

A MATHEMATICAL APPROACH FOR
DESIGNING CELL FORMATION IN
CELLULAR MANUFACTURING SYSTEMS

Zahir Albadawi

A Thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

September 2003

© Zahir Albadawi, 2003

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-83880-3

Our file Notre référence

ISBN: 0-612-83880-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

ABSTRACT

A Mathematical Approach for Designing Cell Formation in Cellular Manufacturing Systems

Zahir Albadawi

One of the major steps in designing cellular manufacturing systems is to form cells. This involves identification of machine groups and part families. This thesis presents a new approach for optimally grouping the original set of machines into distinct uncorrelated cells. The approach involves two phases. In the first phase, machine cells are formed by applying factor analysis to the matrix of similarity coefficients. In the second phase, integer programming is used to assign parts to the cells. Because of its mathematical basis, the approach assures an optimal cell formation through the generation of uncorrelated cell clusters and the property of variance maximization. Using data set examples, the approach was found to perform very well in terms of a number of objective criteria. Moreover, the approach compares favorably to well-known existing methods. Other advantages such as efficient computational performance, and flexibility during cell formation make this approach very powerful.

ACKNOWLEDGMENTS

I would like to thank my supervisors, Professor Mingyuam Chen and Professor Hamdi Bashir, for their mentorship, for their encouragement and support, and for their insightful guidance throughout the course of my research and throughout the process of writing this thesis.

I would like to express my heartfelt thanks to Tina Ehtiati for all the support and motivation she gave me throughout my research work. For me, Tina has been a great example of academic excellence and source of inspiration. My gratitude to all my friends at the Mechanical & Industrial Department of Concordia University and elsewhere for their generous support and friendship, specially Pooriya Sadeghi and his lovely wife, Neda Ehtiati.

Finally, I would like to acknowledge my beloved parents for their continuous love and support, and my brothers, Dr. Nazir Albadavi who lost a year of brave battle against fatal cancer last year, for introducing me to the beauty of science of mathematics at a very early age, Dr. Amir Albadavi for being a constant source of motivation and a role model of hard work, perseverance and success in both personal and professional life, and Mr. Samir Albadavi for his tireless support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES	viii
------------------------------	-------------

LIST OF TABLES	x
-----------------------------	----------

1 Introduction	1
1.1 Motivation.....	1
1.2 Cellular Manufacturing	2
1.3 Benefits of Cellular Manufacturing	3
1.4 Design of Cellular Manufacturing Systems.....	5
1.5 Research Objectives	9
1.6 Research Assumptions	9
1.7 Outline of the Thesis.....	10
2 Literature Review	12
2.1 Array-Based Clustering Algorithms	13
2.1.1 Rank Order Clustering (ROC).....	14
2.1.2 Direct Clustering Algorithm (DCA).....	14
2.1.3 Bond Energy Algorithm (BEA).....	15
2.1.4 Comparison of Array-Based Methods.....	16
2.2 Hierarchical Clustering Algorithms	17

2.2.1 Choice of Similarity Measure.....	18
2.2.2 Single Linkage Clustering (SLC)	19
2.2.3 Average Linkage Clustering (ALC)	20
2.2.4 Complete Linkage Clustering (CLC)	21
2.2.5 Linear Cell Clustering (LCC)	21
2.2.6 Comparison of Hierarchical Algorithms	22
2.3 Methods Based on Artificial Intelligence	23
2.3.1 Artificial Neural Networks	23
2.3.2 Fuzzy Logic.....	24
2.3.3 Genetic Algorithms and Simulated Annealing	25
2.4 Mathematical Programming Methods	27
2.4.1 P-Median Model	28
2.4.2 Assignment Model	28
2.4.3 Quadratic Programming Model.....	28
2.4.4 Assignment Allocation Algorithm (AAA)- Nonlinear Model	29
2.5 Other Heuristics Methods.....	30
2.6 Summary	31
3 Factor Analysis	32
3.1 Introduction.....	32
3.2 Principal Component Analysis.....	35
3.3 Rotation.....	41
3.4 Summary	49

4	A Proposed Mathematical Approach For Cell Formation	50
4.1	Introduction.....	50
4.2	Phase 1: Machine Cells Formation Using Factor Analysis.....	51
4.2.1	Generation of a Similarity Coefficient Matrix	51
4.2.2	Extraction of Initial Cells.....	53
4.2.3	Optimizing the Initial Cell Formation	56
4.3	Phase 2: Assigning Parts to Cells.....	58
4.4	Summary	61
5	Performance Evaluation	62
5.1	Evaluation Procedure.....	62
5.1.1	Percentage of Exceptional Elements	63
5.1.2	Machine Utilization.....	64
5.1.3	Grouping Efficiency.....	64
5.2	The Data Sets	65
5.3	Computational Results.....	66
5.4	Summary	76
6	Conclusions	78
6.1	Overview and Discussion.....	78
6.2	Future Work.....	79
	REFERENCES	80
	APPENDIX	89

LIST OF FIGURES

Figure 1.1: Example of a machine-part matrix.	6
Figure 1.2: Example of an optimal clustering.	7
Figure 1.3: Example of real problem with voids and exceptional elements.	8
Figure 2.1: Categories of cell formation approaches.	12
Figure 3.1: Illustration of the purpose of rotation: The goal here is to rotate the axes so that each dot is close to only one of the two rotated axes.	45
Figure 3.2: Orthogonal and oblique rotation.	47
Figure 4.1: Machine-component matrix.	53
Figure 4.2: Similarity coefficient matrix.	53
Figure 4.3 Machine and cells after rotation	58
Figure 4.4: Final cell formation.	59
Figure 5.1: The result of applying the proposed method on problem 1 (a) initial matrix, (b) final matrix.	67
Figure 5.2: The result of applying the proposed method on problem 2 (a) initial matrix, (b) final matrix.	68
Figure 5.3: The result of applying the proposed method on problem 3 (a) initial matrix, (b) final matrix.	69

Figure 5.4: The result of applying the proposed method on problem 4 (a) initial matrix, (b)	
final matrix.	70
Figure 5.5: The result of applying the proposed method on problem 5(a) initial matrix, (b) final	
matrix	71
Figure 5.6: The result of applying the proposed method on problem 6 (a) initial matrix, (b)	
final matrix.	72
Figure 5.7: The result of applying the proposed method on problem 7 (a) initial matrix, (b)	
final matrix.	73
Figure 5.8: The result of applying the proposed method on problem 8 (a) initial matrix, (b)	
final matrix.	74

LIST OF TABLES

Table 4.1: The computed eigenvalues.....	55
Table 4.2: Machine-cell matrix.	56
Table 4.3: Machine-cell matrix after rotation.	57
Table 5.1: Features and sources of cell formation problems.....	66
Table 5.2: The performance of the algorithm based on the computational results.....	76

INTRODUCTION

We started assembling a motorcar in a single factory. Then as we began to make parts, we began to departmentalize so that each department would do only one thing. As the factory is now organized each department makes only a single part or assembles a part. A department is a little factory in itself. The part comes into it as raw material or as a casting, goes through the sequence of machines and heat treatments, or whatever may be required, and leaves that department finished. *

1.1 Motivation

Batch manufacturing is a dominant manufacturing activity in the world, generating a great deal of industrial output. It accounts for 60 to 80 percent of all manufacturing activities [16]. The major difficulties in batch manufacturing are due to high level of product variety and small manufacturing lot sizes. The product variations present design engineers with the problem of designing many different parts. The impact of these product variations in manufacturing is high investment in equipment, high tooling costs, complex scheduling and loading, lengthy setup time and costs, excessive scrap, and high quality control costs. For this purpose, some innovative methods are needed to reduce product cost. What is also needed is a higher level of integration of the design and manufacturing activities in a company [24]. These factors have led many conventional batch manufacturing systems to convert to cellular manufacturing systems [23].

* Henry Ford, My Life and Work (1922)

Cellular manufacturing, which is an application of group technology (GT) in manufacturing, has been recognized as one of the most recent technological innovations in job-shop or batch-type production to gain economic advantages similar to those of mass production [19]. Converting to cellular manufacturing systems becomes even more pressing in today's severe competitive environment. The design of a cellular manufacturing system must take many structural and operational issues into consideration [69]. One of the first and most important problems faced in practice is to select and group parts or machines with similar features into families; this process is called cell formation [19]. Cell formation is the first and the most important phase on GT application. This initial step influences all other decisions involved in the design of cellular manufacturing systems in which similar parts are aggregated into part families and dissimilar machines are grouped into cells [43]. The above concerns, among many others, make it important to work on a new method for cell formation that has the quality of portability into practice and real life applications.

1.2 Cellular Manufacturing

Cellular manufacturing is a manufacturing strategy to win a war against global competition by reducing manufacturing cost, improving quality, and reducing the delivery lead time of products in a high variety low demand environment [27]. Cellular manufacturing is an application of group technology in manufacturing in which all or a portion of a firm's machines or processes located in close proximity are dedicated to the manufacturing of a family part [57]. Cellular layouts organize departments around a product or a narrow range of similar products. An ideal cell manufactures a narrow range of highly similar products. Such an ideal cell is self-contained with all necessary equipment and resources. Materials sit in an initial

queue when they enter the cell. Once processing begins, they move directly from process to process and the result is very fast throughout.

Cellular Manufacturing is an approach that helps building a variety of products with as little waste as possible. A cell is a group of workstations, machine tools, or equipment arranged to create a smooth flow so families of parts can be processed progressively from one workstation to another without waiting for a batch to be completed or requiring additional handling between operations. Put simply, cellular manufacturing groups together machinery and a small team of staff, directed by a team leader, so all the work on a product or part can be accomplished in the same cell, eliminating resources that do not add value to the product [57]. Introduction of cellular manufacturing systems in manufacturing, results in reduced production lead time, work-in-process, labor, tooling, rework and scrap materials, setup time, delivery time, and paper works [36].

1.3 Benefits of Cellular Manufacturing

Cellular manufacturing offers substantial benefits to companies. These benefits include:

- Cellular manufacturing reduces material handling and transit times. By having the machinery to complete a certain process grouped together in a cell, the product spends more time on the machinery and less time in transit between machines. Because parts are moved within a cell rather than the entire factory, the travel time and distance are reduced, resulting in reducing the material handling cost, which constitutes between 20 and 50% of the total operating cost. By adopting cellular manufacturing, some companies reduced the cost of material handling by over 21% [69].

- Cellular manufacturing is based on the capability to produce families of similar products within each cell. Since similar parts are grouped, then it is possible that adjustments required to setup machinery would not be significant for each family product. The reported average reduction in setup times achieved by some companies rang between 35 to 40% [69].
- The decrease in setup time leads to an increase capacity of the machines as well as a decrease in work-in-progress inventories. Less Work in process (WIP) is easier to manage and allows the manufacturer to operate with shorter lead times.
- With decreased material handling and transit times, accompanied by the elimination of the queue times associated with batch processing, the time to produce one unit of a particular product becomes shortened, thus resulting in shorter delivery dates for the customer. Some companies achieved up to 24% reduction in manufacturing lead times [69].
- Cellular manufacturing systems, associated with one-piece flow, reduce work-in process inventories. With a continuous and balanced flow of product through the cell, no major build up of material occurs between workstations, therefore eliminating the need of excess space to store in-process goods. This also allows workstations and machinery to be moved closer together, so there will be considerable floor space available for adding machines and for expansion.
- Cellular manufacturing system (CMS) also has an impact on manufacturing and design engineering areas. The effect for manufacturing is a reduction of the number and variety of parts, so the process planning for the remaining parts is easier and more consistent. CMS

also paves the way for progression to computer integrated manufacturing. CMS assist in the economical justification or elimination of expensive NC machines in job shops.

- Communication is easy within each cell since every operator is close to the others. Ease of communication improves quality and coordination and the sense of common mission enhances teamwork in the cell.
- A cell on average employs a small number of workers that produce a complete part or product. Workers become multifunctional and are responsible for operating and maintaining numerous pieces of equipment and workstations. They are also able to cover other workstations within the cell when required to do so. In terms of worker productivity, the ability to deal with a product from start to finish creates a sense of responsibility and an increased feeling of teamwork. A common purpose and a sense of “ownership” is created among the production teams. Improving the production of the cell continuously and adjusting quality issues right away and not after an entire batch has been produced, greatly enhances the quality and efficiency of the production.

These results confirm the justification of applying the cellular manufacturing concept in batch manufacturing systems.

1.4 Design of Cellular Manufacturing Systems

Researchers in the past three decades have addressed various issues concerning design and operational control of Cellular Manufacturing Systems (CMS). These include cell formation, cell layout, operator allocation issues, short-term scheduling and performance evaluation.

During the past decade, there has been a tremendous interest in cell formation problem from practitioners as well as from academics. Cell formation, which involves identification of machine groups and part families, is the first step in designing a cellular manufacturing system. Cell formation is considered as a reorganization of an existing job shop into group technology shops using information given about the processing requirements of parts. This information is commonly represented in a matrix called the part-machine matrix with 0 or 1 entry. A 1 indicates that part P requires machine m for an operation, and 0 indicates otherwise [57]. As an example, Figure 1.1 shows the part-machine matrix for a small problem of seven machines and seven parts.

C O M P O N E N T S								
M		1	2	3	4	5	6	7
A		0	1	0	0	1	0	0
C		1	0	0	0	0	0	1
H		0	0	1	1	0	1	0
I		0	0	1	1	0	1	0
N		1	0	0	0	0	0	1
E		0	0	1	1	0	1	0
S		0	1	0	0	1	0	0
		1	2	3	4	5	6	7

Figure 1.1: Example of a machine-part matrix.

To form cells, the part-machine matrix is rearranged to a new matrix such that each part family is completely processed within a cell of machines and each part in a part family processed by every machine in the corresponding machine group. For example, the rearrangement of the matrix in Figure 1.1 is shown in Figure 1.2, where three different machine cells are indicated within blocks. Cell 1 consists of machines 2 and 5, cell 2 consists of machines 3, 4, and 6, while cell 3 consists of machines 1 and 7. Obviously, three part families are formed, parts 1 and 7 constitute the first family, parts 3, 4, and 6 constitute the second family, and the rest of the parts constitute the third family.

C O M P O N E N T S									
M A C H I N E S									
				3	4	6	2	5	
		1	1						
		1	1						
				1	1	1			
				1	1	1			
							1	1	
							1	1	

Figure 1.2: Example of an optimal clustering.

