement of part and a zero in the diagonal blocks indicates the corresponding machine is idle. One of the popular objectives of binary PMIM-based MPG is to minimize the total number of exceptional elements in the off-diagonal blocks.

A number of research papers have been published in the field, seeking block-diagonalization of the binary PMIM. To block-diagonalize the binary PMIM, several approaches and algorithms have been proposed. However, the conventional binary PMIM-based MPG methods do not address the following practical issues [15, 29]:

- sequence of manufacturing operations,
- non-consecutive operations on the same machine,
- volume of inter-cell moves, and
- machine capacity.

Among these factors, sequence of manufacturing operations and volume of inter-cell moves due to the difference in demand of the parts to be manufactured have significant impact on the efficiency of CM system. Since the MPG without considering the operation sequences and production volumes of parts tends to distort the real extent of material handling efforts within and outside the cells, the goal of minimizing binary exceptional elements without considering the operation sequences and production volumes of parts does not necessarily constitute minimization of real material handling [26]. Modeling these factors into MPG at the design stage of CM leads to more realistic solution.

The MPG reflecting the real manufacturing factors above has considerable attention in the recent CM literature. Sarker and Xu [18] and Park and Suresh [17] have provided broad reviews of sequence-based MPG methods. Most of MPG methods use similarity (dissimilarity) coefficient defined between pairs of parts or machines to group parts or machines. A broad review of MPG methods adopting the similarity (dissimilarity) coefficients can be found in Yin and Yasuda [41]. To solve the MPG problem considering the operation sequences and production volumes, a variety of similarity coefficients have been proposed [5, 8, 9, 14, 15, 18, 21, 22, 26, 27, 28]. Park and Suresh [17] addressed some drawbacks using theses similarity coefficients.

The MPG using non-binary PMIM has much attention in the recent CM literature. Harhalakis et al.[6] addressed the importance of capturing shop floor realities in MPG since the methods based on the binary PMIM do not reflect the real manufacturing factors. Instead of the classical binary PMIM, they used the non-binary PMIM of ordinal data considering the operation sequences of parts as the input and proposed a two-stage procedure to minimize the total amount of inter-cell part moves. But since each element of their non-binary PMIM does not represent the actual flows incurred by parts, it cannot be used directly to group machines into cells and parts into families by rearranging the rows and columns of the matrix. Nair and Narendran [15] used Harhalakis et al.'s PMIM [6] considering operation sequences to group machines and parts and proposed a non-hierarchical clustering algorithm based on the weighted inter-machine similarity coefficient considering operation sequences and production volumes of parts. But their algorithm did not use the sequence-based PMIM directly. Wu [39], Yu and Sarker [42, 43], and Xambre and Vilarinho [40] used the inter-machine or inter-cell flow matrix to seek to find the machine cells minimizing the inter-cell flows. Suresh et al. [26] and Park and Suresh [17] used binary-valued precedence matrices of two parts to assess the level of similarity in machine requirements and sequences.

Among the methodologies for solving the MPG problem, mathematical programming approach that attempts to find machine cells and part families by formulating the problem as a linear or nonlinear programming model offers the distinct advantage of being able to incorporate real-world manufacturing data such as ordered sequences of operations, alternative process plans, nonconsecutive part operations on the same machine, setup and processing times. Chu [3] has provided extensive review on the mathematical programming approach for attacking the MPG problem. Kusiak [10] presented the p-median model as a solution methodology for solving the MPG problem and many authors have reported successful applications to the MPG problem with modifications over Kusiak's original p-median model [1, 13, 19, 30, 33, 35]. Unlike the different mathematical programming approaches based on nonlinear objective function [4, 7, 14, 24, 40, 42, 43], the p-median models attempt to optimize linear objective functions with the linear constraints. But most of the existing p-median models deal with the MPG problem based on the binary PMIM.

Recently, Won and Currie [36] have proposed a modified p-median model adopting a new inter-machine similarity coefficient based on the type I production data-based PMIM [38] which incorporates the real manufacturing factors such as the operation sequences with multiple visits to the same machine and production volumes of parts. But their p-median model alone is not sufficient to produce the proper block diagonal solution in which machines and parts are assigned to their most suitable cells and families. Recently, Won [34] and Won and Currie [37] proposed neural network procedures based on the type I production data-based PMIM to find the best non-binary block diagonal solution. But their part-oriented procedures do not consider realistic restriction on the cell size allowable by workers in cells. Furthermore, their neural network algorithms are still sensitive to the order of part input vector presentation, by which the performance of neural network algorithm is severly affected.

In this paper an effective two-phase approach adopting Won and Currie's modified p-median mathematical model [36] is proposed for grouping machines and parts in CM. The proposed approach seeks to find the proper block diagonal solution where all the machines and parts are properly assigned to their most associated cells and families in term of the actual machine processing and part moves. Phase 1 uses the modified p-median formulation to find machine cells. The modified p-median model is not sensitive to the number of part vectors and/or the order of part input vector presentation since it adopts inter-machine similarity coefficient Phase 2 applies iterative reassignment procedure to minimize inter-cell part moves and maximize within-cell machine utilization by reassigning improperly assigned machines and parts to their most associated cells and families. It will be shown that such an ancillary procedure improves the solution quality significantly.

The paper is organized as follows. Section 2 describes the constraints for MPG by introducing the concept of proper block diagonal solution matrix from the types of bottleneck machines and parts based on the non- binary type I production data-based PMIM. Section 3 describes the two-phase approach for grouping machines and parts. In section 4 the proposed approach is illustrated with example data set taken from the literature and compared with existing solution. Section 5 reports the computational results with various data sets available in literature. The last section gives the summary and conclusion of the present study.

