# Cell formation problem - A Lagrangean relaxation to mathematical programming approach and a linear performance measure 

Talal Kattan<br>University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

## Recommended Citation

Kattan, Talal, "Cell formation problem - A Lagrangean relaxation to mathematical programming approach and a linear performance measure" (2007). Electronic Theses and Dissertations. 4627.
https://scholar.uwindsor.ca/etd/4627

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license-CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

# Cell Formation Problem- A Lagrangean Relaxation to Mathematical Programming Approach and a Linear Performance Measure. 

By

Talal Kattan

A Thesis<br>Submitted to the Faculty of Graduate Studies through the Department of Industrial and Manufacturing Systems Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at the<br>University of Windsor

Windsor, Ontario, Canada

2007
© 2007 Talal Kattan

Library and
Archives Canada
Published Heritage Branch

395 Wellington Street Ottawa ON K1A ON4 Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A ON4

Your file Votre référence ISBN: 978-0-494-34973-1 Our file Notre référence ISBN: 978-0-494-34973-1

## NOTICE:

The author has granted a nonexclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or noncommercial purposes, in microform, paper, electronic and/or any other formats.

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

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


#### Abstract

Two topics in the part-machine cell formation problem are discussed: In the first part, a Lagrangean relaxation in a mathematical programming model is proposed to simultaneously set machines into groups and parts into families in a cellular manufacturing system. The objective of this model is to find the optimal number of cells while minimizing inter-cellular part moves and increasing utilization of machines within the cells. The method uses a 0-1 integer programming model. The Lagrangean relaxation relaxes the model through an iterative search. In the second part, we introduce a new performance measure and compare it to some known performance measures. The new measure preserved some important features of previous performance measures and overcomes a number of drawbacks. Both the measure and the model are applied to benchmark problems as well as randomly generated problems. The new measure and model are comparable to the existing models and measures.


## DEDICATION

I would like to dedicate this thesis to all humanity past, present and future. I hope the knowledge obtained during the work towards this degree will allow me to give back and help to improve the live of the unfortunates, and alleviate some of poverty and hunger problems in this world.

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank GOD for the countless bounties he has granted me. I thank him for giving me the ability to deal with my challenges during my research. I thank him for letting me accomplish this thesis work.

Second, I would like to express my sincere gratitude to Dr. M. Fazle Baki for his time, guidance, patience and great support. His supervision and encouragement was really of a paramount importance to me in completing this work. It was a great experience to learn from him. His support with his NSERC fund helped me greatly during my course of study. Special thanks to my committee members, Dr. Y. Aneja for all the time and research help he has given me, Dr. G. Zhang for his suggestions, understanding and help and Dr. Z.J. Pasek for chairing my defence.

My gratitude goes to the faculty and staff of the University of Windsor for their hard work to make this university grow and for awarding me different scholarships. I thank the department of Industrial and Manufacturing Systems Engineering in specific for their generous help, support and for awarding me graduate assistantship throughout my study period.

Unconditional thanks to my parents that always hoped the best for me and are always looking forward to see me succeed. Their enthusiasm was a source of strength and inspiration throughout my life. I am really indebted to my parents who instilled in me the value of education and gave so much but asked for so little. I thank them for their financial and moral support.

An exceptional gratitude to my wife Amany for her enduring support, understanding, and patience in times when she needed help the most during her
pregnancy. She has been working hard and sacrificed to helped me reach my goals. Her love and presence gave me a boost to get to this point. I also thank my son Houzaifa for brining joy to my life. He and unborn baby were good reasons to work harder. I also appreciate the support of my sibling, and all my friends here in IMSE lab at the University of Windsor, and all over the world

Finally, I am very grateful to the country of Canada and whoever is working to make this country a wonderful place to live and improve the standard of living of the people. I thank the government for the financial support and for all the privileges given to citizens and residents of this country and for valuing research, science, and knowledge.

## TABLE OF CONTENTS

ABSTRACT ..... iii
DEDICATION ..... iv
ACKNOWLEDGEMENTS ..... v
LIST OF TABLES ..... ix
LIST OF FIGURES ..... xi
CHAPTER
I. INTRODUCTION
1.1 Manufacturing Systems ..... 1
1.2 Group Technology ..... 4
1.3 Cellular Manufacturing .....  5
1.4 The Cell Formation Problem ..... 9
1.5 Objective of Thesis ..... 13
1.6 Thesis Approach ..... 14
1.7 Outline of Thesis ..... 15
II. LITRATURE REVIEW
2.1 Overview ..... 16
2.2 Grouping Methods ..... 16
2.2.1 Manual Methods ..... 17
2.2.2 Array Based Clustering Methods ..... 20
2.2.3 Similarity Coefficient Clustering. ..... 22
2.3 Mathematical Modeling Methods ..... 24
2.4 Grouping Performance Measures ..... 29
III. PROBLEM STATEMENT
3.1 Problem Description ..... 37
3.2 Previous Approaches ..... 38
3.3 New Approach ..... 40
IV. NEW CELL FORMATION GROUPING APPROACH
4.1 Introduction. ..... 44
4.2 Notations ..... 44
4.4 Lagrangean Relaxation Algorithm ..... 48
4.5 Computational Analysis ..... 53
4.6 Discussion ..... 64
4.7 Review and Elaboration ..... 65
V. LINEAR PERFORMANCE MEASURE
5.1 Introduction ..... 67
5.2 New Performance Measure ..... 67
5.3 Analysis of the New Performance Measure ..... 69
5.4 Comparison with Other Measures ..... 70
5.5 Computational Analysis ..... 71
5.5.1 Simplicity ..... 71
5.5.2 Discriminating Power (Dispersion) ..... 72
5.5.3 Sensitivity (Response) ..... 74
5.5.4 Positive Measure ..... 76
5.5.5 Other Comparison Analysis ..... 77
5.5.6 Testing LPM ..... 80
5.6 Discussion and Review ..... 82
VI. CONCLUSIONS AND RECOMMENDATIONS
6.1 This Work ..... 84
6.2 Scope of future work ..... 86
APPENDICES
Appendix A ..... 87
Appendix B ..... 93
Appendix C ..... 96
REFERENCES ..... 99
VITA AUCTORIS ..... 104