However, in real life the nature of data sets are such that a perfect decomposition is hardly ever obtained. In this situation the goal is to obtain a near perfect decomposition considering the following objectives while partitioning the matrix [47]:

- 1) To have minimum number of zeros inside the diagonal blocks (voids);
- 2) To have minimum number of ones outside the diagonal blocks (exceptional elements).

A void indicates that a machine assigned to a cell is not required for the processing of a part in the cell. When a part passes a machine without being processed on the machine, it contributes to an additional intra-cell handling cost. This leads to large, inefficient cells. An exceptional element is created when a part requires processing on a machine that is not available in the allocated cell of the part. When a part needs to visit a different cell for its processing the inter-cell handling cost increases. This also requires more coordinating effort between cells. Figure 1.3 shows an example of a matrix with voids and exceptional elements.

		C O M P O N E N T S							
M A C H I N E S		1	2	8	3	6	6	2	7
		1	1	1	1	1	1		
		1	1	1	1	0			
		1	1	0	1	1			
				1			0	1	1
						1	1	1	1
							1	1	1
		1	2	8	3	6	6	2	7

Figure 1.3: Example of real problem with voids and exceptional elements [57].

1.5 Research Objectives

A number of cell formation methods have been developed to this day. The goal of most of these methods is to obtain independent machine cells by minimizing inter-cell movement. Generally, however, the performance of many of the existing methods deteriorates as the problem under consideration becomes larger. Also determining the number of cells may not be straightforward. In some methods, the number of cells is a dependent variable, while in others it has to be identified in advance. Moreover, because of the unavailability of software programs supporting them, the industrial applications of the current methods are limited.

The main objective of this research is to develop a new mathematical approach for cell formation. The approach should:

- perform very well in terms of a number of well known criteria,
- compare favorably to well-known existing methods,
- have the flexibility in allowing the user either to identify the required number of cells in advance, or consider it as a dependent variable,
- be supported by software programs in order to facilitate industrial applications.

1.6 Research Assumptions

Below are the assumptions underlying the use of the proposed approach:

- The sequence of operations is ignored by the part-machine matrix and if a part requires more than one operation on a machine, this cannot be identified in this matrix.
- Only the machine types are referred to in the part-machine matrix, not the number of copies available of a given machine type.
- The machine type within the group to which the part is assigned has sufficient capacity to process the parts completely.
- The processing capacities of all the machines of a certain type are considered equal.
- A unique process plan exists for each product which specifies the sequence of machine types required for its production, as well as the corresponding set-up and processing times for each operation.
- Machines belonging to the same work-center type are assumed to be interchangeable; i.e., any member of the group can be utilized to process a part.

1.7 Outline of the Thesis

This thesis is organized in six chapters. Chapter 2 presents a literature review of relevant previous work done in the area of cell formation. Chapter 3 provides a general introduction of factor analysis. In chapter 4 the technical details associated with implementing the proposed approach is discussed and a detailed algorithm for the method is presented. Evaluating the performance of the proposed approach is the subject of chapter 5. This includes

implementation of the approach on a number of notable problems from the literature, and a comparison between the proposed approach with other well-known methods. Finally, the conclusion, the contributions, and directions for future works are presented in Chapter 6.

LITERATURE REVIEW

This chapter provides a survey of research to date including emerging work in the area of stochastic search techniques such as genetic algorithms and neural network approaches. Models are discussed in terms of assumptions, analytic methods, performance criteria, and limitations. Emphasis is given to reporting empirical results and comparative evaluations of techniques. The methodological classification of techniques shown in Figure 2.1 [27] is adopted to improve readability and to facilitate an understanding of the basic advantages or limits of generic approaches. The impact of considering alternative process plan and additional machine copies if available is also considered in these methods.

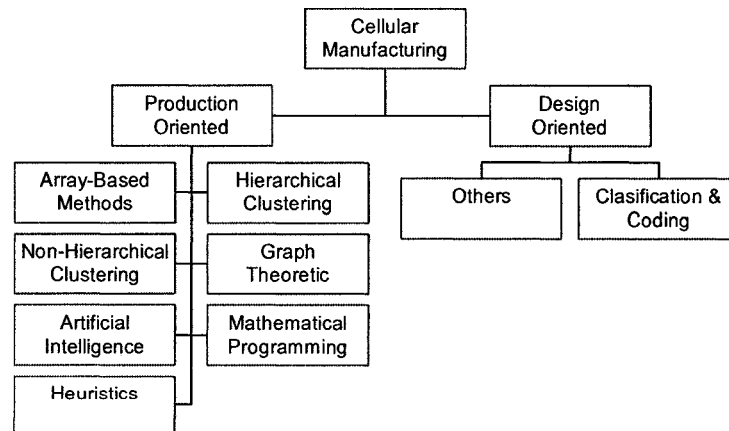


Figure 2.1: Categories of cell formation approaches.

2.1 Array-Based Clustering Algorithms

Array-based clustering is one of the simplest classes of production-oriented cell formation method. This class of algorithms utilizes the machine-part incidence matrix, each row of which represents a part and each column a machine. The elements of this matrix assume values of 0 or 1, a value of 1 for the entry a_{ij} representing an operation upon part i by machine j . Rows and columns are permuted to form a set of blocks with high densities of 1s along the diagonal (see Figures 1.1, 1.2). Any tightly clustered blocks represent the candidate part families and machine cells, which are formed simultaneously. Chandrasekharan and Rajagopalan [15] and Venugopal and Narendran [63] have done analysis of the 0-1 machine-part incidence matrix in order to exploit properties of this matrix for developing cell formation algorithms.

Methods which belong to this class of algorithms are [25]: (i) the rank order clustering approach of King [30], which rearranges the machine-part incidence matrix based on the 'binary rank orders' of its rows and columns; (ii) the direct clustering algorithm of Chan and Milner [12], which forms part and machine families by rearranging the rows and the columns of the incidence matrix, based on the number of non-zero elements in each; (iii) the method of McCormick et al. [46], which maximizes the total 'bond energy' of the machine-part incidence matrix; (iv) Rank order clustering 2 (ROC2) method developed by King and Nakornchai [31] and modified rank order clustering (MODROC) method developed by Chandrasekaran and Rajagopalan [14]. The common objective of all these methods is to maximize the number of operations performed on the part families within their corresponding machine cells. Here only three of them Rank order clustering (ROC), direct clustering analysis (DCA), and bond energy analysis (BEA), were selected for the comparison.

2.1.1 Rank Order Clustering (ROC)

ROC, as proposed by King, [29, 30] is a well-known clustering technique that attempts to create a block-diagonal form by repeatedly reallocating the columns and rows of a machine-part matrix according to binary values to reduce the computational effort of BEA. It provides a simple, effective and efficient analytical technique, which can be easily computerized. In addition, it has fast convergence and a low computational time. Each row (column) in the machine-part matrix is read as a binary word. The procedure converts these binary words for each row (columns) into decimal equivalents. The algorithm successively rearranges the rows (columns) iteratively in order of descending values until there is no change possible.

Although ROC is easy to apply, several limitations have been identified and explained by other researchers as well as by King himself. First, the quality of the results is strongly dependent on the initial disposition of the machine-part incidence matrix [12, 19, 68]. Therefore, identification of exceptional elements and bottleneck machines is somewhat arbitrary [14]. Second, the binary value that is used for the reallocation restricts the size of the problem that the technique can handle [19, 68]. If exceptional elements exist, the influence is much greater in the higher order bits, which can lead to a non-block form. A revised version, called ROC2, has been developed by King and Nakornchai [31] to overcome the size limitation and to increase computational efficiency.

2.1.2 Direct Clustering Algorithm (DCA)

DCA [12] was proposed to form tight groups along the diagonal of the machine-part matrix. It rearranges the matrix by moving the rows with the left-most positive cells to the top and the

columns with the top-most positive cells to the left where a positive cell has $a_{ij}=1$. After several iterations, all the positive cells will form diagonal blocks from the top left corner to the bottom right corner. Identical outcomes result from any initial starting matrix, unlike ROC because DCA initiates the procedure by counting the number of positive cells instead of depending on intuition. This method allows more flexibility in the size of the problem due to computer word length and converges in a relatively few iterations. This procedure again has a limitation because it may not necessarily always produce diagonal solutions. According to Wemmerlov, [68] the proposed algorithm may not produce viable or acceptable solutions because it redirects the diagonal with each iteration. A modified version by Wemmerlov, [68] removes this flaw and can reproduce the examples in the original paper by Chan and Milner [12]. Chu and Tsai[19] showed that even the modified version of DCA has difficulty producing natural diagonal blocks even when they exist in the input matrix. The modified version of DCA has been shown to perform poorly when applied to large, real-world data sets as it tends to form one small group in the northwest corner and then a very large, sparse group containing the rest of the machines and parts [68].

2.1.3 Bond Energy Algorithm (BEA)

The bond energy analysis algorithm was developed by McCormick, Schweitzer and White [46] to identify and exhibit the interrelations within each cell and associations among the clustered groups by means of total bond energy. A bond is claimed to exist between each pair of the neighboring rows and columns if they have positive cells in the machine-part matrix. These bonds create an energy, which is defined as the sum of products of adjoining elements. For a

particular row permutation (π) and column permutation (ρ), the total bond energy (TBE) is given by the following:

$$TBE(\pi, \rho) = \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N a_{ij} (a_{i,j+1} + a_{i,j-1} + a_{i+1,j} + a_{i-1,j}) \quad (2.1)$$

where

$$a_{i,0} = a_{i,n+1} = a_{0,j} = a_{m+1,j} = 0$$

M=the number of machines

N=the number of parts

BEA seeks to maximize the TBE over all $N! M!$ Permutations. BEA algorithm begins with an arbitrarily selected column (or row). It then places that column with the greatest contribution to the total bond energy beside the assigned column (or row). It repeats the same procedure for all the columns and rows. The method is applicable to problems of any size because the BEA has nothing to do with calculating the binary values. However, since the first step of the algorithm is determined by intuition, many possible solutions can be generated; that is, the solution depends on the initial row (or column) selected for starting the process.

2.1.4 Comparison of Array-Based Methods

Comparing each of the three array-based clustering techniques, BEA, ROC and DCA, Chu concluded that BEA significantly outperformed the other two in problems with and without exceptional elements and bottleneck machines [19]. The array-based clustering techniques used in the design of manufacturing cells are both efficient and simple to apply to the part-machine

matrix. However, these algorithms generally do not take into account other types of manufacturing data such as cost of machines and maximum cell size, and they usually require visual inspection of the output to determine the composition of the manufacturing cells.

2.2 Hierarchical Clustering Algorithms

Hierarchical clustering techniques operate on an input data set described in terms of a similarity or distance function and produce a hierarchy of clusters or partitions. At each similarity level in the hierarchy, there can be a different number of clusters with different numbers of members. Unlike the array-based techniques, hierarchical clustering methods do not form machine cells and part families simultaneously. These methods can be described as either divisive or agglomerative. Divisive algorithms start with all data (machines or parts) in a single group and create a series of partitions until each machine (part) is in a singleton cluster. Stanfel [59] is the only researcher to apply a divisive method to cellular manufacturing; therefore, attention is focused on agglomerative clustering algorithms that start with singleton clusters and proceed to merge them into larger partitions until a partition containing the whole set is obtained. Hierarchical clustering methods involve a two-stage process that first calculates similarity coefficients between each pair of individuals (machines or parts). This can be represented as a lower triangular matrix since the similarity coefficient between individuals is commutative. The second stage of the process determines how the pairs with roughly equivalent similarity levels should be merged. The specific logic for each individual method is described below [25]:

(i) The single linkage algorithm of McAuley [45], that defines the similarity between two machines in terms of the number of parts which visit both machines and the number of parts visiting either machine; he then aggregates machines with high similarity into manufacturing cells; (ii) the method proposed by Leskowsky et al. [40], which uses the average common part weighting metric to quantify similarities among parts; (iii) the acknowledged based group technology system of Kusiak [38], which forms work center cells and part families in such a way, that each part is manufactured exclusively by work centers in its corresponding cell or by bottleneck machines; (iv) the numerical taxonomy approach of Carrie [11], which is similar to the one proposed by McAuley; (v) the method of Wei and Kern [67], who, again, enumerate similarities between machines for use in their linear clustering algorithm.

2.2.1 Choice of Similarity Measure

In the hierarchical algorithms, the basis is to define similarity coefficients that incorporate manufacturing data other than just the binary part-machine incidence matrix [22, 53]. A variety of similarity measures have been defined. McAuley [45] used the generic Jaccard coefficient to form machine cells. Carrie, [11] who applied McAuley's work to several real problems, defined a similarity coefficient between pairs of parts to form part families first. There does not appear to be any inherent advantage to forming the part families or machine cells first [68]. Gupta and Seifoddini's [22] similarity coefficient incorporates production requirements, the machine-part incidence matrix, the actual sequence of operations, the average production volume for each part, and the unit processing time for each of the part's operations. Seifoddini and Djassemi [51] modified the Jaccard similarity to take into account production volume. When compared with the Jaccard similarity, the production volume based similarity reduces the sum of

intercellular and intracellular movements as well as improves the scheduling process. Mosier [52, 51] proposed the additive similarity coefficient (ASC), a weighted adaptation of the Jaccard coefficient that incorporates the relative importance of each part, and the multiplicative similarity coefficient (MSC), which is approximately a correlation coefficient.

In the following sections, single linkage clustering (SLC), average linkage clustering (ALC), complete linkage clustering (CLC), and linear cell clustering (LCC), the four most important similarity coefficient based clustering methods are discussed.

2.2.2 Single Linkage Clustering (SLC)

The first step in hierarchical clustering is to group the two individuals, i and j , with the highest level of similarity into one cluster, ij . The combined cluster behaves as if it is a single individual. The similarity between this cluster and individual k , as defined by the SLC [45] algorithm, is the maximum of the similarities between k and the component members of the cluster ij . Iterations continue to merge the groups with the largest similarity coefficient until a single group exists. The most common way to display the hierarchy of clusters generated by the algorithm is in the form of a dendogram. The cell designer must choose a similarity level or threshold in order to define the number of clusters. As the threshold increases, the number of cells increases while the size of each cell decreases. Seifoddini and Wolfe [54] selected a threshold that produces the minimum total material handling cost (intercellular plus intracellular). Hierarchical clustering algorithms do not cluster machines and parts simultaneously, so initially only cells or families are formed. The final step is to reapply

hierarchical clustering or a secondary procedure, such as ROC, to allocate parts (machines) to the families (cells).