2. Constraints for Machine-part Grouping

MPG decision requires the decision on the following parameters or restrictions:

• Number of cells required

Most of the MPG methods require an a priori specification of the number of cells. However, this contradicts the fundamental philosophy of GT that cells exist naturally and that the analyst is to identify them if they exist [15]. At the stage of design, it is only logical that the number of cells should be outcome of the solution procedure and not an input parameter [25] and this justifies to use modified p-median model without the predetermined number of cells for grouping machine cells.

• Cell size

Cell size means the lower and upper bounds on the number of machines allowable in a cell. Many authors assert that an a priori specification of the cell size also contradicts the above-mentioned fundamental philosophy of GT. But the size of a cell needs to be controlled for several reasons, such as available space and socio-logical environment in a cell [28]. In practice, the number of machines in a cell ranges from 2 to 8 [12].

• Singleton cell or family

The definition of cell and family indicates that a cell consists of two or more dissimilar machines and a family contains two or more similar parts. Some methodologies often, however, produce many singleton cells or families which cause inter-cell moves. Considering the philosophy of GT pursuing group production, however, these trivial cells or families must be reassigned into their most associated cells or families. If inter-cell moves can be decreased by reassigning a machine (part) in a singleton cell (family) to its most associated nonsingleton cell (family) within the upper limit of cell size, reassigning a machine (part) in a singleton cell (family) can significantly contribute to minimization of inter-cell flows. MPG algorithm should be able to produce nonsingleton cells and families.

• Empty cell or family

The conventional MPG algorithms often produce empty machine cells or part families. Empty cells (families) are often found if MPG algorithm seeks to maximize an efficiency measure of the block diagonal solution [16]. Since empty cells or part families have no parts to process or no machines to visit, empty cells and families should be reassigned to their most associated cells or families so that actual minimum inter-cell flows and maximum within-cell utilization is accomplished.

Other constraints, such as machine capacity, alternative process plans, and so on, can be included in the MPG procedure. From the discussion of the constraints for MPG, in the present study we attempts to find the rigorous block diagonal solution subject to the following constraints which have often been adopted in literature [11]:

i) The number of machines in a cell cannot exceed the upper limit, U.
ii) Singleton machine cells or part families are not allowed.
iii) Empty machine cells or part families are not allowed.

However, MPG algorithms often unnatural block diagonal solution due to the improperly assigned bottleneck machines and parts. Bottleneck machines are the ones that have more part processing in other cells than their current cell and bottleneck parts are the ones that undergo more flows in other family than their current family. Bottleneck machines and parts critically impact inter-cell flows. If bottleneck machines or parts can be reassigned to their most associated non-singleton cells or families within the upper limit on cell size, those improperly assigned bottleneck machines or parts should be reassigned. To obtain the block diagonal solution where all the machines and parts are properly assigned to their most appropriate cells and families, bottleneck machines are categorized as follows:

- Type I bottleneck machine
  If a machine belonging to a cell has more part processing in other cells, it is called a type I bottleneck machine.
- Type II bottleneck machine
  If a machine belonging to a cell has equal or more part processing in other cells, it is called a type II bottleneck machine.
- Type I bottleneck part
  If a part belonging to a family corresponding to a cell has more flows in other family corresponding to that cell, it is called a type I bottleneck part.
- Type II bottleneck part
  If a part belonging to a family corresponding to a cell has equal or more flows in other family corresponding to that cell, it is called a type II bottleneck part.

Type I bottleneck machine (part) clearly increases inter-cell flows. Type II bottleneck machine (part) does not increases inter-cell flows but impacts within-cell machine utilization. The block diagonal solution matrix that does not have singleton cell (family), empty cells (family) or bottleneck machines (parts) is called the proper block diagonal solution matrix. From the fundamental viewpoint of MPG minimizing inter-cell flows and maximizing within-cell machine utilization, the proper block diagonal solution matrix is the closest to the natural group if the number of cells is not too small and the cell size is not too large. Enhancement procedure proposed in phase 2 uses the bottleneck machines and parts as one of the conditions for stopping the iteration.

The concepts of bottleneck machines and parts, and proper block diagonal solution are illustrated with a production data-based PMIM as shown in <Figure 1> where five machines process six parts. Unlike the conventional binary PMIM that only represents whether or not each part requires processing on machines visited, each non-binary entry in <Figure 1> represents the actual flows incurred by parts. In this system, cell 1 having machines 1,3, and 5 processes part family 1 having parts 2,3,5, and 6, and cell 2 having machines 2 and 4 processes part family 2 having parts 1 and 4. Under the current configuration of machine cells and part families, machine 5 in cell 1 is a type I bottleneck machine since it has more processing on parts in family 2. Similarly, part 5 in family 1 is a type I bottleneck part since it has more flows on machines in cell 2 and part 3 is a type II bottleneck part since it makes more visits on machines in cell 2. Hence the solution matrix in <Figure 1> is an improper block diagonal solution matrix. For this example, let us assume that the upper limit on cell size is 4. <Figure 2> shows the solution matrix after the reassignment of the bottleneck machines and parts. It can be noted that the cell configuration after the reassignment of the bottleneck machines and parts gives less inter-cell flows of 290 units compared with the inter-cell flows of 510 units corresponding to the cell configuration before the reassignment of the bottleneck machines and parts. Within-cell machine utilization is also improved since the number of voids decreases as many as one compared with the cell configuration before the reassignment of the bottleneck machines and parts. This example shows why bottleneck machines and parts are reassigned to their most appropriate cells and families where they have most processing and flows. The solution matrix shown in <Figure 1> gives better solution than the one in shown in <Figure 2> in terms of both inter-cell moves and voids since the former generates inter-cell moves of 510 units and 4 voids while the latter generates inter-cell moves of 290 units and 3 voids.