## LIST OF TABLES

Table 1. 1 Percentage of improvement when CM is implemented ..... 8
TABLE 2. 1 BREAKDOWN OF DIGITS IN CLASSIFICATION AND CODING SYSTEM ..... 19
TABLE 3. 1 COMPARISON OF MATHEMATICAL PROGRAMMING METHODS THAT MIN E\&V OR MAX PM ..... 42
Table 3.2 Comparison of Performance Measures ..... 43
TABLE 4. 1 Step by step values when applying Lagrangean on an example. ..... 53
TAbLE 4.2 LIST OF SEVERAL BENCHMARK PROBLEMS ..... 58
TABLE 4.3 SOLUTION TO BENCHMARK PROBLEMS USING PROPOSED METHOD AND EFFICACIES FOR SOURCE AND PROPSED METHODS ..... 59
Table 4.4 Solution to randomly generated problems using proposed method. 60
TABLE 4. 5 DIFFERENCE IN TIME WHEN NUMBER OF CELLS $\mathrm{G}<N / 2 \mathrm{VS}$. $\mathrm{G}<\mathrm{N}$. ..... 63
Table 4.6 Effect of assigning upper and lower limits on Efficacy value ..... 64
TABLE 5.1 SIMPLICITY FACTOR OF DIFFERENT MEASURES ..... 71
Table 5.2 COMPARISON OF MEAN AND STANDARD DEVIATION FOR DIFFERENT PERFORMANCE MEASURE ..... 73
Table 5.3 Status of E and V For set of cases ..... 74
Table 5.4 Standard deviation of all measures ..... 76
Table 5.5 ANOVA test of Variance ..... 78
TABLE 5.6 Paired t-TEST ..... 79
TABLE 5.7 CORRELATION TEST BETWEEN MEASURES. ..... 80
Table 5.8 The difference in runtime of the model when using Efficacy vs. LPM

TABLE 5.9 SOLUTIONS TO 50 RANDOMLY GENERATED PROBLEMS ON THE MODEL WHEN USING Efficacy vs. LPM .................................................................................... 81

TABLE 5. 10 SUMMARY OF COMPARISON OF DIFFERENT FACTORS BETWEEN MEASURES ... 82

## LIST OF FIGURES

Figure 1. 1 Definition of a Manufacturing System (Black [1991]) ..... 1
Figure 1.2 Flow line layout ..... 2
Figure 1.3 Project layout ..... 3
Figure 1.4 Functional layout ..... 6
Figure 1.5 Cellular layout ..... 6
Figure 1.6 Different steps to Cellular Manufacturing ..... 7
Figure 1.7 EXAMPLE OF ZERO-ONE MACHINE-PART MATRIX ..... 10
Figure 1.8 EXample of perfect diagonal Block ..... 11
Figure 1.9 Void and exceptions in solved matrix ..... 12
Figure 2. 1 Production Flow analysis technique (SNEAd [1989]) ..... 18
Figure 2.2 CLASSIFICATION AND CODING SYSTEM ..... 19
FIGURE 2.3 ROC DENDROGRAM TREE THAT REPRESENT SIMILARITY BETWEEN MACHINES21
Figure 4. 1 Iterative search in Lagrangean relaxation ..... 50
FIGURE 4.2 LAGRANGEAN ALGORITHM FLOW CHART. ..... 51
Figure 4.3 EXAMPLE OF UNSOLVED PROBLEM ..... 52
Figure 4. 4 OPTIMAL SOLUTION FOR THE PROBLEM ..... 52
Figure 4.5 COMMUNICATION PROCEDURES BETWEEN LINGO 9.0 AND VB APPLICATION. ..... 55
Figure 4. 6 USER INTERFACE FORM USED TO INPUT INITIAL LAMBDA ..... 56
Figure 4. 7 USER INTERFACE FORM USED TO READ THE OUTPUT ..... 56
Figure 4. 8 SOLUTION TO 23X20 MATRIX (KUMAR ET AL. [1986]) ..... 61
FIGURE 4.9 OPTIMAL SOLUTION TO $23 \times 20$ MATRIX USING PROPOSED MODEL ..... 61
Figure 4. 10 Solution to 30x41 matrix (Kumar and Vannelli [1987]) ..... 62
Figure 4. 11 OPtimal solution to 30X41 MATRIX USING PROPOSED MODEL ..... 62
Figure 5.1 Discriminating power increases as standard deviation increases. . ..... 73
FIGURE 5.2 SENSITIVITY EVALUATION ON 40x100 MATRIX. ..... 75

## CHAPTER I

## INTRODUCTION

### 1.1 Manufacturing Systems

A Manufacturing System is an economic and industrial term for using men, material, and machinery in industry to make goods to satisfy customers. The manufacturing system takes inputs and produces products as its output (Black [2002]). It also includes interrelated activities and processes that define the primary design of the Manufacturing System (Figure 1.1).

a Physical elements:

- Machine tools for processing
- Tools and tooling
- Material handling equipment
- People (internal customers)
bMeasurable system parameters:
- Throughput time
- Production rate/cycle time
- Work-In-Process inventory
- Percent on-time delivery
- Percent defective
- Daily/weekly/monthly production volumes
- Total cost or unit cost

Figure 1. 1 Definition of a Manufacturing System (Black [1991])

There are five types of Manufacturing Systems: Job Shop, Flow Line, Project Shop, Continuous Process System, and Cellular Manufacturing System.

A Job Shop is a process where parts for different orders can follow different path or sequence on machines. This kind is characterized with its flexibility, skilled labour, and excessive material handling. Machines are grouped according to their functions. Large inventory, long process times and lost orders are some of the disadvantages of this type.

A Flow Line is a process where parts go through the same sequence of operations (Figure 1.2). Volumes are large and runs are long in this system. Lack of flexibility is one disadvantages of this type.


Figure 1. 2 Flow line layout

Project Shop is simply used to manufacture heavy and large sized products.
Material, labour, and machines are brought to a fixed location where the product will get
assembled or manufactured (Figure 1.3). A well defined sequence of operations is done to complete building a finished product.