The major drawback of SLC is the “chaining” problem, which may be caused by two clusters joining together. The two machine cells may join together just because two of their members are similar, but the other members may remain far apart in terms of similarity. In other word two clusters can be grouped based merely upon a single bond between one machine in each cluster [22, 45, 53, 50]. The chaining problem can lead to improper machine assignment in the groups [55]. The main advantage of SLC is its simplicity and less computational requirement.

2.2.3 Average Linkage Clustering (ALC)

To help reduce the chaining problem, Seifoddini and Wolfe [50] applied the average linkage clustering (ALC) algorithm. The similarity between two clusters is defined as the average of the similarity coefficients for all of the members of the two clusters. When clusters ν and t are merged, the sum of pairwise similarity between the two clusters is:

$$S_{tv} = \frac{\sum_{m \in t} \sum_{n \in \nu} S_{ij}}{N_t \times N_\nu} \quad (2.2)$$

where the double summation is the sum of pairwise similarity between all machines of the two groups, and N_t , N_ν are the number of machines in groups t and ν respectively.

Since in ALC two machine cells join together based on the overall similarity coefficient between all their members, it is unlikely that two similar members in two machine cells cause the machine cells to join together but the remaining members may not be similar enough. ALC usually applies the same rules as SLC except that anytime the similarity coefficient matrix should be revised. In the revised similarity coefficient matrix, the similarity coefficient between the newly formed machine cell and the remaining cells should be recalculated. The major drawback of ALC is its computational requirement that is significantly higher than in SLC due to revision of the similarity coefficient matrix anytime a new machine cell is formed [27].

2.2.4 Complete Linkage Clustering (CLC)

Complete linkage clustering (CLC) further reduces the chaining problem by selecting the minimum similarity coefficient as the in-between cluster relationship instead of the maximum [22]. SLC, ALC, and CLC algorithms can deal with both similarity coefficients as well as Euclidean distances in which clusters are merged by selecting the minimum distance between clusters instead of the maximum similarity.

2.2.5 Linear Cell Clustering (LCC)

One of the limitations of SLC and CLC is that the similarity coefficient used in these methods does not give importance to the parts that do not need processing by the machine pairs. The LCC method overcomes this problem. It clusters machines based on the use of a commonality score that defines the similarity between two machines. The commonality score not only considers the parts that require both machines for processing, but also the parts, which do not require both machines. The commonality score is presented as below:

$$C_{mn} = \sum_{p=1}^P \delta(a_{pm}, a_{pn}) \quad (2.3)$$

where:

$$\delta(a_{pm}, a_{pn}) = \begin{cases} (P-1), & \text{if } a_{pm} = a_{pn} = 1 \\ 1, & \text{if } a_{pm} = a_{pn} = 0 \\ 0, & \text{if } a_{pm} \neq a_{pn} \end{cases}$$

p, q are indexes for parts and m, n are indexes for machines.

2.2.6 Comparison of Hierarchical Algorithms

Hierarchical clustering methods can be implemented easily and have an advantage relative to array-based clustering, i.e., they have the flexibility to incorporate manufacturing data other than the binary machine-part incidence matrix [22, 53]. One disadvantage is that the designer must decide on an appropriate similarity level to select the groups. In small applications, this is not a problem since the designer can visually evaluate the dendrogram. However, as applications become too large for output in the form of a dendrogram, other means of storing the hierarchy must be employed, such as minimum spanning trees [45]. The duplication of bottleneck machines is not handled by most algorithms, although Seifoddini and Wolf [53] employed a strategy for this problem. Shafer [55], and Vakharia and Wemmerlov [61] conducted an in-depth comparison of many different hierarchical clustering algorithms with different similarity and distance coefficients. Seifoddini and Hsu [52] showed that the weighted similarity coefficient produces better solutions based on the number of exceptional elements than the Jacard similarity score. They show that grouping efficiency, and the grouping capability indices were not consistent performance evaluation measures [27].

2.3 Methods Based on Artificial Intelligence

Researchers have increasingly applied artificial intelligence (AI) techniques to the cellular manufacturing problem. Many of these methods use solution methodologies patterned after non-hierarchical clustering methods, array-based clustering methods, etc. However, their AI implementation offers advantages over traditional cell formation methods.

2.3.1 Artificial Neural Networks

Artificial neural networks have been applied successfully to many manufacturing areas. Several researchers have applied a supervised learning approach to the classification and coding problem based on the back-propagation learning algorithm [27]. This method can be also applied to a production- oriented method to determine the machine cells and part families. Unsupervised learning techniques are better suited for the general clustering problem. It is not necessary to specify a priori the number of clusters or the representative members of these clusters. Once the part families and machine cells are determined, a supervised model can be trained to assign new parts to the existing cells. Malave and Ramchandran [44] applied a modified version of the Hebbian learning rule to the cell formation problem, while others have applied other unsupervised neural learning algorithms such as competitive learning and Kohonen nets [62]. Several researchers used the neural network classifier based on an unsupervised learning model by Carpenter-Grossberg [10] called adaptive resonance theory (ART1) and its variants [62]. Unsupervised learning techniques such as ART1 cluster the input vectors into separate groups based upon similarities. The artificial neural network technique executed quickly and obtained good clusters. The real advantage is its ability to solve large data sets (10,000 parts and 100 machine types). ART and its variants can be classified as non-

hierarchical clustering methods [27]. Another variant of the ART models, Fuzzy-ART, handles both analogue and binary-valued inputs while utilizing a new learning law. Burke and Kamal [9] compared Fuzzy-ART with ART1, DCA [12], Hebbian Learning [44], and a procedure by Ballakur and Steudel [22] and concluded that Fuzzy-ART was a viable algorithm that outperformed all the other algorithms. However, this comparison was based on very small data sets and did not test the robustness of each algorithm. Fuzzy-ART produced superior solutions in terms of the bond energy recovery ratio (BERR), where BERR is the ratio of the final bond energy to the initial bond energy.

2.3.2 Fuzzy Logic

Most clustering methods assume “that part families are mutually exclusive and collectively exhaustive [18]. While some parts definitely belong to certain part families, it is not always clear which family is appropriate [18, 41, 70]. Li and Ding [41] and Xu and Wang [70] applied fuzzy mathematics to this problem. Chu and Hayya [18] applied a fuzzy c means clustering algorithm to production data. The fuzzy c -means clustering can be classified as a non-hierarchical method and suffers from the same problems associated with those methods. The number of part families, c , must be specified a priori. The authors stated that if c is underestimated, the result is far from optimal. Also, a poor stopping criterion leads to inferior clusters. However, the technique is unaffected by exceptional elements. The workload among machine cells can be balanced better by using a reallocation scheme that utilizes the degree of membership a part has in a particular family. Chu and Hayya compared the fuzzy approach to the optimal 0-1 integer-programming model and a heuristic approach. The fuzzy approach was clearly better than the integer programming (IP) approach in both execution time and the quality of the

solution. It was not as efficient as the heuristic but provided more information than is available from a “crisp” definition of families and cells.

2.3.3 Genetic Algorithms and Simulated Annealing

Genetic algorithms and simulated annealing are very efficient stochastic search algorithms that try to emulate natural phenomena. These algorithms have been used successfully to solve a wide range of optimization problems, especially combinatorial problems. Because of the NP-completeness of the grouping problem and existence of local minima, these stochastic search algorithms [34, 65] offer promising solution techniques for large-scale problems. Boctor [5] and Venugopal et al. [64] used simulated annealing to solve integer programming formulations of the cell formation problem. Genetic algorithms (GAs) have proven to be an effective and flexible optimization tool that can produce optimal or near-optimal solutions. Joines et al. [26] developed a genetic algorithm approach to solve integer-programming formulations of the cell design problem, allowing multi-criteria objective functions and constraints on the number of permissible cells. Venugopal et al. [65] also used GAs to solve a multi-objective integer programming formulation of the cell formation problem. These stochastic search techniques offer capabilities (missing in many of the more traditional methods) that can provide the basis for more practically useful cell formation algorithms. GAs do not make strong assumptions about the form of the objective function as do many other optimization techniques. Also, the objective function is independent of the algorithm, i.e., the stochastic decision rules. The only objective function requirement is that it maps the solutions into a partially ordered set. This offers the flexibility to interchange various objective functions and to utilize multi-criteria objective functions. Convenient substitution of various evaluation functions allows the system

designer to generate and review alternative cell designs quickly. Single-criteria objective functions limit a method's usefulness to that of assisting the cell designer rather than autonomously forming the system. To move toward a satisfactory algorithmic result, multiple criteria objective functions that include such things as setup time requirements, tooling and crewing requirements, alternative routings, cost of machines, inter-cell transfers, and reduced machine utilization are needed. GAs also offer the ability to constrain the number of permissible cells or part families selectively. Most clustering algorithms cannot identify all naturally occurring clusters and find solutions with a constrained number of clusters. The cell designer, at least initially, might specify an unconstrained problem to identify the naturally occurring groups of parts and/or machines. Afterwards, practical limits on the number of cells arising from availability of floor space, maximum work team sizes, or excessive machine redundancy requirements can be imposed. The ability to analyze the ordering of operations within routing sequences is important not just for material flow considerations, but also because cell throughputs are dependent upon setup times, which are usually sequence dependent. Joines [26] and Daskin [20] developed non- classical, array-based clustering techniques using order-based genetic algorithms. Order based GAs have the potential for analyzing operation precedence relationships to further refine the cell design process. Industrial data sets are often too large for visual methods to associate machine cells and part families effectively. GAs can form machine cells and part families simultaneously and avoid visual inspection of the data. Further exploitation of genetic algorithm capabilities makes practical solutions to industrial scale problems more realistic.

2.4 Mathematical Programming Methods

Purcheck [49] was among the first to apply linear programming techniques to the GT problem. As an optimization technique, the objective in cluster analysis is to maximize the total sum of similarities between each pair of individuals (machines or parts) or to minimize the distances between each pair. As stated by Kusiak, [37] the distance between any pair can be any symmetric function. The Minkowski, the weighted Minkowski, and the Hamming distance measures are the most often used in connection with cell formation [35, 37]. Models developed with distance-based objective functions can easily be extended to similarities. Mathematical programming approaches for the clustering problem are nonlinear or linear integer programming problems [35, 37]. These approaches offer the distinct advantage of being able to incorporate ordered sequences of operations, alternative process plans, non-consecutive part operations on the same machine, setup and processing times, the use of multiple identical as well as outsourcing of parts. These formulations also suffer from three critical limitations. First, because of the resulting nonlinear form of the objective function, most approaches do not concurrently group machines into cells and parts into families. Second, the number of machine cells must be specified a priori, affecting the grouping process and potentially obscuring natural cell formations in the data. Third, since the variables are constrained to integer values, most of these models are computationally intractable for realistically sized problems [39]. Large scale problems typically require heuristic and approximate methods with Lagrangean relaxation and subgradient optimization having been proposed, as well as a variety of simulated annealing and genetic algorithm approaches. This section presents some mathematical models that can be used for part family formation and machine grouping.

2.4.1 P-Median Model

Kusiak [35,37] proposed the p-median model to identify part families. This was the first model to form part families using mathematical programming. This mathematical model remains the same for cell formation; except for this case we consider the maximization for similarity instead of minimization of distance. The p-median model is used to cluster n parts (machines) into p part families (machine cells). Constraints specify that each part can belong to only one part family and the required number of part families is p . A part can be only assigned to a part family that has been formed. Similarity between two parts is defined as the number of machines the two parts have in common. This procedure identifies only the part families, and an additional procedure is needed to identify the machine groups.

2.4.2 Assignment Model

Srinivasan et al. [58] proposed an assignment model for the part families and machine grouping problem. They provided a sequential procedure to identify machine groups followed by identification of part families. The objective of assignment model is to maximize the similarity. This approach is reported to be superior both in terms of quality of solution and computational time in comparison with the p-median model.

2.4.3 Quadratic Programming Model

The clustering algorithms and p-median model minimize the distance or maximize the similarity between parts by considering the family group mean or median. However parts within the family interact with each other. Therefore, it becomes important to account for the

total family group interaction. Kusiak et al. [34] proposed a quadratic programming model for this purpose. They proposed solving this model by an eigenvector-based algorithm.

2.4.4 Assignment Allocation Algorithm (AAA)- Nonlinear Model

In a manufacturing situation, for different machine-part combinations the associated costs of voids and exceptional elements may vary and in general are not the same. For example there might be a special machine that all the parts requiring processing on this machine should be placed in a small cell [8]. This can be achieved if a high weighting value is given to the exceptional elements corresponding to this machine for all parts, while identifying the groups. This shows that there is a need to consider the importance of voids and exceptional elements explicitly.

Adil et al. [1] proposed a nonlinear mathematical model to identify part families and machine groups simultaneously without manual intervention. The objective of the model is to explicitly minimize the weighted sum of exceptional elements and voids. These parts can be considered to have potential for subcontracting or developing alternative process plans before allocating them to cells. By changing weights for voids and exceptional elements the user has the flexibility to form large loose cells, or small tight cells to suit the situation. The results obtained using the AAA compare favorably with well-known algorithms in the literature. The AAA is simple and less computer intensive.

2.5 Other Heuristics Methods

The cell design process is relatively complex and often proceeds in stages. As stated earlier, the algorithms for cell formation provide the first rough-cut groups. The exceptional elements and all groups can be individually considered in a more detailed analysis that includes other manufacturing aspects such as part sequence, processing times, machine capacities and the trade-off between the purchase of additional machines and material handling in order to make groups independent. Other than the mathematical programming techniques, most cell formation methods are heuristics. However, those discussed so far have been placed in aggregate categories, e.g., array-based clustering, artificial intelligence techniques, etc., based on their general solution approach. This section explains an additional diverse set of heuristics.

This class of cell-formation methods includes: (i) the cost-based heuristic of Askin and Subramanian [2]. This procedure can be classified as a similarity coefficient-based method that takes into account the fixed machine costs, material handling costs, WIP inventory costs, production cycle inventory costs, variable production costs and setup costs; (ii) the inter class traffic minimization method [3], which forms machine cells and the corresponding part families in order to minimize the total inter-cell traffic of parts within the shop; (iii) identification, clustering, refinement, merging and allocation heuristic(ICRMA) of Tabucanon and Ojha [60], which also aims at reducing the inter-cell material flow in the system through four stages of machine formation and subsequent solution refinement; (iv) a versatile inter-cell flow reduction heuristic, developed by Okogbaa et al. [21], which produces different alternatives for different designer input, e.g., number of cells, cell size restrictions, etc.:(v) technique developed by Minis et al. [48] that groups production machines into cells and parts

into families by minimizing the inter-cell traffic subject to capacity constraints. This method has the capability of including unique, as well as multiple-function, identical machines in the grouping procedure; (vi) A few procedures that work on further improving the solution obtained by cell formation algorithms are by Kern and Wei [28], Logendran, and Shafer[42]. These procedures assume that the option to change the parts processing plans to suit the cell has already been considered. Only a few researchers have addressed the importance of considering alternative process plans during cell formation [35].