parts \ machines	1	3	5	2	4
2	200	100			
3	150			100	50
5	90		100	200	
6	50	50	50		
1			160	160	80
4				180	80

〈Figure 1〉 Improper block diagonal solution

parts \ machines	1	3	2	4	5
2	200	100			
6	50	50			50
1			160	80	160
4			180	80	
5	150		100	50	
3	90		200		100

〈Figure 2〉 Proper block diagonal solution

3. Two-Phase Approach to MPG

3.1 Phase 1: Initial grouping phase

The proposed approach begins with an initial *configuration of machine cells. For an initial* setting of machine cells, a solution where machines are randomly allocated to the predetermined number of cells can be used. But since the initial solution close to the optimal cell configuration can lead to better solution within shorter computation time, it is a good approach for cell designer to start such a near-optimal solution of machine cells. In order to find the initial solution of machine cells close to the optimal cell configuration, phase 1 adopts the production data-based p-median mathematical model in Won and Currie [36].

Given an $n \times m$ type I production data- based PMIM $B(=[b_{ij}])$, we adopt the similarity coefficient between pair of machines j and k defined as

$$s_{jk} = \sum_{i=1}^{m} \Gamma(b_{ij}, b_{ik}) \qquad (1)$$

where

$$\Gamma(b_{ij}, b_{ik}) = \begin{cases} +2\min(b_{ij}, b_{ik}) & \text{if } b_{ij}, b_{ik} > 0 \\ -\max(b_{ij}, b_{ik}) & \text{if } either\ b_{ij} > 0 \text{ or } b_{ik} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The rational for the new weighted similarity coefficient is similar to that in Viswanathan [30] except that the non-binary entries reflecting the operation sequences and production volumes of parts are incorporated into the measure. Like Viswanathan's measure dealing with binary PMIM, the proposed measure also aims at block-diagonalizing the initial type I production data-based PMIM by having less zeros inside the diagonal blocks that mean high within-cell utilization of machines, and having less positive entries outside the diagonal blocks that mean less inter-cell flows of parts.

The p-median model adopting the similarity coefficient given in equation (1) is then described as follows [36]):

(PP)

$$\text{Max} \quad \sum_{j=1}^{m} \sum_{k=1}^{m} s_{jk} x_{jk} \qquad (2)$$

$$\text{subject to} \quad \sum_{k=1}^{m} x_{jk} = 1, \quad j = 1, \cdots, m \qquad (3)$$

$$\sum_{j=1}^{m} x_{jk} \geq L x_{kk}, \quad k = 1, \cdots, m \qquad (4)$$

$$\sum_{j=1}^{m} x_{jk} \leq U x_{kk}, \quad k = 1, \cdots, m \qquad (5)$$

$$x_{jk} = 0 \text{ or } 1, \quad j = 1, \cdots, m; \ k = 1, \cdots, m \quad (6)$$

The objective function (2) maximizes the sum of similarity coefficients between machine pairs. Constraint (3) specifies that each machine needs to be assigned to one and only one cell. Constraint (4) ensures that at least L machines should be assigned to cell k only if that cell k is generated and constraint (5) guarantees that at most U machines can be assigned to cell k only if that cell k is generated. These cell size constraints are used to avoid generating too many small cells and too few large cells and can lead to the solution with balanced workload per cell. Constraint (12) ensures the binary solution for machine allocation.

Like in Viswanathan's formulation, the constraint for the predetermined number of cells is omitted since the objective function seeks to find the optimal solution of machine allocation by maximizing the sum of similarities between machine pairs. Once the initial machine cell is found with the above p-median model, its corresponding part family is formed by assigning parts to its most appropriate cell with the following rule [34, 36, 37]:

Part assignment rule

- Find a cell in which each part has most flows on machines and assign the part to that cell.
- If ties occur, assign it to the smallest cell in which it visits most machines.

The primary criterion of part assignment rule seeks to minimize inter-cell part flows and the secondary criterion seeks to both maximize within-cell machine utilization and balance the workload per cell. Thus, the steps of phase 1 are stated as follows :

Phase 1 : Initial grouping

Step 0 : Set the iteration number = 0. Solve the formulation (PP) to find the initial machine cells and their associate part families by assigning parts with the part assignment rule.

3.2 Phase 2 : Reassignment phase

It has been noticed that the objective function in the conventional p-median models based on the binary PMIM is not directly to minimize the number of exceptional elements which are the main source of inter-cell part flows [31]. There is no explicit relationship between the similarity score and the number of exceptional elements. Similarly, the initial solution of machine cells obtained in phase 1 may not be satisfactory from the viewpoint of minimizing the sum of inter-cell flows since the objective function in the proposed p-median model does not seeks to directly minimize the sum of actual inter-cell flows. If an MPG problem has lots of bottleneck machines and parts, the solution found in phase 1 may be unsatisfactory due to the existence of singleton cells (families), empty cells (families), and bottleneck machines (parts)

The objective of phase 2 is to find the proper block diagonal solution of cells and families where all machines and parts are assigned properly to their most associated cells and families so as to both minimize inter-cell part flows and maximize within-cell machine utilization. It has been shown that reassigning improperly assigned machines and parts enhances the solution quality of binary PMIM-based MPG significantly [2, 33]. In phase 2, reassignment procedure is used to further enhance the initial solution so that it finds a proper block diagonal matrix in which each machine cell has a proper block of part family corresponding to it. The reassignment procedure minimizes inter-cell flows and maximizes within-cell utilization by removing undesirable assignment of machines and parts. Since the sum of similarity coefficients and the total amount of actual inter-cell flows are highly correlated, the proposed p-median model tends to yield the solution of machine allocation minimizing inter-cell flows. But the solution by the p-median model does not tell much about the proper assignment of machines and parts. To enhance the solution so that all the machines and parts are properly assigned, the proposed reassignment procedure reuses the cell size restriction.