Figure 1.3 Project layout

Continuous Process System normally processes liquid or gas which flows through sequenced integrated operations to link material with finished product. The Japanese have envisioned their discrete system to work similar to this type of system where products flow throughout the line in Just-In-Time style.

To achieve the objective of producing Just-In-Time, systems ought to be designed in a way to produce small lot economically. This requires a highly flexible manufacturing system or a cell based layout system called Cellular Manufacturing System (CMS) (Black [1991]).

### 1.2 Group Technology

In the twentieth and twenty first century, business and industry have taken new ways to operate. There was a substantial change in the management style. Industry has moved from Frederick Taylor's Theory of Scientific Management to the need for all levels of work force to share the responsibility. High quality is in demand at all times. Modern data exchange brought significant change to how systems work. Higher productivity with lower cost in the business and industrial world, lack of standardization and worldwide competition are all reasons why Group Technology had emerged and helped to overcome theses issues.

The concept of Group Technology has received a great deal of attention in North America and has led to an increasing effort to intensify the research in this field. Research has proven that Group Technology is very successful when implemented properly.

Group Technology is a management tool to help eliminate waste caused by duplication of effort. It is defined as finding a set of similar problems and grouping them, subsequently, discovering a single solution to apply to these problems, thus saving time and effort (Snead [1989]).

It is also defined as an approach to identify items by their attributes and looking for similarities among them; grouping the items into groups according to similarities and finally increasing the efficiency and effectiveness of managing the items by taking the advantage of the similarities (Shunk [1985]).

### 1.3 Cellular Manufacturing

For the past few decades, many firms with batch manufacturing or a job shop departmental layout have been trying to implement a practical concept called Cellular Manufacturing (CM) in order to increase standardization in the workplace and be more efficient.

CM is one of the applications of Group Technology in factory reconfiguration and shop floor layout design (Irani [1999]). CM has been proposed as an alternative to conventional layout, as illustrated in Figure 1.4 and 1.5, since it allows small batch production to gain similar economic advantages as mass production does while retaining the flexibility of job-shop production. Justification of CMSs and comparison between cellular layout and functional layout are reviewed and compared using comprehensive analysis by Agarwal and Sarkis [1998]. Shambu and Suresh [2000] reviewed the performance of a hybrid system of both a functional layout and a cellular layout. The study was done using a computer simulation.

CF or clustering is only the first step to designing a CMS (Figure 1.6). Other steps and issues taken into the design of CMS are machines' availability, sequence of operations and scheduling the machines within the cells, flexibility of machines, layout of the cells, quality control and finally human factors issues in the CMS.


Figure 1.4 Functional layout


Figure 1.5 Cellular layout


Figure 1.6 Different steps to Cellular Manufacturing

Some of the advantages of employing cellular manufacturing on the production floor (Snead [1989]) is to minimize material handling while maximizing facility, machine
and labour utilization. It's also used to increase employee morale and experience by decentralization. Wemmerlov and Hyer [1989] have shown the following dramatic improvement when CM is implemented (Table 1.1):

It was also found in a survey study conducted that the top five motivational factors for implementing CMS are:

1. Reduce cycle time
2. Improve product quality
3. Reduce WIP
4. Reduce material handling
5. Improve shop floor control

Table 1.1 Percentage of improvement when CM is implemented


Areas of research in the field of CMS were abundant throughout the literature. The three main areas of topical significance are:

1. Grouping and cell formation methods and their variations
2. Performance measures of grouping and their variations
3. Practical cellular manufacturing systems' design approaches

Grouping methods have drawn attention from many researchers. Although major grouping methods have not been formally divided into standard categories, they can be put into 3 major categories according to many papers published in the field of group technology (GT):

1. Manual methods such as Production Flow Analysis (PFA) and classification and coding methods
2. Clustering using a similarity coefficient (SC)
3. Grouping using mathematical models such as Integer Programming

### 1.4 The Cell Formation Problem

In CM, machines and parts are divided into groups so that each part family gets processed in one machine cell. Such grouping is possible and some parts require visiting more than one cell resulting in intercellular moves. When a new CMS is designed, cells are created so they can output a complete product. However, if an existing layout is identified as a CMS, the objective is to minimize the intercellular moves because they represent an additional capacity on cells to achieve independence.

Converting a functional layout to CMS requires route cards of parts that represent the route of each part throughout the manufacturing or assembly process. The information is then transformed into a matrix called machine-part incidence matrix. The matrix is a representation of the operations conducted for each part on different machines.

A machine-part incidence matrix contains zeroes and ones (Figure 1.7). An incidence matrix is constructed where rows represent machines while columns represent parts. An entry of " 1 " in row $i$ and column $j$ means that part $j$ visits machine $i$ for processing. On the other hand, an entry of " 0 " or blank in row $i$ and column $j$, means that there is no operation done on part $j$ by machine $i$. Information such as the sequence of operations and number of similar types of machine is not used at the stage of grouping. These are usually considered for further analysis of the CMS.

|  |  | Parts |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Machines | 1 |  | 1 |  | 1 | 1 |  |  | 1. |  |  |  | 1 |
|  | 2 |  |  | 1 |  |  |  |  |  |  | 1 | 1 |  |
|  | 3 |  | 1 |  |  | 1 |  | 1 | 1 |  |  | 1 | 1 |
|  | 4 |  |  | 1 | 1 |  |  |  |  |  | 1 | 1 |  |
|  | 5 | 1 |  |  | 1 | 1 |  | 1 |  |  |  |  | 1 |
|  | 6 | 1 |  |  |  |  | 1 |  |  | 1 |  |  | 1 |
|  | 7 |  |  |  |  |  |  |  |  |  | 1 | 1 |  |
|  | 8 | 1 |  |  |  |  | 1 |  |  |  |  |  |  |

Figure 1. 7 Example of zero-one machine-part matrix

One of the approaches to represent formation of cells is to diagonally block the matrix. Diagonal blocking the matrix in which cells are formed diagonally and row and column are interchanged so as to bring all the 1's to the diagonal. Figure 1.8 shows a matrix with a perfect block diagonal structure. In other cases where the diagonal structure is not perfect the blocks' boundaries are not clear and there are 1's outside the blocks.