2.6 Summary

This chapter has given an overview of the methods of cell formation and important factors involved. It can be concluded from the overview that these methods suffer from one or more drawbacks. Their major common drawbacks include the deterioration of performance as the problem under consideration becomes larger, the inflexibility in determining the number of cells, i.e., in some methods, the number of cells is a dependent variable, while in others it has to be identified in advance, and the limited industrial application due to the unavailability of software programs supporting them. Therefore, new cell formation approaches that overcome these limitations are clearly needed. In response to this need, this thesis presents a new mathematical approach for the manufacturing cell formation, which is mainly based on factor analysis. Providing a background on this technique is the subject of the following chapter.

FACTOR ANALYSIS

Since cell formation can be considered as a dimension reduction problem in which a large number of interrelated machines are grouped into a smaller set of independent cells, the proposed approach applies factor analysis, a dimension reduction technique, to the similarity coefficient matrix to form machine cells. The objective of this chapter is to provide the reader with an introduction to this technique.

3.1 Introduction

Factor analysis, is a powerful tool of multivariate analysis. This method seeks to discover if a complex set of observed variables could be explained in terms of a smaller number of variables called factors. In many experimental setups, variables cannot be chosen as independent phenomenon, and have covarying influences. Correlations among variables may result from a situation where distinct groups or clusters of variables possess a common property not shared by the remaining set, for example they may be measuring the same driving principle behind the behavior of the system, which is not measured by the rest of the variables. In many systems there are only a few such driving forces. When this happens, we can take advantage of this redundancy of information and simplify our problem by replacing a group of correlated variables with a single new variable. Factor analysis attempts to answer what is the most elementary question in the multivariate analysis: how can we explain the systematic behavior of the observed variables by means of a smaller set of computed variables? In other words the

objective of a factor analysis model is to identify the underlying structure of a system when we suspect that the observed variables depend on a “smaller number of unobserved variables” which account for the true variance/covariance structure of the data set. This will result in a clustering of the original variables to uncorrelated groups, and thus a dimension reduction of the original data set.

In order to better visualize how can factor analysis replace the correlated variables with a new variable one can bring the example of finding the regression line in the scatter-plot of two variables. A regression line can be fitted to represent the best summary of the linear relationship between the variables. If we could define a variable that would approximate the regression line in such a plot, then the variable would capture most of the “essence” of the two items. In a sense we have reduced the two variables into one factor. Note that the new variable is a linear combination of the two old variables. This example described above, combining two correlated variables into one factor, illustrates the basic idea of factor analysis. If we extend the two-variable example to multiple variables, then the computations become more involved, but the basic principle of expressing two or more variables by a single factor remains the same. When there are more than two variables, we can think of them as defining a “space”, just as two variables define a plane, and the generalization of the curve fitting will be a hyper-plane fitting through the data points.

The most straightforward and quantitatively rigorous method that seeks to achieve this objective is the principal component analysis (PCA). This method generates a new set of variables, called principal components (corresponding to factors in factor analysis). Each principal component is a linear combination of the original variables. In this method all the

principal components are generated in a way that they are orthogonal to each other so the correlation between them is zero. The principal components, as a whole, form an orthogonal basis for the space of the data points. But what makes the principal components basis special is not only the orthogonality, but what is special about them is that the extraction of the principal components amounts to a variance maximizing rotation of the original variable space. What does this mean?

The first principal component is a single axis in space. When you project the data points on this new axis, a new variable is generated. This variable is generated in a way that its variance is the maximum among all possible choices of the first axis. The second principal component is another axis in space perpendicular to the first. Projecting the data points on this axis generates another new variable, whose variance is the maximum among all the choices for the second axis. The full set of the principal components is as large as the original set of variables, but usually the variances of the first few principal components exceed 80% of the total variance of the original data. This property means that the data points can be rigorously separated into distinct clusters when projected into a space spanned by the first few principal components. This also results in the dimensionality reduction objective of factor analysis.

In this chapter we will begin by giving the mathematical background of PCA method and then we proceed to describing the basic concepts of factor analysis used in next chapters. This discussion is followed by the presentation of the rotation method and the famous algorithms used for this purpose are introduced. The references used for the mathematical theorems of PCA and their proofs are the following [4,32].

3.2 Principal Component Analysis

From a purely mathematical viewpoint the purpose of a principal component (PC) model is to transform p correlated random variables to an uncorrelated orthogonal set of variables. A number of methods exist which can be used to orthogonalize a set of random variables. The PCA method uses the eigenvector, eigenvalue analysis of the covariance or correlation matrix of the variables to achieve this aim. Let X_1, X_2, \dots, X_p represent a set of random variables distributed according to some multivariate probability function with zero mean. And assume that we have p sample points of this p -dimensional probability function, so that $X = (X_1, X_2, \dots, X_p)^T$ can be seen as a $(p \times p)$ matrix, where each column corresponds to the coordinates of data points in our p dimensional data space, and each row can be seen as a set of continuous random variables of zero mean. The covariance matrix of this set is formed as the expectation: $E(X X^T) = \Sigma$, where each element c_{ij} shows the covariance between the random variables of set X_i and X_j . This matrix is a symmetric matrix. If the matrix Σ is diagonal, the sets of random variables X_i are already orthogonal and uncorrelated and they represent an orthogonal basis for illustrating the clusters of the data points. As the covariance between the random variables X_1, X_2, \dots, X_p increase, they contain more redundant information, and can be replaced by a smaller number of new orthogonal set of variables (factors), which may be regarded as the genuine explanatory variables of the data set. As we said before the objective of the principal component analysis is to compute these new variables as linear combinations of the old random variables in a way, which also maximizes the variances of these new random variables. So now let's form the linear combination $\xi = \Pi^T X$, where:

$$\Pi_{p \times p}^T = \begin{bmatrix} \pi_{11} & \pi_{21} & \cdots & \pi_{p1} \\ \pi_{12} & \pi_{22} & \cdots & \pi_{p2} \\ \vdots & \vdots & \vdots & \vdots \\ \pi_{1p} & \pi_{2p} & \cdots & \pi_{pp} \end{bmatrix} \quad (3.1)$$

is a $(p \times p)$ matrix of fixed non-random coefficients and $\xi = (\xi_1, \xi_2, \dots, \xi_p)^T$ are the new random variables formed as follows:

$$\xi_1 = \pi_{11} X_1 + \pi_{21} X_2 + \cdots + \pi_{p1} X_p = \Pi_1^T X$$

$$\xi_2 = \pi_{12} X_1 + \pi_{22} X_2 + \cdots + \pi_{p2} X_p = \Pi_2^T X$$

$$\vdots$$

$$\xi_p = \pi_{1p} X_1 + \pi_{2p} X_2 + \cdots + \pi_{pp} X_p = \Pi_p^T X \quad (3.2)$$

The new random variables ξ_i are known as principal components (factors) and the coefficients π_{ij} are known as the loadings. Now the problem is how to calculate the coefficient π_{ij} so that ξ_i are orthogonal, and also the variances of ξ_i are maximized. In order to do so lets first compute the variance and the covariance of ξ_i

$$\text{var}(\xi_i) = E(\xi_i^2) = E\left[\left(\Pi_i^T X\right)\left(\Pi_i^T X\right)^T\right] = \Pi_i^T E(X X^T) \Pi_i = \Pi_i^T \Sigma \Pi_i$$

$$\text{for } i = 1, 2, \dots, p \quad (3.3)$$

$$\text{cov}(\xi_i, \xi_j) = E(\xi_i, \xi_j^T) = E\left[\left(\Pi_i^T X\right)\left(\Pi_j^T X\right)^T\right] = \Pi_i^T E(X X^T) \Pi_j = \Pi_i^T \Sigma \Pi_j$$

$$\text{for } i \neq j \quad (3.4)$$

In order to maximize the variance of ξ_i we have to maximize the quadratic form $\Pi_i^T \Sigma \Pi_i$. We can now formulate the problem as maximizing the following Lagrangian expression

$$\begin{aligned} \phi_i &= \Pi_i^T \Sigma \Pi_i - \lambda_i (\Pi_i^T \Pi_i - 1) \\ i &= 1, 2, \dots, p \end{aligned} \quad (3.5)$$

with respect to Π_i , with the constraint of $\Pi_i^T \Pi_i = 1$ for $(i = 1, 2, \dots, p)$ (i.e. standardizing the eigenvectors to unit length). For solving the maximization problem we set the partial derivatives to zero

$$\frac{\partial \phi_i}{\partial \Pi_i} = 2 \Sigma \Pi_i - 2 \lambda_i \Pi_i = 0 \quad (3.6)$$

which reduces to

$$\begin{aligned} (\Sigma - \lambda_i I) \Pi_i &= 0 \\ i &= 1, 2, \dots, p \end{aligned} \quad (3.7)$$

Equation (3.7) is an eigenvalue, eigenvector equation where the terms $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ are the real, non-negative roots of the determinant polynomial of degree p

$$|\Sigma - \lambda_i I| = 0 \quad (3.8)$$

where I is the identity matrix. The equation is solved for parameter λ_i , and then Π_i can be calculated using the values of λ_i in equation (3.7). Therefore the objective function (3.5) is maximized by the eigenvectors Π_i , which correspond to the eigenvalues λ_i . Furthermore, it is also known that the eigenvalue-eigenvector solution of symmetric matrices, has an additional property according to which any two eigenvectors Π_i and Π_j which correspond to eigenvalues λ_i and λ_j respectively are orthogonal, that is $\Pi_i^T \Pi_j = 0$, and therefore $\text{cov}(\xi_i, \xi_j) = \Pi_i^T \Sigma \Pi_j = 0$. The variance of each factor is

$$\text{var}(\xi_i) = \Pi_i^T \Sigma \Pi_i = \lambda_i \quad (3.9)$$

The result of this discussion is that when $\xi_i = \Pi_i^T X$, and $\xi_j = \Pi_j^T X$ are any two linear combinations of p random variables X , such that Π_i and Π_j are the eigenvectors of $E(X X^T) = \Sigma$ (the covariance matrix) then ξ_i and ξ_j are uncorrelated random variables. And given a set of random variables X_1, X_2, \dots, X_p , the set of variables $\xi_1, \xi_2, \dots, \xi_p$ which are produced as the linear combinations $\xi_i = \Pi_i^T X$, with the eigenvectors of the covariance matrix Σ as the coefficients of the equation, are orthogonal and have maximum variance.

Ranking the roots in the decreasing order of explanatory variance or power, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$, maximizes the quadratic forms $\Pi_i^T \Sigma \Pi_i$ in a sequential manner so that the number of the orthogonal components that account for a given percentage of variance can be kept to a minimum. The expansion (3.2) may be terminated at any stage $1 \leq r \leq p$ if it is felt that a satisfactory percentage of the variance of the original variables has been accounted for by the $\xi_{i(i=1,2,\dots,r)}$, for some $1 \leq r \leq p$. Since each PC maximizes the variance, we can always retain the number of $r \leq p$ which accounts for some predetermined minimal percentage of the total variance. Alternatively all PCs can be computed and the last $p-r$ omitted if they account for an insignificant proportion of variance. Both approaches result in identical coefficients Π_i . The higher the correlation among the values of X_i the smaller is the number of PCs required to account for a fixed percentage of variance. Once the eigenvectors are known, equation (3.2) can be inverted to express the random variables in terms of the PCs.

$$\begin{aligned}
X_1 &= \pi_{11}\xi_1 + \pi_{12}\xi_2 + \dots + \pi_{1p}\xi_p = \Pi_1^T \xi \\
X_2 &= \pi_{21}\xi_1 + \pi_{22}\xi_2 + \dots + \pi_{2p}\xi_p = \Pi_2^T \xi \\
&\vdots \\
X_p &= \pi_{p1}\xi_1 + \pi_{p2}\xi_2 + \dots + \pi_{pp}\xi_p = \Pi_p^T \xi
\end{aligned} \tag{3.10}$$

where Π_i is the i th row vector of Π in matrix form $X = \Pi \xi$.

Now let Λ and Π be the eigenvalues and the eigenvectors of Σ .

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{bmatrix} \quad (3.11)$$

Then covariance between X_i and ξ_j are given by the i th and j th elements of $\Pi\Lambda$, which can be written as:

$$\begin{aligned} \text{cov}(X_i, \xi_j) &= E(X_i \xi_j^T) = E(\Pi_i^T \xi \xi_j^T) = \Pi_i^T E(\xi \xi_j^T) = \lambda_j \pi_{ij} \\ (i, j &= 1, 2, \dots, p) \end{aligned} \quad (3.12)$$

We have assumed un-standardized ξ to this point. In order to standardize ξ we divide it by its standard deviation (the square root of the variance). Standardizing the new variables we obtain

$$\xi_j^* = \lambda_j^{-\frac{1}{2}} \xi_j \quad (3.13)$$

so that ξ^* becomes orthonormal (orthogonal and of unit length). Then the covariance between X and ξ^* are given by

$$\text{cov}(X_i, \xi_j^*) = \lambda_j^{\frac{1}{2}} \pi_{ij} \quad (3.14)$$

The coefficients between X_i and ξ_j in equation (3.10), which is referred to as the *covariance loading* indicates how strongly the variable X_i could be reflected by factor ξ_j . When a normalized covariance matrix (correlation matrix) is used, the covariance loadings are called *correlation loadings*. The loadings can be of aid when deciding which components are to be retained and which to be deleted.

3.3 Rotation

Factor analysis as introduced previously is a multivariable method that has as its aim the explanation of relationships among several difficult to interpret correlated variables in terms of a few conceptually meaningful, relatively independent new variables which in the factor analysis literature are called the factors.

In the past section we provided the mathematical basis of the computation of these new factors (principal components) using the principal components method. In this section we intend to define the main steps involved in factor analysis based on the previous discussion. The factor analysis method is generally composed of three steps: The first step involves setting up the data for the input, which includes the formation of the correlation matrix for the original variables. The second step involves using the correlation matrix to determine a set of initial factors. This is usually accomplished by the method of principal components, which we have described in the previous section. Derivation of the factors allows us to represent the information contained in the original variables in terms of a smaller number of new variables, which are statistically independent. The factor loadings describe the correlations between the factors emerging from the factor analysis and the original variables used in the construction of

the factors. The primary use of such a matrix, where the numbers in each column are the correlations of a specific factor with the original variables, is to pinpoint those variables that are highly correlated (i.e. load high) with a given factor, so that the factor can be conceptually interpreted. The third step involves some further manipulation of the original factors in order to make them easier to interpret. This involves rotation to achieve conceptual meaningfulness. The concept of rotation is discussed in next section.