It is evident that reassigning type I bottleneck machines or parts to other cell or family where it has most processing or flows leads to minimizing inter-cell flows. In the mean time, reassigning type II bottleneck machine or part to other cell or family does not impact minimizing inter-cell flows. But reassigning type II bottleneck machines or parts to other cell or family where it has most visits can lead to maximizing within-cell utilization as shown in previous subsection. The concept of proper block diagonal solution matrix and type I bottleneck machines or parts gives the condition for stopping the implementation of phase 2.

Stopping condition 1

- All the cells (families) are non-singleton and non-empty,
- the number of cells which are both non- singleton and non-empty is equal to the number of families which are both non- singleton and non-empty, and
- no type I bottleneck machines (parts) are found.

Then, the step for checking the stopping condition of phase 2 can be stated as follows :

Phase 2 : Reassignment phase

Step 1 : Check the stopping condition 1. If it is true, stop. Otherwise, increase the iteration number by 1 and go to step 2.1 of bottleneck machine reassignment.

Since the primary objective of MPG is to create independent cells minimizing inter-cell moves, type I bottleneck machines and parts should be reassigned. Type II bottleneck machines and parts also need to be reassigned since reassigning them can lead to higher within-cell utilization. To find the proper block diagonal solution, the machines (parts) belonging to singleton cells (family) should be reassigned to their most appropriate non-singleton cells (family). Since the reassignment of type II bottleneck machines and parts may create large cells difficult to control, however, the procedure for checking if the cell size restriction is violated is needed. To avoid the generation of singleton cells (families), bottleneck machine (part) is reassigned to its most appropriate non-singleton cell (family) where it undergoes most flows. If ties occur, the smallest cell (family) is selected.

From the discussion above, we can define the improperly assigned bottleneck machines that need to be reassigned. A machine is called the improperly assigned bottleneck machine if it belongs to one of the following categories :

- type I bottleneck machine,
- type II bottleneck machine, or
- machine belonging to singleton cell.

The procedure for reassigning improperly assigned bottleneck machines can then be stated as follows :

Bottleneck machines reassignment procedure

Step 2.1 : Check if a machine is an improperly assigned bottleneck machine. If it is true, go to step 2.2. If it is false, repeat this step for remaining machines.

Step 2.2 : Check if the cell size restriction is not violated by reassigning that improperly assigned bottleneck machine to its most associated non-singleton cell. If it is true, reassign it and go to step 2.1. If it is false, do not reassign that bottleneck machine and go to step 2.1.

Similarly, we can define the improperly assigned bottleneck parts that need to be reassigned. A part is called the improperly assigned bottleneck part if it belongs to one of the following categories :

- type I bottleneck parts,
- type II bottleneck parts, or
- parts belonging to singleton families.

Reassigning improperly assigned bottleneck parts is implemented in the same way as reassigning improperly assigned bottleneck machines. However, the step for checking if the part family size limit is violated is not assigned into this procedure since, unlike the machine cell size limit prespecified by cell designer, it cannot be determined in advance before the optimal configuration of cells and families is found. The procedure for reassigning improperly assigned bottleneck parts can be stated as follows:

Bottleneck parts reassignment procedure

Step 3 : Check if a part is an improperly assigned bottleneck part. If it is true, reassign them to their most associated non-singleton family and repeat this step for remaining parts. If it is false, do not reassign it and repeat this step for remaining parts.

However, the steps 2 and 3 may be repeated infinitively if the stopping condition 1 is satisfied but the cells size restriction is violated. In that case, the configuration of machine cells and part families remains the same as the one in previous iteration since the improperly assigned bottleneck machines can not be reassigned. Hence, another stopping condition is needed to check if the configuration of block diagonal solution does not change compared with the one in previous iteration.

Stopping condition 2

- The configuration of machine cells and part families is the same as the one in the previous iteration.

Then the whole algorithm is stated as follows:

Phase 1: Initial grouping

Step 0 : Set the iteration number = 0. Solve the model (PP) to find the initial machine cells and part families.

Phase 2 : Reassignment phase

Step 1 : Check the stopping condition 1. If it is true, stop. Otherwise, increase the iteration number by 1 and go to step 2.

Step 2 : Use bottleneck machines reassignment procedure to reassign improperly assigned bottleneck machines.

Step 3 : Use bottleneck parts reassignment procedure to reassign improperly assigned bottleneck parts.

Step 4 : Check the stopping condition 2. If it is true, stop. Otherwise, go to step 1.

4. Illustrative Example and Comparison

To show the application of the proposed approach, a data set taken from Vakharia and Wemmerlöv (1990) will be considered. The data set contains 19 part types and 12 machine types. Since all the parts have equal production volumes in their data set, it is assumed that the production volume of each part is the unity. The upper limit on the cell size is set at 7.

Iteration = 0

Step 0 : Set the iteration number = 0. Implementing the p-median model (PP) yields the following solution with five machine cells, $MC_1 = \{1,6,9\}$, $MC_2 = \{2, 3, 5\}$, $MC_3 = \{4,8\}$, $MC_4 = \{7,10\}$, and $MC_5 = \{11,12\}$ and five part families, $PF_1 = \{4,11\}$, $PF_2 = 8$, $PF_3 = \{1, 2, 3, 7, 9, 10\}$, $PF_4 = \{5, 6, 14, 17, 18\}$, and $PF_5 = \{12, 13, 15, 16, 19\}$. <Figure 3> shows the type I production data-based solution matrix corresponding to the configuration of machine cells and part families obtained at the end of phase 1.