Where the rows and columns are partitioned will determine different solutions.

|  |  | Parts |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Machines | 1 |  | \% 1 | 1 |  |  |  |  |  |  |  |  |  |
|  | 2 | 1 | 1 | $1$ |  |  |  |  |  |  |  |  |  |
|  | 3 |  |  |  | $1$ | $1$ | \% |  |  |  |  |  |  |
|  | 4 |  |  |  | $11$ | $1$ | 1 |  |  |  |  |  |  |
|  | 5 |  |  |  |  | $1$ | $1$ |  |  |  |  |  |  |
|  | 6 |  |  |  |  |  |  | $13$ | ? | \% 1 | $1$ | 1 | $1$ |
|  | 7 |  |  |  |  |  |  | ${ }^{1}$ |  | $15$ | 1 | 3 | 1 |
|  | 8 |  |  |  |  |  |  | $4$ | 1 | 4 | 4 | 1 | $1$ |

## Figure 1.8 Example of perfect diagonal Block

Some of the advantages of block diagonal structure are:

1. Feasibility of cell formation.
2. Identification of intercellular moves.
3. Alternative cell configuration for alternative number of cells.
4. Identification of bottleneck parts that have to visit more than two cells.
5. Identification of bottleneck machines that are shared by two or more parts from different families.

One of the earliest beliefs of Group Technology is that machine cells exist naturally, and the task of the researcher or manager is to discover them [Burbidge, 1971]. Therefore, researchers have developed and improved many grouping methods to achieve this goal.


Figure 1.9 Void and exceptions in solved matrix

Grouping methods are techniques or set of procedure to create block diagonal cells out of machines and parts. Voids (V) are zero elements in a cell block and is referred to a non utilized capacity of machine. Exception (E) is the operation off blocks and it is referred to an operation done on a part in a cell different than the one it belongs to. Voids and exceptions are illustrated in Figure 1.9.

### 1.5 Objective of Thesis

In this thesis a CF methodology is proposed to form cells and group machines and parts into these formed cells using a mathematical model. The idea is to obtain a system layout that can perform well to optimize the production and movement of parts in regard to cost. The primary focus of this thesis is on the solution of the cell formation problem. The problem was approached by using a performance measure as an objective similarly to others, however an exact method was used to solve the problem. The number of cells will not be set ahead, but rather the optimal number of cells will be determined by the model. Finally, the machine cells and part families are going to be grouped simultaneously.

In the second part a linear performance measure that is comparable to previously known measures is proposed. The new measure overcomes some drawbacks of previous measure and can be used as an objective function in the model to reduce the time to solve the problem

### 1.6 Thesis Approach

To achieve a comprehensive development of CMS design model, the research approach consists of the following steps:

1. Review literature and explain the importance of the model.
2. Setup objectives and goals. Draw a flowchart that explains how the model and algorithm are constructed.
3. Formulate the model and write the algorithm for dynamic production requirements incorporating limitations identified.
4. Code the model into a language that can be recognized by available commercial optimization software. Code the algorithm into a language that can be linked to the optimization software. Link the model and algorithm together to construct a package.
5. Searching for benchmark problems to serve as input data to validate the result of the model and compare its performance.
6. Solve the problem instances using the package and analyze results.
7. Evaluate the performance of the package and suggest modification that could improve the results.
8. Develop a new performance measure and incorporate it into the model.
9. Compare the performance measure to previously known measures and evaluate the result of using the measure into the model.
10. Draw a conclusion and discuss the direction of future work.

### 1.7 Outline of Thesis

A literature review of the CMS and CF problem and performance measures are given in chapter 2 . The chapter will review existing approaches to solving the CF problem. In addition, the review will categorize the various approaches by method of solution. The chapter will also introduce some important previous performance measures and why they were developed

Problem statement, description of the nature of the problem and how it was approached in this thesis in comparison to how it was approached by previous researchers will be presented in chapter 3 .

Chapter 4 will introduce the mathematical model with the algorithm. The chapter will provide an explanation of the notations and formulation. Solution procedure and applicable examples are considered with outputs and computational experimentation recorded for comparison.

In chapter 5, a linear performance measure will be introduced. In this chapter the given measure will be compared against well known measures in regard to highlighted features.

Finally, a summary and future improvements are presented in chapter 6.

## CHAPTER II

## LITRATURE REVIEW

### 2.1 Overview

The literature review is divided into 3 sections. The first section covers general grouping methods. The second section highlights various mathematical grouping methods. The last section goes through work done related to performance measures.

### 2.2 Grouping Methods

A number of researchers have been developing techniques for solving the CF problem. These techniques were classified and reclassified a number of times. Surveys of different grouping techniques are given in King and Nakoranchai [1982], Heragu [1994] and Chu [1989]. Comparison studies can also be found in Chu [1989] and Cheng, Kumar and Motwani [1994].

One classification of grouping methods can be given as:

1. Manual methods
2. Array based clustering methods
3. Similarity clustering methods
4. Mathematical programming and meta-heuristic methods.

The Manual methods refer to methods such as visual inspection or eyeballing, Part Code method and Production Flow Analysis (PFA).

The array based method rearranges arrows and columns in one or more steps. Ultimately, a diagonal block is formed within the matrix. The advantage of this method is that part families and machine groups are identified simultaneously.

The Similarity Coefficient (SC) method assesses the grouping of part family and machine groups on the ground of similarity of operations between every two parts. A similarity value of one suggests that they have high similarity and should be grouped together. Alternatively, a similarity value of zero suggests that there is no match between the two items which should be grouped separately. The disadvantage of this method is that additional steps are needed to complete the CF of the problem. It also cannot identify the similarity of part family and machine group simultaneously, and only one can be identified at a time.

Mathematical programming methods are those applying mathematical formulas. The method is to maximize or minimize an objective to find the best feasible solution. Meta-heuristic methods are used for large size problems but will not guarantee an optimal solution. They will lead to a solution close to optimal.