In statistical data analysis the location of a set of coordinate axes is arbitrary, in the sense that the variances, angles, and distances of the data points are invariant with respect to position of the axes. The $r \leq p$ dimensional subspace of the ordinary principal components model is chosen so as to satisfy two general constraints, the orthogonality of axes and the stepwise optimization of variance. When searching for clusters of variables or searching for the best interpretable factors, however, different constraints are usually required. Once the optimal “correct” r -dimensional subspace is found we may wish to introduce an additional condition, namely, that each factor be maximally correlated with a single subset or cluster of random variables. This permits a straightforward identification of the factors in terms of the clusters, if such clusters exist. This generally implies the elimination, from a principal component solution, the initial conditions of variance maximization and component orthogonality since a second linear transformation is applied to the principal components. Such secondary transformations or rotations can be orthogonal or more generally oblique.

The reason why usually the original principal components, before rotation, are not useful in scientific work is that, the principal component method is designed to extract approximately as much variance as possible with the extraction of each successive component,

resulting in a sharp drop off in variance from the first factor to the last factor. The unrotated factors tend to be highly complex factor constructs that relate to or overlap with many of the variables rather than just a few. The first factor for example can have appreciable loadings for almost all variables. Such complex, overlapping factors are usually difficult to interpret because they contain within them many unrelated parts. For this reason the factor matrix is rotated to another form, which is mathematically equivalent to the original unrotated matrix but which represents simpler and much more useful factor constructs. Following the computation of the correlations, extraction of the unrotated factors, and the rotation of the unrotated factors, one can pick out the variables in each rotated factor that have high loadings.

Translation of a factor with a “simple structure” to quantitative terms would be a factor structure in which each of the original variables relates highly to only one factor and each factor can be identified as representing what is common to a relatively small number of variables. Thus, a simple factor structure is achieved when, for each factor, the factor loadings for most variables are zero and the remaining factor loadings are relatively large. In such a case, the factor can be conceived as describing the variation shared in common by the subset of variables highly related to it and not describing the variation in the other variables. Figure (3.1) illustrates geometrically how rotation conceives simple structure.

In order to understand the geometrical description better let's first consider the equations from the previous discussion:

$$X_1 = \alpha_{11}\xi_1 + \alpha_{12}\xi_2 + \cdots + \alpha_{1p}\xi_p$$

$$X_2 = \alpha_{21}\xi_1 + \alpha_{22}\xi_2 + \cdots + \alpha_{2p}\xi_p$$

$$\vdots$$

$$X_p = \alpha_{p1}\xi_1 + \alpha_{p2}\xi_2 + \cdots + \alpha_{pp}\xi_p \quad (3.15)$$

where α_{ij} are the correlation loadings, and the X_i are the original variables and the ξ_i are the principal components (factors) before rotation. For simplification let's consider the matrix form of the set of the equations above.

$$X = A\xi \quad (3.16)$$

where A is the matrix of the correlation loadings. Now we can look at this equation from a new point of view. If we consider the space spanned by the orthogonal axes $\xi_1, \xi_2, \dots, \xi_r$, we can illustrate the original variables X_1, X_2, \dots, X_p , with α_{ij} being the coordinate of X_i on axis ξ_j .

Figure (3.1) portrays an exemplary case of rotation of the principle axes for a two-dimensional case, i.e. there are two main principal components. In this figure there are as many dots as there are variables, and each dot correspond to a particular variable. As we said above, associated with each dot (variable) are two loadings, which are the coordinates of each variable on the principal components. If these axes are rotated then we shall have two new rotated factors. The objective is now to rotate the axis so that each dot is only close to one of the two rotated axes.

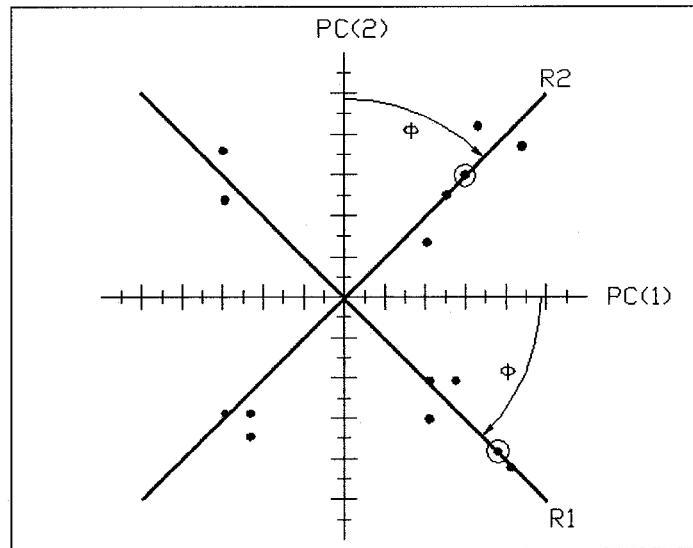


Figure 3.1: Illustration of the purpose of rotation: The goal here is to rotate the axes so that each dot is close to only one of the two rotated axes.

The conceptual accomplishment of the rotation is that now the variables can be seen as clustered into two subgroups, one subgroup lying close to one rotated axis, and the other subgroup closer to the other rotated axis. Now the new rotated axes (factor) can be interpreted in terms of the particular subgroup of variables lying close to that factor.

In performing a rotation two methods can be chosen. First, the new axes can still be kept orthogonal after rotation. This method is called orthogonal rotation. Second method is that each axis may be rotated independently, so they are not necessarily perpendicular to one another after rotation, this is called oblique rotation. An important statistical difference between orthogonal and oblique rotation is that factors resulting from the orthogonal rotation of the principal components will remain statistically uncorrelated, whereas factors resulting from an oblique rotation are usually correlated to some extent.

Another desirable property of orthogonal rotation over oblique rotation is that the amount of the total variation accounted for by the factors under consideration is unaffected by the rotation. But sometimes the goal of simple structure is better achieved by permitting the factor axes to become oblique. There are three algorithms for orthogonal rotation available in most factor analysis computer programs. They are called varimax, quartimax, and equimax methods. The essential difference between these three methods is that the varimax method attempts to achieve simple structure with respect to the columns of the factor loading matrix, quartimax attempts to achieve simple structure with respect to rows of the factor loading matrix, and the equimax attempts to achieve simple structure with respect to both rows and columns of the factor loading matrix.

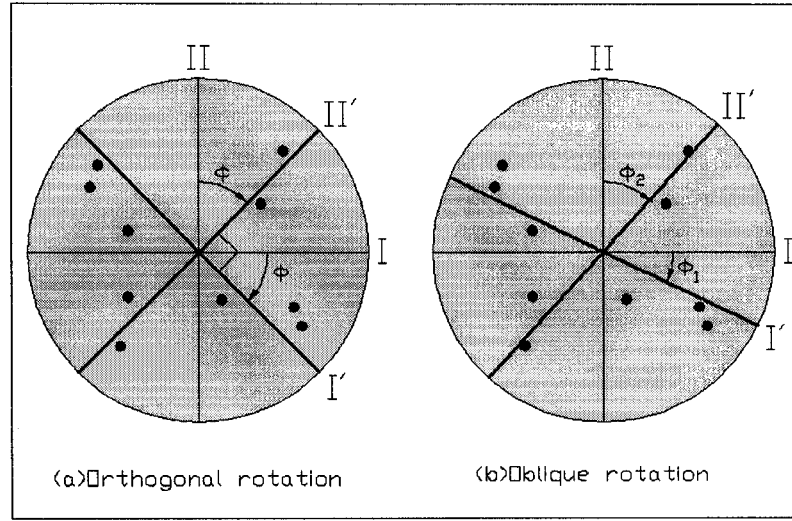


Figure 3.2: Orthogonal and oblique rotation.

The criterion that is used in the varimax method for finding the optimal position of the component axes is as the follows: Let B^T denote the $(r \times p)$ new (rotated) loading matrix with typical element b_{ij} , and consider the expression:

$$V_j^* = \frac{1}{p} \sum_{i=1}^p (b_{ij}^2)^2 - \frac{1}{p^2} \left(\sum_{i=1}^p b_{ij}^2 \right)^2 \quad (3.17)$$

for $j = 1, 2, \dots, r$. This equation represents the variance of the (squared) loadings b_{ij} on the j th principal component. Reminder: for an arbitrary random variable x the variance can be calculated as

$$\text{var}(x) = E[x - E(x)]^2 = E(x^2) - E(x)^2 \quad (3.18)$$

Squared loadings are used to avoid negative signs, but they represent contributions to the total variance explained by the j th component for the i th variable. The varimax algorithm finds the b_{ij} , which maximizes the sum

$$V^* = \sum_{j=1}^r V_j^* \quad (3.19)$$

which results in a pattern of elements of B^T where some are made as small as possible, and the others are made as large as possible. Actually the varimax criterion tries to obtain principal components with a high correlation for some variables or no correlation at all with others. For this reason it minimizes the number of principal components and is well suited for locating clusters that lie at right angles to each other. The varimax criterion seeks to maximize the variance of the loadings across the variables.

An older criterion is the equimax criterion, which seeks to maximize the variance across the principal components. Let

$$Q_i^* = \frac{1}{r} \sum_{j=1}^r (b_{ij}^2)^2 - \frac{1}{r^2} \left(\sum_{j=1}^r b_{ij}^2 \right)^2 \quad (3.20)$$

for $i = 1, 2, \dots, p$, where b_{ij} are the new loadings. Equation (3.20) represents the variance of the squared loadings, which in turn represents the contribution of the variables to variance. The quartimax algorithm maximizes

$$Q^* = \sum_{i=1}^r Q_i^* \quad (3.21)$$

the sum of variances of the rotated loadings. Since the quartimax criterion attempts to maximize the variance across the components, and concentrates the variance on the first component, it tends to produce a dominant component, which makes it undesirable for a cluster analysis problem.

A large number of algorithms have been developed to perform oblique rotation. Those most conveniently available are the oblimin, quartimin, biquartimin, and covarimin algorithms. All of these represent algorithms designed to satisfy various types of simple structure criteria. Unfortunately, not any one algorithm always gives the best solution, so a number of algorithms have to be tested to find which one works best for a given data set.

The results of rotation are usually quantified and evaluated through the consideration of the factor loading matrices. The goal of a simple structure is achieved when, comparing the factor loadings of initial factors versus rotated factors, the factor loading matrix of rotated factors contains high loadings on only a few variables for each factor, with close to zero loadings otherwise.

3.4 Summary

In this chapter, we reviewed the basic concepts involved in PCA and factor analysis methods. We also reviewed the concept of rotation. In the next chapter we will report on the application of the methods presented in this chapter for solving the problem of cell formation and the experiments designed for the implementation and evaluation of this method.

A PROPOSED MATHEMATICAL APPROACH FOR CELL FORMATION

4.1 Introduction

The approach presented in this chapter is based on factor analysis method and seeks to find out if the original set of variables (machines) can be grouped into distinct uncorrelated clusters (cells). For this purpose, the principal component analysis (PCA) method is used to first generate a set of orthogonal factors from the linear combination of the original variables and then to project the original variables into a new space spanned by these factors. After the original variables are projected into the space generated by the factors, each factor can be maximally correlated with a subset (cluster) of the original variables (machines) through the study of the loading coefficients. These clusters are the initial cells and the orthogonality of the factors guarantee the non-correlation of the cells thus formed. The factors obtained by the PCA method usually have a complex structure and are difficult to interpret since they may have a significant loading for many of the original variables. For solving this problem another method from statistical data analysis is used. Rotation methods transform the factors to simpler and more interpretable constructs. After rotation, each variable will be only related to one of the factors and each factor will have high loadings for only a small set of variables.

The proposed approach is implemented in two phases. In the first phase, the machine cells are formed by using factor analysis. In the second stage, the parts are assigned to the

machine cells by using an integer programming model. The steps of the proposed approach are demonstrated through an example [57].

4.2 Phase 1: Machine Cells Formation Using Factor Analysis

To apply factor analysis to a cell formation problem, the following major steps are carried out:

- 1) Generation of a similarity coefficient matrix for the machines. By considering similarity between machines as the measure of distance between every pair of machines, we can obtain the correlation matrix required for the principal component analysis method in the form of a similarity coefficient matrix; 2) Extraction of the initial cells using the PCA method. The percentage of the total variance explained by each factor is used to decide the number of cells; 3) Optimization of the initial cell formation using certain exchange techniques such as rotation.

The machine-part matrix is required for the generation of the similarity coefficient matrix. We proceed to discuss the three stages of the factor analysis approach to cell formation in the following sections. Also the program for calculation of eigenvalues and eigenvectors of the similarity coefficient matrix is provided in the appendix of the thesis.

4.2.1 Generation of a Similarity Coefficient Matrix

The machine-part matrix can be regarded as a given data set of m different binary patterns, where m is the number of machines. Each pattern corresponds to an n dimensional column vector, where n is the number of parts. Similarity coefficient, S_{ij} , is defined between any two machines by equation (4.1).

$$S_{ij} = \frac{\sum_{k=1}^N X_{ijk}}{\sum_{k=1}^N (Y_{ik} + Z_{jk} - X_{ijk})} \quad (4.1)$$

where:

X_{ijk} = operation on part k performed both on machine i and j ,

Y_{ik} = operation on part k performed on machine i ,

Z_{jk} = operation on part k performed on machine j .

This coefficient indicates maximum similarity when the two machines process the same part types, $S_{ij}=1$, and maximum dissimilarity when the two machines do not process the same part types, $S_{ij}=0$

The measure defined by equation (4.1) is called Jacard similarity measure [56]. Other similarity measures proposed in the literature could also be used. The similarity coefficient matrix thus defined is symmetric, with non-negative elements, and with diagonal elements equal to one. Therefore it is shown that it this matrix is a positive semi-definite matrix.

Optimally the data matrix has to be standardized. Since the input matrix is binary, the data matrix never needs to be standardized in this problem. Figure 4.1 is an example of an initial machine-part matrix involving six machines (labeled 1 to 6) and eight components (labeled 1 to 8). Applying equation (4.1) to the machine-component matrix given in Figure 4.1 yields the similarity coefficient matrix given in Figure 4.2. This figure illustrates machine pair similarity coefficient matrix for this initial matrix. The matrix in Figure 4.2 is considered as the

correlation matrix for the extraction of the initial cells since diagonal elements are 1 and S_{ij} is symmetric.