Step 1 : Since machine 1 in cell 1 has more processing in cell 3, stopping condition 1 is violated. Set the iteration number at 1 and go to step 2.

Iteration = 1

Step 2 : From the solution matrix shown in <figure 3>, it can be noticed that machines 1,6 and 9 are type I bottleneck machines and machines 2,3, and 5 are the type II bottleneck machines belonging to singleton part family 2. Since the cell size limit is not violated even if bottleneck machines 1, 2, 3, 5, and 9 are reassigned to cell 3 and bottleneck machine 6 is reassigned to cell 4, those bottleneck machines are reassigned. <Figure 4> shows the solution matrix after the reassignment. The reassignment of bottleneck machines results in the system with three nonempty machine cells and five part families : $MC_1 = \varnothing$, $MC_2 = \varnothing$, $MC_3 = \{4, 8, 1, 2, 3, 5, 9\}$, $MC_4 = \{7, 10, 6\}$, $MC_5 = \{11, 12\}$, and $PF_1 = \{4, 11\}$, $PF_2 = \{8\}$, $PF_3 = \{1, 2, 3, 7, 9, 10\}$, $PF_4 = \{5, 6, 14, 17, 18\}$, and $PF_5 = \{12, 13, 15, 16, 19\}$. Note that reassigning the bottleneck machines makes cells 1 and 2 empty.

Step 3. Since parts 4,8, and 11 are the improperly assigned bottleneck parts belonging to the part families corresponding to empty machine cells, those parts should be reassigned. Part 17 is a type II bottleneck part and also needs to be reassigned. <Figure 5> shows the solution matrix after the reassignment of bottleneck parts.

Step 4. Since the stopping condition 2 is violated, go to step 1.

Step 1. Since machine 7 is a type I bottleneck machine, the stopping condition 1 is violated. Set the iteration number at 2 and go to step 2.1.

Iteration = 2

Step 2. Machine 7 needs to be reassigned. However, bottleneck machine 7 is reassigned to its most associated cell 1, the cell size limit is violated. Therefore, do not reassign bottleneck machine 7.

Step 3. No bottleneck part is found.

Step 4. Since the configuration of cells and families is not changed, the stopping condition 2 is satisfied and the algorithm stops.

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

〈Figure 3〉 Solution matrix at step 0 of iteration 0

To show the comparative efficiency of the proposed approach, a well-known ill-structured data set 43 part types and 16 machine types taken from Gupta and Seifoddini [5] will be illustrated and compared with the existing solution from the reference algorithm. The proposed procedures have been coded in PASCAL and implemented on a Pentium III PC with 1 GHz using an extended version of HYPER LINDO which can solve the integer linear programming problem with 1,000 or less binary variables. Limit on the computation time is taken as the number of pivot iterations automatically preset by the HYPER LINDO program given an instance of optimization problem. The lower and upper limits on the cell size are set at 2 and 6, respectively.

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

〈Figure 4〉 Solution matrix at step 2 of iteration 1

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

〈Figure 5〉 Solution matrix at step 3 of iteration 1

To compare the goodness of non-binary block diagonal solution, the weighted grouping efficiency (WGCI) suggested in Won and Currie [34, 36, 37] is selected to evaluate the overall performance in terms of inter-cell part moves. The WGCI measure is defined as follows :

$$\text{WGCI} = 1 - \frac{\text{the sum of exceptional } b_{ij}\text{s}}{\sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij}} \tag{7}$$

To evaluate the performance in terms of within-cell machine utilization, the number of voids has been applied together.

〈Table 1〉 Initial grouping result

cell no.	machines	parts
1	1, 15	33
2	2, 9, 16	2, 4, 10, 18, 28, 32, 37, 38, 40, 42
3	3, 14	6, 7, 17, 34, 35, 36
4	4, 5, 8	3, 5, 8, 9, 15, 16, 19, 21, 23, 29, 41
5	6, 10	1, 12, 13, 14, 26, 31, 39, 43
6	7, 13	25
7	11, 12	11, 20, 22, 24, 27, 30

<Table 1> shows the initial configuration of machine cells and part families. Initial grouping by the p-median model yields seven-cell solution. However, it can be noticed that phase 1 produces two singleton part families consisting of just one part. <Figure 6> shows the proper block diagonal solution matrix at the end of phase 2 in which all machines and parts are assigned to their most associated cells and families.

Parts \ Machines	2	9	16	1	3	14	4	5	8	13	15	6	10	7	11	12
2	300	600	300			300			300			300				
4		75														
10	2600	1300	1300													
18		339	339													
28	320	640							320							
32	430	1290	860									860				
37	3000	4500	3000	1500					3000			3000				
38	750	2250	1500						1500							
40	2600	2600										2600				
42	2300	1150	1150	1150								1150				
6						1200						1200				
7			3000		3000							3000				
17					1800	1800						1800				
34					275							275				
35					500	500										
36					600											
3									1000	1000					1000	
5							1000	1000			1000					
8								1500	750			750				
9							10000	10000	10000						10000	
15								14000	14000							
16								39								
19							390	780	780		390	1560				
21							810	810	810		810					
23							5	10	5			10				
29							1500	1500								
41								1239	2478		1239					
14							1500	3000				4500				
33								1000			1000	1000				
43								1239	2478		2478	3717				
12									1150			1150	1150			
1									100			100	100	100		
13												2478	1239	1239		
26													750			
31									310				310			
39												5000	5000			
25													390	390		
11									1239							1239
20									304						304	
22								1200								1200
24									70	70					70	70
27									39						39	78
30															11300	11300