### 2.2.1 Manual Methods

Burbidge [1971] developed one of the most prominent manual methods, the Production Flow Analysis (PFA). Flowcharts are part of the analysis and they show material routings or process plan for each part on visited machines through different departments. This flow is noted and transferred to build the initial form of the machinepart incidence matrix which is used by other methods to create a CMS. PFA relies on relative judgments and does not have a clear approach. It is designed to improve
manufacturing and does not consider the design features or shapes of the part nor does it provide feedback on needed process improvements. One way the method is done is to subdivide the system into hierarchical stages and analyze each stage (Figure 2.1). A case study of PFA can be found in Dos Santos and De Araujo [2003].


Figure 2. 1 Production Flow analysis technique (Snead [1989])

The classification and coding method groups parts according to their shape, dimension, composed material, tolerance and operational requirement, explains Heragu [1994]. Each part is coded alpha-numerically with a series of ten to thirty digits codes with each code representing an attribute of the part. The part coding method is useful in a design-retrieval process (Mahesh and Srinivasan [2002]). MICLASS is one example of classification and coding and it is a hybrid code system that has the first twelve digits standardized. These digits represent the shape, form, dimensions, tolerances, and materials of the part as shown in Figure 2.2.


Figure 2. 2 Classification and coding system

Table 2. 1 Breakdown of digits in classification and coding system

| Code | Description |
| :--- | :--- |
| 0 | Organization and operation |
| 1 | Primary or raw materials |
| 2 | Commodities |
| 3 | Components |
| 4 | Toobassemblies and assemblies |
| 5 | Productive plant and spare parts |
| 6 | Auxiliary plant, services and utilities |
| 7 | Reserved for future need |
| 8 |  |
| 9 |  |

Another classification and coding system used for the design retrieval process is the Brisch Birn system. This system is all-numeric and is modified to meet the needs of the company. A breakdown of the first digit subclasses is shown in Table 2.1.

### 2.2.2 Array Based Clustering Methods

Clustering is a statistical tool to group entities or their attributes into clusters such that individual elements within a cluster have a significant degree of "natural association" among themselves and that there is very little "natural association" between clusters.

The array based clustering analysis method involves rearranging rows and columns of the machine-part matrix into diagonal block clusters. This approach was proposed by McCormick et al. [1972] and used by other researchers such as King [1980], King and Nakornchai [1982], Chandrasekharan and Rajagopalan [1986a] and Chandrasekharan and Rajagopalan [1987]. The method can be easily diagrammed and visualized, however, it is less practical (Figure 2.3). When solving the matrix, the clustering could grow to have no existing solution. Bond Energy Analysis (McCromic et al. [1972]), Rank Order Clustering (ROC), and Direct Clustering Algorithm are few examples of array-based clustering. Chu and Tsai [1990] performed a comparative study of BEA, ROC, and DCA. They found that BEA outperformed the other two at all times.

ROC was developed by King [1980]. The method reads the pattern of entries in each row and column of the machine part matrix as a binary word. It then rearranges the rows or columns in decreasing order. The procedure is repeated until all rows and columns are in rank order. This algorithm had a few major limitations, therefore, an extension of ROC (ROC2) was developed (King and Nakornchai [1982]). ROC2 is more
efficient and faster than ROC, however, solution generated is highly dependent on the initial matrix.

## Machines



Figure 2. 3 ROC dendrogram tree that represent similarity between machines

MODROC is a hybrid of ROC and Similarity Coefficient techniques (discussed in the next section). MODROC consists of three stages to the algorithm. In the first stage the ROC algorithm is applied for two iterations and the matrix is rearranged. A block is formed in the upper-left corner of the matrix. In the second stage, the rectangular block is identified and represents a primary cell with corresponding part family and machine group. The columns of the block are sliced off from further consideration. ROC
algorithm is applied on the truncated matrix repeatedly until all part families are identified. In the third stage, a similarity measure is used to compare the similarity between cells. After finding the similarities, the algorithm finds the pair of cells with the highest measure and joins the two into one cell. Once the cells are joined, the part families are joined correspondingly. Similarity measure is updated and checked. Stage three is repeated until similarity measure of each pair of part families is equal to zero. If this condition is not met, the process is repeated until the number of cells is equal to one.

The array based algorithms is simple and efficient in computation. Their limitation lays in being dependent on the initial configuration of the matrix (Srinivasan [1994]). This is a problem often found in clustering methods where different initial seeds will lead to local optimal points instead of global point.

### 2.2.3 Similarity Coefficient Clustering

McAuley [1972] introduced Similarity Coefficient (SC) to solve cell formation problem using Single Linkage Clustering (SLC). This method is based on hierarchal process of machine grouping done according to the computed similarity coefficients. Hierarchal clustering techniques use a matrix of similarity between parts to produce a hierarchy of cluster or partition in the progressive manner. The techniques are found to have some problems that lead to improper machine assignments in the groups. This will result into a situation called the chaining problem, when two machines have very high similarity measure and one of them has been included in a machines group already. The second machine will be included in that group automatically even though the similarity measure between the new included machine and the rest of machine in the group is low.

Seifoddini [1989] introduced the Average Linkage Clustering (ALC) to help overcome the chaining problem in the SLC algorithm. In ALC, the SC was the average of similarity between each pair of objects taken from two different clusters. Bottleneck machines and exceptional elements were identified. A bit-level data storage was used to reduce computational time and storage requirement. Complete Linkage Clustering (CLC) was further used to reduce the chaining problem by selecting the minimum SC as the linkage between the pair of objects drawn from two clusters (Gupta and Seifoddini [1990]), (Mosier[1989]).

Chandrasekharan and Rajagopalan [1986a] developed an algorithm called Ideal Seed Non-hierarchical Clustering (ISNC). At first, the problem is formulated as a graph that consist of a machine subgraph and a part subgraph. Then the k-means (Macqueen [1967] and Anderberg [1973]) is adapted to construct $k$ parts and $k$ machines and an evaluation criterion called group efficiency (discussed later) is used to compare different grouping alternatives.