	C	O	M	P	O	N	E	N	T	S
M		1	2	3	4	5	6	7	8	
A		1	1			1			1	
C				1	1		1	1		
H		1	1	1		1	1		1	
I					1			1	1	
N				1	1			1		
E		1	1				1			
S		1	2	3	4	5	6	7	8	

Figure 4.1: Machine-component matrix [57].

1.00	0.00	0.67	0.17	0.00	0.40
0.00	1.00	0.25	0.40	0.75	0.17
0.67	0.25	1.00	0.12	0.12	0.50
0.17	0.40	0.12	1.00	0.50	0.00
0.00	0.75	0.12	0.50	1.00	0.00
0.40	0.17	0.50	0.00	0.00	1.00

Figure 4.2: Similarity coefficient matrix.

4.2.2 Extraction of Initial Cells

Assuming the machines as the original set of variables, and the similarity coefficient matrix as an estimate of the correlation matrix explaining the correlations between each pair of machines, we proceed to use the factor analysis framework for grouping the machines into separate independent clusters, which form the initial cells.

To extract out the initial cells, equation (4.2) is solved for the eigenvalues and the eigenvectors.

$$(S - I\lambda)Y = 0 \quad (4.2)$$

where S is the similarity coefficient matrix; I is the unit matrix; λ are the characteristic roots (eigenvalues); and Y are the eigenvectors.

From matrix computation theory, the similarity coefficient matrix should have m independent eigenvectors, where each eigenvector (factor) represents a cell. These cells have low inter-correlations because the eigenvectors are uncorrelated, and therefore there should be low similarities between machines that are associated with different cells. Furthermore, the elements of the eigenvectors reflect the degree of association between the cells and the machines.

To determine how many cells are needed to group machines, the user has two options, either to identify the required number of cells in advance, or consider it as a dependent variable. In both cases, the cells have to be ranked in a descending order according to the percentage of total variance explained by each cell. The total variance of each cell is the sum of the variances of all machines in the cell, or the eigenvalue corresponding to that cell. If the number of cells has to be determined by the user, then cells with highest eigenvalues should be selected. This criterion ensures that a high percentage of the variance is accounted for.

The computed eigenvalues for the matrix given in Figure 4.2 are listed in Table 4.1. The eigenvalue criterion indicates that only the first two cells are needed to group the machines. Table 4.1 illustrates the initial statistics for each cell. The total variance explained by each cell is listed in the column labeled eigenvalue. The next column contains the percentage of the total variance attributable to each cell. The last column, the cumulative percentage,

indicates the percentage of variance attributable to that cell and those that precede it in the table. Table 4.1 shows that almost 70% of the total variance are attributable to the first two cells. The remaining 4 cells together, account for only 30% of the variance. Thus, a model with 2 cells may be adequate to represent the data. One of the best advantages of this method is having the possibility of obtaining the optimum number of cells by considering the cells with the greater percentages of the total variance.

Table 4.2 displays the correlation loadings that relate the machines to these two cells. Each row of Table 4.2 contains the loadings used to express a machine in terms of the cells. Cells with large loadings (in absolute value) for a machine are closely related to that machine. For example, machine 1 is related more to cell 2 and not cell1, and the correlation between machine1 and cell 1 is 0.36. Unfortunately, it is difficult to decide whether machine 5 is related to cell 1 or 2. Thus, to achieve more interpretability, the initial cells corresponding to the resulting eigenvalues have to be optimized.

Table 4.1: The computed eigenvalues.

Cells	Eigenvalue	% of total variance	Cumulative percentage
Cell 1	2.38	39.70%	39.70%
Cell 2	1.81	30.00%	69.70%
Cell 3	0.78	13.00%	82.70%
Cell 4	0.53	9.00%	91.70%
Cell 5	0.29	4.80%	96.50%
Cell 6	0.2	3.50%	100.00%

Table 4.2: Machine-cell matrix.

Machines	Cell 1	Cell 2
1	0.36	-0.47
2	0.47	0.37
3	0.45	-0.41
4	0.38	0.31
5	0.44	0.46
6	0.33	-0.41

4.2.3 Optimizing the Initial Cell Formation

Although the matrix obtained in the extraction phase indicates the relationship between the cells and the individual machines, it is usually difficult to identify meaningful cells based on this matrix. In other words, in many cases, it is not easy to determine which machine corresponds to which cell. Often the machines and cells do not appear correlated in any interpretable pattern and most cells are correlated with many machines. To overcome this problem, the structure of the initial cells needs to be simplified. This could be done by using one of the rotations algorithms, such as the varimax method, which attempts to minimize the number of machines that are more correlated with a single cell.

To explain the basic concept of this method, consider the graphical representation of the two cells in Figure 4.3. As shown in Figure 4.3, the purpose of the varimax rotation algorithm is to rotate the axes so that each machine is close to only one of the two rotated axes. Mathematically, the algorithm finds new coordinates for each machine, b_{ij} , that maximizes objective function (4.3)

$$\sum_{j=1}^r \left[\frac{1}{p} \sum_{i=1}^p (b_{ij}^2)^2 - \frac{1}{p^2} \left(\sum_{i=1}^p b_{ij}^2 \right)^2 \right] \quad (4.3)$$

where r represents the number of cells and p represents number of the machines.

Applying the varimax method to the initial cells shown in Table 4.2 yields the matrix shown in Table 4.3. As shown in Table 4.3, machines 2, 4 and 5 have the highest loadings (correlation) on cell 1, while machines 1, 3 and 6 have the highest loadings on cell 2. Therefore, the best grouping for the six machines is to group them into two cells. Cell 1 consists of machines 2, 4, and 5, while cell 2 consists of machines 1, 3, and 6.

Table 4.3: Machine-cell matrix after rotation.

Machines	Cells	
	1	2
M1	-4.02E-02	0.594
M2	0.596	3.37E-02
M3	7.26E-02	0.607
M4	0.488	2.11E-02
M5	0.633	-6.16E-02
M6	-2.30E-02	0.523

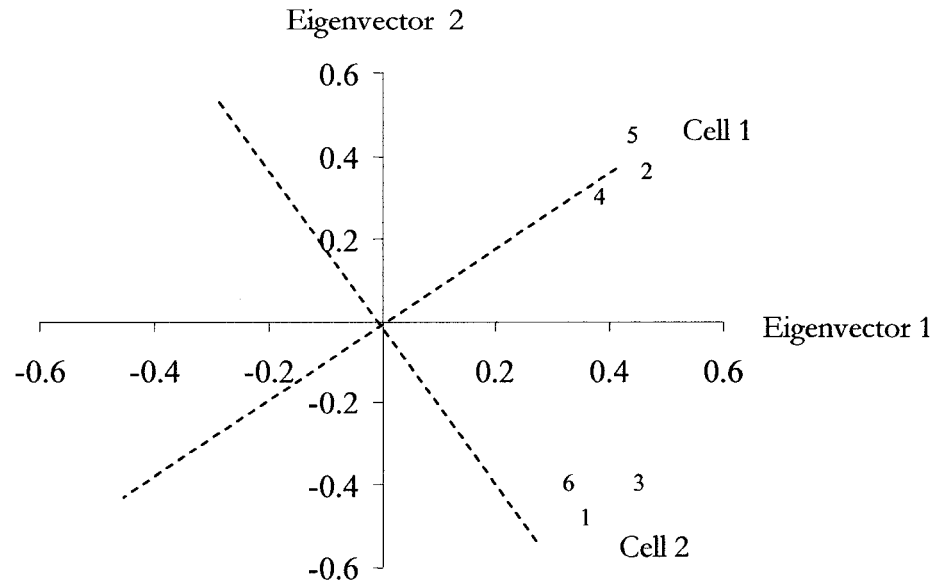


Figure 4.3 Machine and cells after rotation

4.3 Phase 2: Assigning Parts to Cells

To complete the cell formation, the parts need to be allocated to the machine cells. This can be done in one of the following ways: (i) Allocate each part to the machine group which performs the maximum number of operations on that part. If a machine group is not assigned to any parts, assign these machines to the groups where they can perform the maximum number of operations; (ii) Algorithms such as ROC or the DCA can be performed on the part columns alone, for the machine groups obtained [57].

Using the first option, the following simple integer-programming model is constructed for allocating parts to machine groups.

$$\text{Maximize } \sum_k \sum_c n_{kc} r_{kc} \quad (4.4)$$

Subject to:

$$\sum_c r_{kc} = 1 \quad \forall k \quad (4.5)$$

where $r_{kc} = 1$ if part k is assigned to cell c , and otherwise zero, n_{kc} is the number of visits to cell c by part k . Applying the model to the problem under consideration has yielded the cell given in Figure 4.3.

	C O M P O N E N T S							
M	1	2	8	5	6	3	4	7
A	1	1	1	1	1	1		
C	1	1	1	1	0			
H	1	1	0	0	1			
I			1			0	1	1
N					1	1	1	1
E						1	1	1
S	1	2	8	5	6	3	4	7

Figure 4.4: Final cell formation.

One major problem with clustering arises when a large number of components need to be processed in a machine (such a machine is called a bottleneck machine) or when exceptional elements exist in the machine-part matrix. Under these conditions, the parts or machines may not be divisible into mutually exclusive groups. In many cases, there is usually more than one copy of each type of machine. The machine-part matrix does not indicate the existence of such copies. If additional copies are not available, a machine can be purchased if

the associated reduction in inter- group travel cost is greater than the cost of duplication. If the saving in inter-group material handling cost equals or exceeds the duplication cost, the purchase of a new machine is justified. It is, however, recommended that other factors such as set up costs and cost saving due to better scheduling be considered in the decision-making process. Moreover, if the fraction is very small, other alternatives such as subcontracting or generating an alternative process plan should be considered prior to evaluating the duplication alternative. To determine the way duplication should be carried out, additional processing or human interfaces are needed [19].

The evaluation follows these steps: First of all, the algorithm clusters the problem set. If the results indicate that bottleneck machines or exceptional elements exist, additional procedures such as those described in this section are then applied. Such procedures, which introduce a human interface with the algorithm, offer much greater flexibility in dealing with exceptional elements and bottleneck machines.

I) The case of bottleneck machines

When bottleneck machines exist, components similar in nature may be dispersed over more than one cluster; therefore, it is difficult to form the diagonal block structure. The following procedure [31] can be followed to improve performance:

- 1) Identify the bottleneck machines.
- 2) Temporarily remove the bottleneck machines from the matrix.
- 3) Reapply the algorithm to generate a diagonal structure.
- 4) Duplicate the bottleneck machines into each group.

II) The case of exceptional elements

Exceptional elements may disrupt the clustering process and result in a poor cell formation. The following procedure [31] can be followed to improve performance:

- 1) Identify the exceptional elements.
- 2) Temporarily remove the exceptional elements from the matrix.
- 3) Reapply the algorithm to the matrix.
- 4) Restore the previously ignored exceptional elements.

4.4 Summary

This chapter has presented a mathematical approach for manufacturing cell formation. The approach consists of two phases. In the first phase, the similarity coefficients matrix is used as an input to factor analysis. Using principal components analysis, factor analysis extracts out initial cells. The initial cells are then optimized using the varimax rotation algorithm. Once the machine cells are formed, a simple integer-programming model is implemented in the second phase to identify part families and allocate them to the cells. Since the machine cells and part families are identified quantitatively, the proposed approach is expected to perform very well. A detailed analysis of performance is presented in the following chapter.

PERFORMANCE EVALUATION

In this chapter a number of objective criteria are used to evaluate the performance of the proposed approach. Moreover, the approach is tested against well-known existing methods in solving test problems from literature. This is followed by an analysis of the results.

5.1 Evaluation Procedure

For problems of even moderate sizes, determination of algorithm performance becomes very difficult. A variety of performance measures have been proposed. Stated by Chu [17] the performance of cell formation algorithms can be based on their computational efficiency or their grouping effectiveness. According to Chu [17] and Wei [67], computational efficiency of a method can be measured by computational complexity, execution time, or memory storage requirements. The determination of the grouping effectiveness measure is itself a challenging task. Some measurement criterion is necessary to compare the clustering solution to the original data, a standard result, or solutions from other algorithms. This criterion can be an independent measure or an aggregate measure. Two of the most commonly used independent measures are the number of exceptional elements produced and the total bond energy [46]. Since many heuristics use an objective function based on costs, a natural aggregate measure can be based on the minimum cost. Chandrasekharan and Rajagopalan [13] developed

grouping efficiency to measure the effectiveness of forming disjointed block diagonal sub-matrices. Grouping efficiency (GE) is a weighted sum of inter-cell movements and within-group utilization. A perfectly diagonal block solution with no voids in the blocks and no exceptional elements has an efficiency of 100%. Grouping efficiency has been used widely to determine cluster performance. Chandrasekharan and Rajagopalan [13] suggested giving equal weighting to inter-cell movement and machine utilization. However, Kumar and Chandrasekharan [33] observed in cases with more than two cells and large or sparse solution matrices, the machine utilization factor overshadows the inter-cell movement factor, making it virtually absent in the computation of the criterion.

Since they have been widely used in the literature, three objective criteria were selected for performance evaluation. These criteria are percentage of exceptional elements, machine utilization, and grouping efficiency [17].

5.1.1 Percentage of Exceptional Elements

The quality of a clustering can be measured by the number of machine-part cells that remain outside the diagonal blocks [12,30]. These off-diagonal machine-part cells are called exceptional elements. Percent of exceptional elements (PE), obtained from dividing the number of exceptional elements by the total number of elements with '1' in the entry, is used to include the possible effect of problem size. Better clustering algorithms results in a smaller percentage of exceptional elements.

$$PE = \frac{\text{number of exceptional elements}}{\text{total number of operations}} \times 100 \quad (5.1)$$

5.1.2 Machine Utilization

Machine utilization (MU) indicates the percentage of times the machines within the clusters are used in production. MU can be computed as [13]:

$$\text{Machine Utilization, MU} = \frac{N}{\sum_{k=1}^Q m_k p_k} \quad (5.2)$$

where N is the total number of ones within the part family-machine cell, Q is the number of cells, m_k is the number of machines in the k th cell, and p_k is the number of parts in the k th cell. Generally speaking, the higher the value, the better the machines are being utilized. However, in some special cases, even if the clustering method produces a smaller percentage of exceptional elements and higher total bond energy, the machine utilization may be lower.