〈Figure 6〉 Solution matrix at the end of phase 2

Machines

Parts	1	2	9	16	3	6	10	14	15	4	5	8	11	7	12	13
2		300	600	300		300		300				300				
4			75													
10		2600	1300	1300												
18			339	339												
28		320	640									320				
32		430	1290	860		860										
37	1500	3000	4500	3000		3000						3000				
38		750	2250	1500								1500				
40		2600	2600			2600										
42	1150	2300	1150	1150		1150										
6						1200		1200								
7				3000	3000	3000										
13						2478	1239							1239		
14						4500				1500	3000					
17					1800	1800		1800								
19						1560			390	390	780	780				
26							750									
31							310					310				
43					275	275										
35					500			500								
36					600											
39						5000	5000									
43						3717			2478		1239	2478				
1						100	100					100		100		
3												1000	1000			1000
5									1000	1000	1000					
8						750					1500	750				
9										10000	10000	10000	10000			
11												1239			1239	
12						1150	1150					1150				
15											14000	14000				
16											39					
20												304	304			
21									810	810	810	810				
23						10				5	10	5				
24												70	70		70	70
27												39	39		78	
29										1500	1500					
30													11300		11300	
33						1000			1000		1000					
41									1239		1239	2478				
25							390							390		
22											1200				1200	

<Figure 7> Solution matrix by Gupta and Seifoddini's algorithm

Gupta and Seifoddini's ill-structured data set is addressed as the benchmark test material for comparative purpose in our computational experiment. <Figure 7> shows the type I production data-based solution matrix based on the methods by Gupta and Seifoddini. The solution by Gupta and Seifoddini's algorithm yields five machine cells and five part families, and the WGCI of 77.86%. The solution by the proposed approach produces the same number of cells and families as the solution by Gupta and Seifoddini's algorithm with higher value of WGCI of 80.48%. Within-cell machine utilization in terms of the number of voids is also higher since Gupta and Seifoddini's algorithm yields 94 voids, whereas the proposed approach yields 70 voids. The solution matrix by Gupta and Seifoddini's algorithm generates one singleton machine cell and two singleton part families, i.e., $PF_4 = \{7\}$, $PF_4 = \{25\}$, and $PF_5 = \{22\}$. Since machines 7 belonging to singleton cell 4 has more processing for the parts belonging to family 2, and machines 12 and 13 belonging to cell 5 corresponding to singleton part family 5 has more processing for the parts belonging to family 3, those improperly assigned machines should be reassigned to reduce inter-cell flows and this implies that the number of cells needs to be decreased. Parts belonging to singleton families also should be reassigned. It is interesting to note that machine 13 belonging to cell 5 is not even assigned any operations of parts to process in that cell. Furthermore, the portion of operations on parts belonging to singleton families is very small. On the contrary, our approach produces the proper block diagonal solution with higher WGCI and fewer voids under the same number of cells as Gupta and Seifoddini's algorithm. This justifies reassigning improperly assigned bottleneck machines and parts to their most associated cells and families.

5. Computational Experiment

The purpose of computational experiment with the proposed approach is to report the efficiency of the proposed two-phase procedure on various problem sets with different data structure. The solutions found with the proposed two-phase procedure can be used as a benchmark for future comparative study since the proper block diagonal solutions to those data sets have never been reported in terms of WGCI measure.

<Table 2> reports the computational result with fourteen test data sets taken from literature. The lower limit on the cell size is set at 2 in all problems so as to avoid the formation of singleton cells and the upper limit on the cell size can be seen in column 3. Column 4 reports the number of iterations implemented until the stopping conditions of phase 2 are satisfied and column 5 shows the number of cells (families) found after the algorithm is terminated. The rightmost column reports the value of WGCI in percentage. It can be noticed from the table that it did not take many iterations to stop the algorithm since the largest number of iterations required to terminate the steps of phase 2 is 3. In problems 8, 9, 13 and 14 the proposed p-median model (PP) find the best proper block diagonal solution since no further iterations of phase 2 are required.

The average value of the WGCI measure applied to 14 data sets ranging from well-s tructured data set to ill-structured data set is 84.54%. Lee and Garcia-Diaz [13] have reported the performance of their network-flow based

〈Table 2〉 Computational result with the data sets taken from literature

Source	Size	U	No. of phase 2 iterations	No. of cells	WGCI(%)
1. Selvam and Balasubramanian [22]	5×10	5	2	2	100.00
2. Gupta and Seifoddini [5]	43×16	6	3	5	80.48
3. Harhalakis et al. [6]	20×20	5	3	5	84.75
4. Vakharia and Wemmerlöv [28]	19×12	7	2	3	73.21
5. Seifoddini and Djassemi [21]	41×30	10	3	7	92.50
6. Seifoddini and Djassemi [21]	41×30	10	3	7	87.92
7. Lee and Garcia-Diaz [13]	16×12	5	3	3	72.12
8. Nair and Narendran [15]	7×7	3	0	3	76.92
9. Nair and Narendran [15]	20×8	4	0	3	80.49
10. Nair and Narendran [15]	12×10	5	2	4	80.70
11. Nair and Narendran [15]	40×25	10	3	8	77.66
12. Wu [39]	6×8	5	2	2	96.38
13. Wu [39]	13×13	5	0	4	90.81
14. Won and Lee [38]	5×5	3	0	2	89.62
				average	84.54

procedure applied to real industry data. Since they did not use the performance measure based on the type I production data-based PMIM, the fair comparison with the proposed algorithm is impossible. But the cell efficiency they used as the performance measure to evaluate the quality of the solutions obtained considers the ratio of the total number of part moves within cells to the total number of part moves in the whole system. In this regard the cell efficiency is very close to the WGCI. Their method produced the cells and families with the cell efficiency ranging from 72.3% to 82.8%. Considering the average WGCI by the proposed approach applied to a variety of well-structured or ill-structured data sets, it produces good-quality proper block diagonal solutions.