Another non-hierarchical clustering technique called Zero-One Data Ideal Seed Algorithm for Clustering (ZODIAC) was introduced by Changrasekharan and Rajagobalan [1987]. ZODIAC is an improved version of ISNC which identify the part families and machine cells simultaneously. However, the initial seed selection of ZODIAC can still lead to form numerous singleton cells (cells with one member).

Graph theory method was introduced by Rajagopalan and Batra [1975]. The method is divided into 3 phases: First, cliques are identified in the machine-part graph. A clique represents a cell or subset of a machine cell. Second, a graph partitioning approach is used to identify machine groups. One or more cliques can form a cell if machine
relation is strong. Third and last, parts are assigned to cells (Srinivasan and Narendran [1991]).

### 2.3 Mathematical Modeling Methods

Mathematical Modeling, such as the p-median model (Kusiak [1987]) is also an important and speedy technique used to obtain a quick answer for larger size instances of the problem. Mathematical modeling tends to find the number of cells required and gathers the set of machines and parts into these cells by formulating the problem in linear or non-linear programming models. Kusiak's model overcomes the difficulties in representing and visualizing clusters for a large matrix and in obtaining the diagonal structure of the clustered matrix. The p-median of McAuley [1972] was used in the model and was among the first to solve the machine-part problem using mathematical programming by maximizing the total sum of the similarity coefficients between pairs of parts with the constraint that each part would be assigned to one family only.

This formulation was a successful starting point for many other researchers in mathematical programming. However, many used a modification to overcome the limitations of the original formulation. Others designed their own formulation in order to bypass the disadvantages of the first model (Wei and Kern [1989]). One of the limitations of the p -median approach is that the number of cells is determined a priori.

Dynamic programming was used by Steudel and Bailakur [1987] to maximize the bond between machines and then form machine cells that are subject to size restrictions. It was found that the model was flexible and good for uncertainty in demand. The
downside of this model is that any extension added to the model will require the model to be reconstructed again.

A binary integer programming approach that group machines based on compatibility of parts, was developed by Gunasingh and Lashkari [1989]. However, the model assumes that part families are known. Logendran [1993] developed a binary integer programming technique based on the simultaneous grouping of parts and machines to manufacturing cells. The objective of his model has been formulated as a maximization of the weighted sum of the fractions that represents the (negative of) total moves and in-cell utilizations. However, his model also assumes that the number of manufacturing cells should be set ahead of time within the model. Others, such as Gunasingh and Lashkari [1991] were able to simultaneously form cells based on tooling requirements, available tools and processing times.

Albadawi et al. [2005] used integer programming model in one of two phases of a mathematical model for cell formation. The first phase uses factor analysis to a matrix of similarity coefficient to form machine cells. The second phase used integer programming to assign parts to cells. The problem is tested on six problems from literature and found to perform well comparing to other methods. The performance of the approach does not deteriorate when larger size problems tested. An evaluation of trade off between process plan selection and cell formation was considered in a linear mixed integer model by Kizil and Ozbayrak [2004]. The evaluation starts with the developed algorithm, then machine cells will be determined by ROC algorithm.

A comprehensive mixed integer model was proposed by Defersha and Chen [2006] that incorporate several issues such as alternative routing and sequence of
operations. The formulation solves small size problems based on parts tooling requirements, and tooling available machines. A general integer programming with two stages was introduced by Slomp et al. [2005]. The model considers several aspects such as labour grouping cell size restriction in addition to machine-part grouping. The procedure is based on goal programming. A nonlinear mathematical model to minimize the total costs of inter- and intra-cell movements simultaneously was introduced by Tavakkoli-Moghaddam et al. [2007]. The model solves CMS problems with stochastic demand and an approximate approach was used to linearize the model.

Lashkari and Gunasingh [1990] introduced a Lagrangean relaxation model. The procedure solves problems based on tooling requirements and processing times. The objective is to maximize the sum of the compatibility indices of all parts and machines.

Assignment model to solve the cell formation problem is presented by Srinivasan et al. [1990]. Kusiak definition of similarity is used to construct a similarity matrix for machines. Similarities are maximized by an assignment model and groups are identified and finally parts are assigned to cells. The approach showed improvement over the pmedian in terms of computational time, however, the quality of solutions are moderate in cases of ill-structured problems.

Srinivasan and Narendaran [1991] developed GRAFICS which is an extension of assignment model. In this approach, initial seeds of machine groups are generated, then parts are assigned to machine groups using the "maximum density rule". An Iteration procedure is used to improve the result. GRAFICS is considered a non-hierarchical approach and found to deliver better results than other non-hierarchical cluster algorithm such as ZODIAC.

While some parts are totally exclusive, some are not clear to which cluster they should belong. Fuzzy logic was applied by Xu and Wang [1989] to the problem. Part features are transformed into fuzzy numbers. The membership functions are designed in a way to allow the fuzzy numbers to differentiate parts according to processing needs. To use fuzzy logic, a non-binary matrix was constructed. Each element in the matrix indicates the level of utilization of each part for each machine. Elements with fractional value means that other machine will be needed to process this part. Fuzzy logic was also considered by Josien and Liao [2002], Pai et al. [2005] and Torkul et al. [2006].

Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and Neural Network (NN) have been applied in a wide variety of application to solve hard engineering and math problems. The major advantage of using a meta-heuristic algorithms is that it can improve the computational performance. However, it cannot solve the problem optimally. Also the final solution is very sensitive to the initial seed or solution selected. Thus the solution quality is questionable.

GA is frequently used in combinatorial optimization problems due to its efficiency and flexibility. Venugopal and Narendran [1992] used this algorithm with the objectives of minimizing intercellular move and work load unbalance between the cells. The problem was solved as a multi-objective optimization problem. Dimpoulos and Mort [2000] considered GA to introduce a new similarity coefficient for CF. The proposed coefficient performed as good as other well known coefficients. Solimanpur et al. [2004], Rajagopalan and Fonseca [2006] and James et al. [2007] are recent literature that uses GA algorithm in CF.