5.1.3 Grouping Efficiency

Grouping efficiency (GE) is an aggregate measure, which takes both the number of exceptional elements and machine utilization into consideration. GE can be defined as [13]:

$$\text{Grouping efficiency, GE} = \eta = q\eta_1 + (1-q)\eta_2 \quad (5.3)$$

where η_1 is a measure of the density of ones in the diagonal blocks of the binary machine-part matrix, which is given by:

$$\eta_1 = \frac{N}{\sum_{k=1}^Q m_k p_k} \quad (5.4)$$

η_2 represents the density of zeros out side the diagonal blocks and define as:

$$\eta_2 = 1 - \frac{NE}{MN - \sum_{k=1}^Q m_k p_k} \quad (5.5)$$

MN is the size of machines-part matrix, and NE is the number of exceptional elements. q is the weight ranging between 0 and 1. Many researchers used $q=0.5$, but Kumar and Chandrasekharan [33] concluded that for any number of cells greater than 2, an equal value for q and $(1-q)$ may widen the disparity between the first and the second term of equation (5.3) They have suggested that q can be selected based on the size and number of cells in the solution as follows:

$$q = \frac{\sum_{k=1}^Q m_k p_k}{MN} \quad (5.6)$$

The result of maximizing η is the maximized concentration of ones in the diagonal clusters and minimized number of exceptional elements. As a general rule, the higher the grouping efficiency, the better the clustering results.

5.2 The Data Sets

Since clustering results are highly data dependent, the selection of data sets for testing is very important. The testing data can be either generated from a random number generator via computer or collected from the literature [17]. In order to compare the performance of the tested algorithms with other clustering algorithms published elsewhere, we decided to use the

existing data sets. Eight sets of data (problems) from the literature as shown in Table 5.1 have been collected for the evaluation. Table 5.1 summarizes the special features, the sources, and weighting factors, q , of these data sets.

Table5.1: Features and sources of cell formation problems.

Test Problems	Size	No. of Cells	Weighting Factor	References
1	5x7	2	0.5	King & Nakornchai [31]
2	5x7	2	0.5	Waghodekar & Sahu[66]
3	12x10	3	0.5	McAuley [45]
4	15x10	3	0.5	Chan & Milner [12]
5	8x20	3	0.5	Chandrasekharan & Rajagopalan [14]
6	14x24	4	0.5	King [29]
7	16x43	4	0.5	King & Nakornchai [31]
8	16x43	5	0.5	Burbidge [7]

5.3 Computational Results

To evaluate its performance, the proposed approach was applied to test problems listed in Table 5.1. The results of implementation are presented in Figures 5.1 to 5.8.

Initial Machine-Component Matrix									
M A C H I N E	COMPONENT								
		1	2	3	4	5	6	7	
	1		1		1	1	1		1
	2	1		1					
	3	1		1			1	1	
	4		1		1		1		
	5	1				1		1	
		1	2	3	4	5	6	7	

(a)

Final Cell Formation Matrix									
M A C H I N E	COMPONENT								
		3	1	7	2	4	6	5	
	1	1	1						2
	2	1	1	1			1		3
	3		1	1				1	5
	4				1	1	1	1	1
	5				1	1	1		4
		1	3	7	2	4	6	5	

(b)

Figure 5.1: The result of applying the proposed method on problem 1 (a) initial matrix, (b) final matrix.

Initial Machine- Component Matrix								
MACHINE A C H I N E	COMPONENT							
		1	2	3	4	5	6	7
	1		1		1			1
	2			1		1		
	3	1	1		1			1
	4	1		1			1	
	5			1	1	1	1	
		1	2	3	4	5	6	7

(a)

Final Cell Formation Matrix								
MACHINE A C H I N E	COMPONENT							
		5	3	6	7	2	4	1
	2	1	1					2
	5	1	1	1			1	5
	4		1	1				4
	3				1	1	1	3
	1				1	1	1	1
		5	3	6	7	2	4	1

(b)

Figure 5.2: The result of applying the proposed method on problem 2 (a) initial matrix, (b) final matrix.

Initial Machine-Component Matrix												
M	COMPONENT											
	1	2	3	4	5	6	7	8	9	10		
A	1							1	1	1		
	2	1	1	1	1						2	
C	3				1	1	1	1			3	
	4	1	1								4	
H	5					1	1	1			5	
	6	1	1	1	1						6	
I	7	1	1	1							7	
	8								1	1	8	
N	9	1	1			1	1				9	
	10					1	1	1			10	
E	11	1	1			1	1				11	
	12								1	1	1	12
		1	2	3	4	5	6	7	8	9	10	

(a)

Final Cell Formation Matrix												
M	COMPONENT											
		1	2	3	4	5	6	7	8	9	10	
A	2	1	1	1	1							2
	4		1	1								4
C	6	1	1	1	1							6
	7		1	1	1							7
H	3	1				1	1	1				3
	5					1	1	1				5
I	9		1	1		1	1					9
	10					1	1	1				10
N	11		1	1		1	1					11
	1								1	1	1	1
E	8								1	1		8
	12								1	1	1	12
		1	2	3	4	5	6	7	8	9	10	

(b)

Figure 5.3: The result of applying the proposed method on problem 3 (a) initial matrix, (b) final matrix.

Initial Machine-Component Matrix											
M	COMPONENT										
		1	2	3	4	5	6	7	8	9	10
A	1			1	1		1				
	2	1						1			1
C	3		1			1			1		
	4				1		1			1	
H	5		1			1			1		
	6			1			1			1	
I	7							1			1
	8		1			1			1		
N	9			1	1		1			1	
	10	1						1			1
E	11	1						1			1
	12	1						1			1
	13		1			1			1		
	14			1	1		1			1	
	15		1			1			1		
		1	2	3	4	5	6	7	8	9	10

(a)

Final Cell Formation Matrix											
M	COMPONENT										
		9	4	3	6	8	5	2	1	10	7
A	1	1	1	1	1						
	2	1	1	1	1						
C	3	1	1	1	1						
	4	1	1	1	1						
H	5	1	1	1	1						
	6					1	1	1			
I	7					1	1	1			
	8					1	1	1			
N	9					1	1	1			
	10					1	1	1			
E	11								1	1	1
	12								1	1	1
	13								1	1	1
	14								1	1	1
	15								1	1	1
		9	4	3	6	8	5	2	1	10	7

(b)

Figure 5.4: The result of applying the proposed method on problem 4 (a) initial matrix, (b) final

matrix.

Initial Machine-Component Matrix																					
MACHINE	COMPONENT																				
		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		1	1		1	1		1	
	2			1	1		1	1						1					1		1
	3			1					1	1		1		1	1		1	1		1	
	4			1	1		1	1			1								1		1
	5	1				1	1				1		1			1		1			
	6	1				1				1	1		1			1					1
	7			1	1		1	1				1	1						1		1
	8			1	1		1	1											1		1
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

(a)

Final Cell Formation Matrix																					
MACHINE	COMPONENT																				
		3	4	6	7	18	20	2	8	9	11	13	14	16	17	19	1	5	10	12	15
	2	1	1	1	1	1	1						1								
	4	1	1	1	1	1	1												1		
	7	1	1	1	1	1	1				1									1	
	8	1	1	1	1	1	1														
	1	1						1	1	1	1	1	1	1	1	1					
	3							1	1	1	1	1	1	1	1	1					
	5			1											1		1	1	1	1	1
	6						1			1							1	1	1	1	1
		3	4	6	7	18	20	2	8	9	11	13	14	16	17	19	1	5	10	12	15

(b)

Figure 5.5: The result of applying the proposed method on problem 5(a) initial matrix, (b) final matrix

Initial Machine-Component Matrix																									
M	COMPONENT																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
A	1					1	1																		
	2			1	1																				
C	3			1	1																1				
	4	1	1														1		1	1			1		
H	5	1	1														1			1			1		
	6								1	1	1	1		1	1	1						1			
I	7	1	1				1										1			1					
	8				1				1	1		1		1	1	1						1			
N	9				1				1			1	1		1										
	10			1																				1	
E	11			1	1																1			1	
	12						1	1																	
	13					1	1	1										1					1		
	14								1		1		1		1										
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

(a)

Final Cell Formation Matrix																									
M	COMPONENT																								
		3	4	21	24	1	2	17	19	20	23	5	9	10	11	12	13	14	15	16	22	6	7	8	18
A	2	1	1																						
C	3	1	1	1																					
	10	1			1																				
H	11	1	1	1	1																				
						1	1	1	1	1	1														
I						1	1	1		1	1														
						1	1	1		1													1		
N	6												1	1	1	1		1	1	1	1				
	8											1	1	1		1		1	1	1	1				
E	9											1	1			1	1		1						
	12											1			1		1		1						
	1																						1	1	
	12																						1	1	1
	13										1												1	1	1
		3	4	21	24	1	2	17	19	20	23	5	9	10	11	12	13	14	15	16	22	6	7	8	18

(b)

Figure 5.6: The result of applying the proposed method on problem 6 (a) initial matrix, (b) final matrix.

Initial Machine-Component Matrix																																														
M	COMPONENT																																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43			
A																																														
C		1								1																			1				1						1	1		1		1		
							1										1																			1	1	1								
H					1			1						1				1		1		1							1																	
I						1	1	1						1	1			1		1															1	1	1					1	1		1	1
		1											1															1																		
N		1	1	1					1	1					1				1	1	1		1	1				1	1											1	1			1		1
			1		1						1								1											1				1						1	1			1		1
E		1										1	1														1	1					1										1			
				1					1											1						1			1																	
			1																						1																					
						1												1																												
		1																																												
							1																																							
						1																																								

(a)

Final Cell Formation Matrix Matrix																																																		
M	COMPONENT																																																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47			
A	1	1	1	1	1	1																																												
C	1	1	1	1	1	1																																												
H	1	1				1																																												
I						1	1	1	1	1	1	1																																						
N						1	1	1	1	1	1	1	1																																					
E						1	1		1	1	1	1	1																																					
						1			1	1	1	1	1																																					

(b)

Figure 5.7: The result of applying the proposed method on problem 7 (a) initial matrix, (b) final matrix.

[illegible]

(a)

[illegible]

(b)

Figure 5.8: The result of applying the proposed method on problem 8 (a) initial matrix, (b) final

matrix.

The proposed approach was found to perform very well in terms of the criteria used. As shown in Table 5.2 the computed percentage of exceptional elements ranged from 0% to 14.8%, the percentage of machine utilization ranged from 50.2% to 100%, and the percentage of group efficiency ranged from 75.02% to 96%. The best clustering performance results that were obtained from the literatures are also included for comparison purposes [19]. As shown in Table 5.2, on average, the proposed approach compares favorably to rank order clustering, ROC [29], single direct clustering analysis, DCA [12], and bound energy [46] methods. Boe and Cheng [6] compared eleven algorithms by the number of times they give the best results in the test. The best result to a problem is defined as a grouping solution with the highest grouping efficiency. ROC, DCA, and BEA each produce the best results in 45%, 54%, and 82% of the problems, respectively. Boe and Cheng [6] also compared these eleven algorithms in terms of average grouping efficiency, and average CPU time. BEA has the highest average grouping efficiency and ROC and DCA each has the lowest average of CPU time among the eleven proposed algorithms. For these reasons these algorithms are considered as the standard algorithms for comparisons in the literature. Table 5.2 shows that the proposed approach produced the best results for all problems. It means that in all problems our algorithm gives the solution matrix with a desirable structure. On average the proposed approach generated higher grouping efficiency and machine utilization, with the lowest number of exceptional elements.

Table 5.2: The performance of the algorithm based on the computational results.

Problem	NC	ROC			DCA			BEA			Proposed Approach		
		PE%	MU%	GE%	PE%	MU%	GE%	PE%	MU%	GE%	PE%	MU%	GE%
1	2	12.5	82.4	85.6	18.8	76.5	79.4	12.5	82.4	85.6	12.5	82.4	85.6
2	2	18.8	72.22	77.29	18.8	72.22	77.29	12.5	82.35	85.62	12.5	82.35	85.62
3	3	13.2	78.6	86.1	18.4	81.6	86.5	13.2	82.5	88.1	12.8	85	89.4
4	3	0	92	96	0	92	96	0	92	96	0	92	96
5	3	21.3	86	86.6	*	*	*	14.8	100	95.8	14.8	100	95.8
6	4	6.6	66.3	82.3	6.6	66.3	82.3	3.3	68.6	83.9	3.3	68.6	83.9
7	4	0.8	50.2	75.02	*	*	*	0.8	47.17	73.5	0.8	50.2	75.02
8	5	2.4	61.8	80.7	2.4	59.9	79.8	2.4	61.81	80.7	2.4	61.8	80.7
<i>Average</i>		<i>9.45</i>	<i>73.69</i>	<i>83.70</i>	<i>8.13</i>	<i>56.07</i>	<i>62.66</i>	<i>7.43</i>	<i>77.1</i>	<i>86.15</i>	<i>7.38</i>	<i>77.79</i>	<i>86.5</i>

5.4 Summary

This chapter presented the results of the implementation of the proposed method on the data sets from the literature. The proposed approach was found to perform very well in terms of a number of objective criteria such as exceptional elements, machine utilization and grouping efficiency. Moreover, the approach compares very favorably to well-known existing methods such as rank order clustering (ROC) method, direct clustering algorithm (DCA) method, and bound energy method. Clustering and data reorganization methods such as bond energy algorithm (BEA), do not always give a solution matrix with a desirable structure. For some simple problems, the user may manually interchange some rows and columns of a solution matrix so as to obtain non-overlapping blocks. However, for large problems, the manual method is impossible. There are no known systematic procedures to replace the manual method for this purpose [6]. Array sorting (ARS) methods such as the ROC algorithm have

fast convergence and a low computation time. However, they do not always give good grouping solution to problems and must rely on the user to identify exceptional element and bottleneck machines that prevent the formation of a block structure in a solution matrix. The DCA algorithm cannot produce exact diagonal matrices [6]. The proposed approach performs better than the mentioned algorithms in two ways: (i) it gives the best diagonal block solution matrices; (ii) it employs specific data storage and analysis techniques, which greatly simplify the machine-component grouping process. The results of performance evaluation show that this approach not only converges quickly, but also gives quality results.

CONCLUSIONS

6.1 Overview and Discussion

During the last three decades of research numerous algorithms have been developed to solve cell formation problems and the interest appears to remain even today. Designing appropriate cells is the first step towards configuring a cellular manufacturing system. A properly designed cell seeks to provide a structural basis on which other issues could be studied further.