6. Summary and Concluding Remarks

In this paper a two-phase approach adopting the modified *p*-median model is developed to group machines and parts in CM. To find the machine cells and part families maximizing or minimizing efficiency measure of the block diagonal solution, most of the conventional algorithms admit generating the machine cells and families where machines and parts are improperly assigned. This often leads to the improper block diagonal solution where

- machine belonging to a cell has more processing for the parts corresponding to other cell,
- part belonging to a family has more flows in other family,
- many singleton machine cells (part families) consisting of a machine (part) are found, or
- empty cells are found since the machines in those cells are not assigned any processing of parts.

Considering the fundamental philosophy of GT that seeks to exploit the similarities between machine pairs or part pairs, the improper block diagonal solution implies that the resulting CM system does not fully utilize the similarities between machine pairs or part pairs. Unlike the conventional MPG algorithms allowing the machines and parts to be improperly assigned, our two-phase approach seeks to find the configuration of machine cells and part families where all the machines and parts are properly assigned to their most associated cells and families within the limit on the cell size.

To find the initial solution of machine cells and part families, phase 1 adopts the modified p-median formulation which uses a new similarity coefficient based on non-binary production data-based PMIM that captures shop flow reality as follows:

- operation sequences of parts,
- non-consecutive multiple visits to the same machines, and
- production volumes.

The modified formulation does not require the number of cells to be predetermined in phase 1. To minimize inter-cell part moves and maximize within-cell machine utilization, phase 2 uses the solution from phase 1 to find the proper block diagonal solution by reassigning improperly assigned machines and parts to their most associated cells and families. Such an ancillary enhancement procedure leads to significant improvement of the initial block diagonal solution by removing the above-mentioned improper assignment of machines and parts.

Computational results applied intermediate-size data sets available in literature show that the proposed approach finds good-quality proper block diagonal solutions within short iteration of algorithm. The proposed algorithm can also be extended to more complicated MPG problem so that it can incorporate the manufacturing factors, such as alternative process plans, replicate machines, and machine capacity.

References

[1] Ben-Arie, D. and P.T. Chang, "An extension to the p-median group technology algorithm," *Comuters and Operations Research*, Vol.21, No.2(1994), pp.119-125.

[2] Chen, S.J. and C.S. Cheng, "A new neural network-based cell formation algorithm in cellular manufacturing," *International Journal of Production Research*, Vol.33, No.2(1995), pp.293-318.

[3] Chu, C.H., "Recent advances in mathematical programming for cell formation," *Planning, Design, and Analysis of Cellular Manufacturing Systems*, Kamrani, A.K. Parsaei, H.R. and Liles, D.H.(Eds.), Elsevier Science B.V. Amsterdam, 1995, pp.1-46.

[4] Adenso-Díaz, B., S. Lozano, and I. Eguía, "Part-machine grouping using weighted similarity coefficients," *Computers and Industrial Engineering*, Vol.48 No.2(2005), pp.553-570.

[5] Gupta, T. and H. Seifoddini, "Production data based similarity coefficient for machine-part grouping decisions in the design of a cellular manufacturing system," *International Journal of Production Research*, vol.28 No.7(1990), pp.1247-1269.

[6] Harhalakis, G., R. Nagi, and J.M. Proth, "An efficient heuristic in manufacturing cell formation for group technology," *In-*

ternational Journal of Production Research, Vol. 28, No.1(1990), pp.185-198.

[7] Jayaswal, S. and G.K. Adil, "Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings," *International Journal of Production Research*, Vol.42, No.12(2004), pp.2419-2433.

[8] Jeon, G.W. and H.R. Leep, "Forming part families by using genetic algorithm and designing machine cells under demand changes," *Computers and Operations Research*, Vol.33, No.1(2006), pp.263-283.

[9] Kiang, M.Y., U.R. Kulkarni, and K.Y. Tam, "Self-organizing map network as an interactive clustering tool an application to group technology," *Decision Support Systems*, Vol.15, No.4(1995), pp.351-374.

[10] Kusiak, A., "The part families problem in flexible manufacturing systems," *Annals of Operations Research*, Vol.3(1985), pp.279-300.

[11] Kusiak, A., W.J. Boe, and C.H. Cheng, "Designing cellular manufacturing systems : branch-and-bound and A* approaches," *IIE Transactions*, Vol.25, No.4(1993), pp.46-56.

[12] Kuisak A. and C.H. Chow, "A branch-and-bound algorithm for solving the group technology problem," *Annals of Operations Research*, Vol.26(1990), pp.415-431.

[13] Lee, H. and A. Garcia-Diaz, "Network flow procedures for the analysis of cellular manufacturing systems," *IIE Transactions*, Vol.28, No.4(1996), pp.333-345.

[14] Lozano, S., D. Canca, F. Guerrero and J.M. Garcia, "Machine grouping using sequence-based similarity coefficients and neural networks," *Robotics and Computer-Integrated Manufacturing*, Vol.17, No.5 (2001), pp.399-404.

[15] Nair, G.J. and T.T Narendran, "CASE : A clustering algorithm for cell formation with sequence data," *International Journal of Production Research*, Vol.36, No.1 (1998), pp.157-179.