SA has been successfully implemented in combinatorial optimization problems such as the travelling salesman problem and the flow shop scheduling problem. Boctor [1991] applied this technique to solve the cell formation problem. A. An initial feasible solution is required from which neighbour solutions are generated. If the neighbour solution is better than the initial solution, it is accepted. If the solution is worse, it might still be taken into consideration with a certain probability. The algorithm searches for the best solutions assuming that worse solution might lead to better ones and therefore avoid being trapped in local optima. Xambre and Vilarinho [2003], and Baykasoglu [2004] also used this meta-heuristic algorithm to solve the CF problem.

Neural Network has been applied by Kaparathi and Suresh [1993]. They have found that they can solve a large problem that result in close to perfect solution in a short period of time using adaptive resonance theory model (ART-1). They also found out that the quality of the result could be improved if they reversed the zero to one and one to zero in the matrix. Venugopal and Narendran [1994] used ART model with Self Organized Feature Map (SOFM) model and compared their results to ZODIAC algorithm. Kamal and Burke [1996] introduced an improved version of ART called FACT. This method consists of three stages and can do the grouping under multiple objective environments. The first stage is used to generate hierarchy of different clustering to select from. The second stage is used to extract information in the weight vector for simultaneous grouping. The third stage is used to group elements without clustering. Most recent work research in CF using NN method was done by Pierreval et al. [2003], Ozturk et al. [2006], Saidi-Mehrabad and Safaei [2007]. Recent TS can also be found in Cao and Chen [2004], Chen and Cao [2004], and Lei and Wu [2005].

Wang and Roze [1994] argue that the p-median original model was designed first to form part families and to deduce the number of machine cells present based on the machine-part incidence matrix. They suggest a modification to reduce the number of constraints when running the model. They also consider various constraints pertaining to cell size, in cases where there are a maximum number of parts or machines per cell. They have suggested another modification to improve the model (Wang and Roze [1997]).

Won and Lee [2004] suggest two modified versions of the p-median model for a faster implementation and cell formation. In their paper, they explain that the classical pmedian model is limited to small-size cell formation problems since it requires many binary variables. They propose an easy to implement formulation for large scale problems with 30 or more machines. They also report that many researchers introduced lower limit constraints to avoid the formation of singleton cell (i.e. cells containing only a single machine). They introduce a special set of machines called the candidate set, which is far more likely to serve as a seed machine. These machines would play a role in reducing the number of constrains without betraying which machine is the seed. Others who considered the p-median model are Hwang and Hui [2003], Won and Currie [2006], and Mukattash et al. [2007].

### 2.4 Grouping Performance Measures

As it can be seen, numerous methods for the block-diagonalization of a machinepart matrix have been suggested in hopes of reaching the best cell formation. There are so many cases of ill-structured incidence matrices that it is unclear as to which one gives the
best solution (Sarker \& Khan [2001]). As a result, there has to be a measure that can classify each method according to its performance when measured against certain criteria.

The performance measure is applied as a factor to be considered in addition to other traditionally used parameters for data collection and preparations of manufacturing systems. It is specifically adopted in capacity planning, facility planning and flexible manufacturing system loading. These applications urged the need for more development and standardization of efficiency measures. With the ability to quantify the attributes and decide the optimum technique and performance of CMS, more interest have arose to apply these measure in techno-economic, managerial and decision making problems (Sarker and Mondal [1999]).

Chandrasekharan and Rajagopalan [1986a] were the first pioneers in this area, publishing the first quantitative measure of the performance of a solution. Later, by using this measure they were able to show that ZODIAC (Chandrasekharan and Rajagopalan [1987]) can produce better solution from a binary matrix than any other method. The concept of performance measure is developed to provide a quantitative standard on a rational scale for comparing different solutions to the same problem. Chandrasekharan and Rajagopalan [1986a] define "grouping efficiency" as follows:

$$
\begin{equation*}
\eta=\mathrm{q} \eta_{1}+(1-\mathrm{q}) \eta_{2}, \tag{1}
\end{equation*}
$$

Where $\eta_{1}$ is the ratio of the number of ones in the diagonal blocks to the total number of elements of zeros and ones in the diagonal blocks. $\eta_{2}$ is the ratio of number of zeros in the off-diagonal blocks to the total number of elements of zeros and ones in the off-diagonal blocks and $q$ is a weighting factor between the values of zero and one.

This performance of grouping depends on two aspects: inter-group utilization and intercellular movement. So, a better grouping increases utilization and decreases intercellular movement. Some of the properties of this efficiency function are that it is non-negative and that its results are between zero and one, ( $0 \leq \eta \leq 1$ ). It also gives a weighting factor of $q$, thus choosing the relative weights of inter-group utilization and the intercellular movement. The weight can be altered by the user to assign a relative importance to the measure in non-utilized machines (voids) or intercellular moves (exceptions).

Although this measure has been a competent one, it has stimulated the need for more accurate measures to satisfy various criteria, such as machine capacity, manpower scheduling and material handling costs in the clustering problem. More and more researchers are looking for the limitation of this measure and are trying to work on a modification or develop a completely new measure.

Kumar and Chandrasekharan [1990] developed another measure called efficacy that gives equal weights to the number of voids and number of exceptions. They put together limitations, claiming that after analyzing 100 different matrices using grouping efficiency, their reported range of values were between $75 \%$ and $100 \%$. Thus, a worse case scenario, with a larger number of exceptional elements, can still result in 75\% efficiency. An analysis of the expression reveals that it is not true that the value $q=0.5$ leads to an equal weights to voids and exceptional elements (Kumar and Chandrasekharan [1990]). The second term becomes less effective as the matrix size increases. Therefore, the weight $q$ is more rational when linked to the size of the matrix. The grouping efficacy is defined as follows:

$$
\begin{equation*}
\Gamma=\frac{1-\Psi}{1-\Phi}=\frac{e-e_{0}}{e+e_{v}}, \tag{2}
\end{equation*}
$$

Where, $\Psi$ is the ratio of number of exceptions to the total number of operations, $\Phi$ is the ratio of the number of voids to the total number of operations, $e$ is the number of operations, $e_{v}$ is the number of voids and $e_{0}$ is the number of exceptions.