This thesis presented a new mathematical approach for the manufacturing cell formation. The principal objective of the proposed approach is to formulate a multivariate analysis model to generate optimal machine cells and part families. The original contribution of this thesis is to apply the rotation method to optimizing the cell formation. The approach has the capability of providing good solutions compared to other existing methods. Furthermore, because of its mathematical foundation the performance of the proposed approach does not deteriorate as the problem under consideration becomes larger. Real life problems are typically large, and solving such problems requires special solution procedures. This research provides a suggestion of a solution for this purpose.

The user of this approach has the option to identify the required number of cells in advance, or consider it as a dependent variable. This approach also generates all alternative solutions and provides the user with the opportunity to evaluate different options and choose

one of them. Furthermore, the proposed approach provides a mathematical means to test the suitability of adopting a cellular manufacturing system and it also has the flexibility to allow the cell designer to use various objective functions and incorporate design constraints during cell formation.

Another aspect of this research relates to the portability of it into practice. The approach does not require special software programs since it uses algorithms, which are available in many commercial software packages.

6.2 Future Work

The following recommendations are suggested for further research:

- The similarity coefficient measure discussed in this research requires only information provided in the machine-part matrix. The possibility of using the similarity coefficient that considered manufacturing features such as part volumes, part sequence, processing time, setup time, and other issues while computing the similarity measures can be consider as future work.
- Further research in this area should focus on developing approaches that meet the test of industrial application.

REFERENCES

- [1] G.K. Adil, D. Ragamani, and D. Strong. A mathematical model for cell formation considering investment and operational costs. *European Journal of Operational Research*, 69, 1993,330-341.
- [2] R. Askin and S. Subramanian. A cost-based heuristic for group technology configuration. *International Journal of Production Research*, 25(1), 1987, 101-113.
- [3] A. Ballakur and H.J. Steudel. A within-cell utilization based heuristic for designing cellular manufacturing systems. *International Journal of Production Research*, 25(5), 1987, 639-665.
- [4] A. Basilevsky. *Statistical Factor Analysis and Related Methods*. John Wiley & Sons ,Inc., 1994, Canada.
- [5] F. Boctor. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, 29(2), 1991, 343-356.
- [6] W. Boe and C.H. Cheng. A close neighbor algorithm for designing cellular manufacturing systems. *International Journal of Production Research*, 29(10), 1991, 2097-2116.
- [7] J.L. Burbidge. *The Introduction of Group Technology*. Halster Press and John Wiley, 1975, New York.

- [8] J.L. Burbidge. Comment on clustering methods for finding GT groups and families. *Journal of Manufacturing Systems*, 12(5), 1993, 428-429.
- [9] L. I. Burke and S. Kamal. Fuzzy art and cellular manufacturing. *In Artificial Neural Networks in Engineering*, St Louis, 1992, 779-784.
- [10] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, 37, 1987, 54-115.
- [11] A.S. Carrie. Numerical taxonomy applied to group technology and plant layout. *International Journal of Production Research*, 11(4), 1973, 399-416.
- [12] H.M. Chan and D.A. Milner. Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*, 1(1), 1982, 65-74.
- [13] M.P. Chandrasekharan and R. Rajagopalan. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24(2), 1986, 451-464.
- [14] M.P. Chandrasekharan and R. Rajagopalan. Modroc: an extension of rank order clustering for group technology. *International Journal of Production Research*, 24(5), 1986, 1221-1233.
- [15] M.P. Chandrasekharan and R. Rajagopalan. Groupability: Analysis of the properties of binary data matrices for group technology. *International Journal of Production Research*, 27(6), 1989, 1035-1052.

- [16] P.W. Chevalier. Group technology as a Cad/CAM integrator in batch manufacturing. *International Journal of Operations and Production research*, 3, 1984, 3-12.
- [17] C.-H. Chu. Cluster analysis in manufacturing cellular formation. *International Journal of Management Science*, 17(3), 1989, 289-295.
- [18] C.-H. Chu and J.C. Hayya. A fuzzy clustering approach to manufacturing cell formation. *International Journal of Production Research*, 29(7), 1991, 1475-1487.
- [19] C.-H. Chu and M. Tsai. A comparison of three array-based clustering techniques for manufacturing cell formation. *International Journal of Production Research*, 28(8), 1990, 1417-1433.
- [20] M.S. Daskin. An overview of recent research on assigning products to groups for group technology production problems. *Technical Report Working Paper*, 1991, Northwestern University.
- [21] O.O. Geoffrey, M. Chen, C. Chanaghit, and L.S. Richard. Manufacturing system cell formation and evaluation using a new inter-call reduction heuristic. *International Journal of Production Research*, 30(5), 1992, 1101-1118.
- [22] T. Gupta and H. Seifoddini. Production databased similarity coefficient for machine component grouping decisions in the design of a cellular manufacturing system. *International Journal of Production Research*, 28(7), 1990, 1247-1269.
- [23] T. Gupta. Design of manufacturing cells for exible environment considering alternative routeing. *International Journal of Production Research*, 31(6), 1993, 1259-1273.

- [24] S. A. Irani, *Hand Book of Cellular Manufacturing Systems*. Lohn Wiley INC., 1999, Canada.
- [25] A.K. Jajodia. Design of Manufacturing Cell with Multiple, Functionality Identical Machines., *M.Sc. Thesis*, Maryland, MB: University of Maryland, 1990, 108pp.
- [26] J.A. Joines. *Manufacturing cell design using genetic algorithms*. Ms thesis, North Carolina State University, 1993, Raleigh, NC.
- [27] J.A. Joines, C.T. Culbreth, and R.E. King. *A Comprehensive Review of Production Oriented Manufacturing Cell Formation Techniques*. Technical Report, NCSU-IE Technical Report, 1996, North Carolina State University.
- [28] G.M. Kern and J.C. Wei. The cost of eliminating exceptional elements in-group technology cell formation. *International Journal of Production Research*, 29(8), 1991, 1535-1547.
- [29] J.R. King. Machine-component grouping formation in group technology. *International Journal of Management Science*, 8(2), 1980, 193-199.
- [30] J.R. King. Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research*, 18(2), 1980, 213-232.
- [31] J.R. King and V. Nakornchai. Machine-component group formation in-group technology: Review and extension. *International Journal of Production Research*, 20(2), 1982, 117-133.
- [32] D. Kleinbaum, and L. Kupper. *Applied regression analysis and other multivariable methods*. Duxbury Press, 1978, California.

- [33] K.R. Kumar 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*, 28(2), 1990,233-243.
- [34] K.R. Kumar, A. Kusiak, and A. Vannelli. Grouping of parts and components in flexible manufacturing systems. *International Journal of Production Research*, 24, 1986, 387-397.
- [35] A. Kusiak. The generalized group technology concept. *International Journal of Production Research*, 25(4), 1987, 561-569.
- [36] A Kusiak. Group technology: Models and solution approaches. *First Industrial Engineering Research Conference*, 1992, 349-352.
- [37] A. Kusiak and W. Chow. Decomposition of manufacturing systems. *R4*, 4(5), 1988, 457-471.
- [38] A. Kusiak and I. Wadodd. Knowledge based system for group technology. *The 1st International Conference on CIM*, 1988, 184-193.
- [39] H. Lee and A. Garcia-Diaz. A network flow approach to solve clustering problems in group technology. *International Journal of Production Research*, 31(3), 1993, 603-612.
- [40] Leskowsky, Z., Logan, L. and Vannelli, A., *Group Technology Decision Aids in an Expert System for Plant Layout*. *Modern Production Management System*, Elsevier Science Publisher, 1987,561-583.

- [41] J. Li and Z. Ding. Fuzzy cluster analysis and fuzzy pattern recognition methods for formation of part families. In *16th North American Manufacturing Research Conference (NAMRC)*, 1988, pages 588-563.
- [42] R. Logendran. A workload based model for minimizing total intercell and intracell moves in cellular manufacturing. *International Journal of Production Research*, 28(5), 1990, 913-925.
- [43] B. Mahadevan and S. Venkataramanaiah. Re-aligning Research Objectives in Cellular Manufacturing Design: A User's Perspective. *Indian Institute of Management*, 2002.
- [44] C. O. Malave and S. Ramachandran. A neural network-based design of cellular manufacturing system. *Journal of Intelligent Manufacturing*, 2, 1991, 305-314.
- [45] J. McAuley. Machine grouping for efficient production. *Production Engineer*, 51(2), 1972, 53-57.
- [46] W. T. McCormick, Jr., P.J. Schweitzer, and T.W. White. Problem decomposition and data reorganization by a cluster technique. *Operations Research*, 20(5), 1972, 993-1009.
- [47] J. Miltenburg and W. Zhang. A comparative evaluation of nine well-known algorithms for solving the cell formation in group technology. *Journal of Operations Management*, 10(1), 1991. 44-72.
- [48] I. Minis, G. Harhalkis, and S. Jajodia. Manufacturing cell formation with multiple, functionally identical machines. *Manufacturing Review*, 3(4), 1990, 252-261.

- [49] G.F.K. Purcheck. A linear programming method for the combinatorial grouping of an incomplete power set. *Journal of Cybernetics*, 5, 1975, 51-76.
- [50] H. Seifoddini. Improper machine assignment in machine component grouping in group technology. *In Fall Industrial Engineering Conference*, 1986, 406-409.
- [51] H. Seifoddini and M. Djassemi. Merits of the production volume based similarity coefficient in machine cell formation. *JMS*, 14(1), 1995, 35-44.
- [52] H. Seifoddini and C. Hsu. Comparative study of similarity coefficients and clustering algorithms in cellular manufacturing. *JMS*, 13(2), 1994, 119-127.
- [53] H. Seifoddini and P.M.Wolfe. Application of the similarity coefficient method in group technology. *IIE Transactions*, 18(3), 1986, 271-277.
- [54] H. Seifoddini and P.M.Wolfe. Selection of a threshold value based on material handling cost in machine-component grouping. *IIE Transactions*, 19(3), 1987, 266-270.
- [55] S.M. Shafer and D.F. Rogers. Similarity and distance measures for cellular manufacturing part ii. an extension and comparison. *International Journal of Production Research*, 31(6), 1993, 1315-1326.
- [56] N. Singh. *System Approach to Computer-Integrated Design And Manufacturing*, John Wiley & Sons, Inc, 1996, U.S.A.
- [57] N. Singh, and R. Rajamani. *Cellular Manufacturing Systems*. Champan & Hall, Inc, 1996, UK.

- [58] G. Srinivasan, T. Narendran, and B. Mahadevan. An assignment model for the part families problem in group technology. *International Journal of Production Research*, 28(1), 1990, 145-152.
- [59] L.E. Stanfel. Machine clustering for economic production. *Engineering Costs and Production Economics*, 9, 1985, 73-81.
- [60] M.T. Tabucanon and R. Ojha. Icrm-a heuristic approach for intercell flow reduction in cellular manufacturing. *Material Flow*, 4, 1987, 189-197.
- [61] A.J. Vakharia, Y. Chang, and H.M. Selim. Cell formation in group technology: A combinatorial search approach. *Technical Report Working Paper*, University of Arizona, 1994, Tucson.
- [62] V. Venugopal and T. T. Narendran. A neural network approach for designing cellular manufacturing systems. *Advances in Modeling and Analysis*, 32(2), 1992, 13-26.
- [63] V. Venugopal and T. T. Narendran. Design of cellular manufacturing systems based on asymptotic forms of a boolean matrix. *International Journal of Production Research*, 67, 1993, 405-417.
- [64] V. Venugopal and T.T. Narendran. Cell formation in manufacturing systems through simulated annealing: An experimental evaluation. *International Journal of Production Research*, 63(3), 1992, 409-422.

- [65] V. Venugopal and T.T. Narendran. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers in Industrial Engineering*, 22(4), 1992, 469-480.
- [66] P.H.Waghodekar and S. Sahu. Machine-component cell formation in group technology: Mace. *International Journal of Production Research*, 22(6), 1984,937-948.
- [67] J.C. Wei and G.M. Kern. Commonality analysis: A linear cell clustering algorithm for group technology. *International Journal of Production Research*, 27(12), 1989,2053-2062.
- [68] U. Wemmerlov and N.L. Hyer. Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management*, 6(2), 1986, 125-147.
- [69] U. Wemmerlov and N.L. Hyer. Cellular manufacturing in the U.S. industry: a survey of users. *International Journal of Production Research*, 27(9), 1989,1511-1530.
- [70] H. Xu and H.P. Wang. Part family formation for GT applications based on fuzzy mathematics. *International Journal of Production Research*, 27(9), 1989, 1637-1651.

APPENDIX

```
/*#####
```

Program: Calculation of eigenvalues and eigenvectors.

Program description: This program creates the interface for inputting machine-part matrix to Matlab and calculates the similarity coefficient matrix. The output is the eigenvalues and eigenvectors of the calculated similarity coefficient matrix.

```
#####*/
```

```
% Interface for inputting machine-part matrix to Matlab
```

```
N=input('Row?\n')
```

```
M=input('column?\n')
```

```
A=zeros(N,M);
```

```
rep=input('If you want to enter the row and column of A matrix press y\n','s');
```

```
while (rep=='y')
```

```
i=input('Row of A matrix?\n');
```

```
j=input('Column of A matrix?\n');
```

```
A(i,j)=1;
```

```
if (i==N)
```

```
    rep=input('if you want to continue press y, if not press n\n','s');
```

```
end
```

```

end

save -ASCII Afile.asc A;

/*#####*/

clear

% the row number is the number of machines, and the size of N

% in similarity matrix is the same N.

N=input('Row?\n')

M=input('column?\n')

A=zeros(N,M);

n=zeros(1,N);

l=zeros(N,N);

p=zeros(N,N);

% calculation of similarity matrix from the machine-part matrix

load ('Afile.asc');

A=Afile;

for i=1:N

    for j=1:M

        if A(i,j)==1

            n(1,i)=n(1,i)+1;end

        end

    end

end

end

```

```

for i=1:N

    for k=1:N

        for j=1:M

            if (A(i,j)==A(k,j)) & (A(i,j)==1)

                l(i,k)=l(i,k)+1;end

            end

        end

    end

end

for i=1:N

    for j=1:N

        p(i,j)=l(i,j)/(n(1,i)+n(1,j)-l(i,j));

    end

end

% calculation of eigenvalues and eigenvectors

[v,d]=eig(p);

save -ASCII pfile.asc p;

save -ASCII vfile.asc v;

save -ASCII dfile.asc d;

load('pfile.asc');

load('vfile.asc');

load('dfile.asc');

```