[16] Ng, S. M., "On the characterization and measure of machine cells in group technology," *Operations Research*, Vol.44, No.5(1996), pp.735-744.

[17] Park, S. and N.C. Suresh, "Performance of Fuzzy ART neural network and hierarchical clustering for part-machine grouping based on operation sequences," *International Journal of Production Research*, Vol.41, No.14(2003), pp.3185-3216.

[18] Sarker, B.R. and Y. Xu, "Operation sequences-based cell formation methods: a critical survey," *Production Planning and Control*, Vol.9, No.8(1998), pp.771-783.

[19] Sarker, B.R. and Y. Xu, "Designing multi-product lines : job routing in cellular manufacturing systems," *IIE Transactions*, Vol.32, No.3(2000), pp.219-235.

[20] Seifoddini, H. and M. Djassemi, "Merits of production volume based similarity coefficient in machine cell formation," *Journal of Manufacturing Systems*, Vol.14, No.1(1995), pp.36-44.

[21] Seifoddini, H. and C.P. Hsu, "Comparative study of similarity coefficients and clustering algorithms in cellular manufacturing," *Journal of Manufacturing Systems*, Vol.13, No.2(1994), pp.119-127.

[22] Selvam, R.P. and K. N. Balasubramanian, "Algorithmic grouping of operation sequences," *Engineering Costs and Pro-*

duction Economics, Vol.9, No.1-3(1985), pp.125-134.

[23] Singh, N. and D. Rajamani, *Cellular Manufacturing Systems : Design, Planning, and Control*, Chapman & Hall. London, 1996.

[24] Solimanpur, M., P. Vrat and R. Shankar, "Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing," *European Journal of Operational Research*, Vol.157, No.3(2004), pp.592-606.

[25] Srinivasan, G., T.T. Narendran and B. Mahadevan, "An assignment model for the part-families problem in group technology," *International Journal of Production Research*, Vol.28, No.1(1990), pp.145-152.

[26] Suresh, N.C., J. Slomp and S. Kaparthi, "Sequence-dependent clustering of parts and machines : a Fuzzy ART neural network approach," *International Journal of Production Research*, Vol.37, No.12(1999), pp.2739-2816.

[27] Tam, K.Y., "An operation sequence based similarity coefficients for part families formations," *Journal of Manufacturing Systems*, Vol.9, No.1(1990), pp.55-68.

[28] Vakharia, A.J. and U. Wemmerlöv, "Designing a cellular manufacturing system : a material flow approach based on operation sequences," *IIE Transactions*, Vol.22, No.6(1990), pp.84-97.

[29] Venugopal, V. and T.T. Narendran, "A genetic algorithm approach to the machine-component grouping problem with multiple objectives," *Computers and Industrial Engineering*, Vol.22, No.4(1992), pp.469-480.

[30] Viswanathan, S., "A new approach for solving the p-median problem in group technology," *International Journal of Production Research*, Vol.34, No.10(1996), pp.2691-2700.

[31] Wang, J. and C. Roze, "Formation of machine cells and part families in cellular manufacturing : an experimental study," *International Journal of Production Research*, Vol.35, No.5(1997), pp.1259-1286.

[32] Wemmerlöve, U. and D.J. Johnson, "Cellular manufacturing at 46 user plants : implementation experiences and performance improvements," *International Journal of Production Research*, Vol.35, No.1(1997), pp.29-49.

[33] Won, Y., "Two-phase approach to GT cell formation using efficient p-median formulations," *International Journal of Production Research*, Vol.38, No.7(2000), pp. 1601-1613.

[34] Won, Y., "Effective artificial neural network approach for non-binary incidence matrix-based part-machine grouping," *Korean Operations Research and Management Science Society Journal*, Vol.31, No.4(2006), pp69-87.

[35] Won, Y. and K.R. Currie, "Efficient *p*- median mathematical programming approaches to machine-part grouping in group technology manufacturing," *Engineering Optimization*, Vol.36, No.5(2004), pp.555-573.

[36] Won, Y. and K.R. Currie, "An effective *p*-median model considering production factors in machine cell/part family formation," *Journal of Manufacturing Systems*, Vol.25, No.1(2006), pp.58-64.

[37] Won, Y. and K.R. Currie, "Fuzzy ART/ RRR-RSS : a two-phase neural network

nlgorithm for part-machine grouping in cellular manufacturing," *International Journal of Production Research*, Vol.45, No.9 (2007), pp.2073-2104.

[38] Won, Y. and K.C. Lee, "Group technology cell formation considering operation sequences and production volumes," *International Journal of Production Research*, Vol.39, No.9(2001), pp.2755-2768.

[39] Wu, W., "A concurrent approach to cell formation and assignment of identical machines in group technology," *International Journal of Production Research*, Vol.36, No.8(1998), pp.2099-2114.

[40] Xambre, A.R. and P.M. Vilarinho, "A simulated annealing approach for manufacturing cell formation with multiple identical machines," *European Journal of Operational Research*, Vol.151, No.2(2003), pp.434-446.

[41] Yin, Y. and K. Yasuda, "Similarity coefficient methods applied to the cell formation problem : A taxonomy and review," *International Journal of Production Economics*, Vol.101, No.2(2006), pp.329-352.

[42] Yu, J. and B.R. Sarker, "Directional decomposition heuristic for a linear machine-cell location problem," *European Journal of Operational Research*, Vol.149, No.1(2003), pp.142-184.

[43] Yu, J. and B.R. Sarker, "A directional decomposition heuristic for one-dimensional, non-equidistant machine-cell location problems," *Computers and Operations Research*, Vol.33, No.1(2006), pp.64-92.