Grouping capability index (GCI) was introduced by Hsu [1990] to consider missing factors in efficiency and efficacy. He argued that only requirements of the machining of parts are considered in these measures. Factors such as processing times and operations, which are taken into consideration in GCI, are neglected in the two previously mentioned measures. He also claimed that his measure is more consistent in measuring how much a manufacturing system is suitable for cellular manufacturing. The grouping capability index includes only the number of exceptions, and the total number of ones in the matrix .The measure totally ignores the zero entries from the grouping efficacy measure and therefore the effect of voids on the solution quality is neglected. The GCI is defined as:

$$
\begin{equation*}
G C I=1-\frac{e_{0}}{e}, \tag{3}
\end{equation*}
$$

Where, $e_{0}$ represents the number of exceptions, and the total number of ones within the matrix is given by $e$.

Grouping measure was developed by Miltenburg and Zhang [1991] to also measure resources utilization and intercellular movements. They used this measure as a primary measure along with clustering measure and bond energy measure to evaluate nine different algorithms. The grouping measure is expressed as:

$$
\begin{equation*}
\eta_{p}=\eta_{u}-\eta_{m}, \tag{4}
\end{equation*}
$$

Where, $\eta_{u}$ is the ratio of the total number of ones in the diagonal blocks to the total number of elements of zeros and ones in the diagonal blocks. $\eta_{m}$ is the ratio of number of ones in the off-diagonal blocks to the total number of ones in the matrix. Higher value of $\eta_{u}$ implies higher usage of parts while higher value of $\eta_{m}$ implies fewer intercellular movements of the parts. Miltenburg and Zhang noticed that when the matrix is finally organized in block diagonal form, cells are better arranged when ones are clustered more tightly around the diagonal.

Shargal et al. [1995] experimented with different efficiency measures (neighbour clustering efficiency, ones clustering efficiency, ones zero clustering efficiency and others) and found that these measures relatively give similar values of efficiency. The measures were applied on 14 different problems with different sizes. They concluded that the choice of algorithm alone does not guarantee high value of efficiency measures. Also they indicated that most measures given in literature are computation intensive and therefore they are time consuming, however, they are specialized in particular area in measuring efficiency of different clustering solutions.

Nair and Narendran [1996] proposed the grouping index (GI) measure to overcome some of the drawbacks of the grouping efficiency and grouping efficacy measures. This measure considers that the block-diagonal space is critical factor to the weights of both voids and exceptions on the efficiency measures. Therefore the weighting factor is directly related to size and sparsity of the matrix. They have also added a correction factor into their derived measure in addition to diagonal space and weighting factor. The measure provides good discriminating power for problems with various sizes. The GI is defined as follows:

$$
\begin{equation*}
\gamma=\frac{1-\frac{q e_{v}+(1-q)\left(e_{0}-A\right)}{B}}{1+\frac{q e_{v}+(1-q)\left(e_{0}-A\right)}{B}}, \tag{5}
\end{equation*}
$$

Where, $B$ is the density (i.e., number of ones) of the solved matrix, $q$ is the weighting factor, and $A$ is a correction factor. The number of exceptions is given by $e_{0}$, and the number of voids is given by $e_{v}$.

Sarker [1997] considered a new measure called "A doubly weighted grouping efficiency measure" to eliminate the effect of the number of voids and the number exceptions in a goodness of grouping. The measure consist of two weighted term of efficiencies, the weighted intra-block efficiency of the diagonal blocks, and the weighted relative efficiency of the off-diagonal blocks in solved matrix. The measure is defined as follows:

$$
\begin{equation*}
\eta_{Q}=\left(\frac{q_{1} e_{1}+\left(1-q_{1}\right) e_{v}}{e_{1}+e_{v}}\right)\left(\frac{q_{2} e_{1}+\left(1-q_{2}\right) e_{0}}{e_{1}+e_{0}}\right) \tag{6}
\end{equation*}
$$

Where, $q_{1}$ and $q_{2}$ are weighting factors that varies from 0 to $1, e_{1}$ is the number of ones in the diagonal blocks of the matrix, $e_{v}$ is the number of voids in the diagonal blocks of the solved matrix, and $e_{0}$ is the number of exceptional elements in the solved matrix. This measure gives more weight to exceptions than voids, therefore, it is expected to assign a lower value weight for the number of exceptions. For equal weight on the two terms the value of $q_{1}$ and $q_{2}$ are replaced by $q$.

Sarker and Mondal [1999] conducted a survey and critical review of 13 existing measures. They found that the lack of standardization of measures and their definitions is getting in the way of research growth and application. They also mentioned that since it is quite impossible to have a universal scale of efficiency measure for overall CMS, tackling the principal components of an efficiency measure should be done first. An overall measure that can serve as a general model must be developed to allow for subjective selection of parameters for each specific case and could be applicable to different situations.

Keeling et al. [2007] conducted a simulation study on grouping efficiency measures and their impact on factory measures for the machine-part cell formation problem. Surveys and comparison of performance measure were conducted by Sarker [2001] and Sarker and Khan [2001] on mostly used measures that evaluated the goodness of grouping as well as several other measures that examine the problem from different perspectives or include different production factors.

Even though, many measures have been developed, only three certain measures to evaluate block diagonal matrix in previous literature and problems were highlighted in the analysis in this thesis. These measures were selected due to their popularity in the literature and because they do not require information beyond the data available in evaluated machine-part matrix.

Only a few number of papers considered using the performance measure or weighted void and exception as an objective function in mathematical models. These papers were discussed in the next chapter.

## CHAPTER III

## PROBLEM STATEMENT

### 3.1 Problem Description

Since the global economy is rapidly changing towards customized production, manufacturers tend to switch to lean manufacturing. This requires a flexible organization to produce different products at competitive prices. The majority of the production currently manufactured is in batch type production systems. In addition, the need for flexibility and just-in-time (JIT) processing are forcing many traditional manufacturing system to be restructured into cellular systems. In CMS, the reduction in set-up times can be achieved through finding similarity in operations of products rather than increasing the lot size.

CMS has been extensively researched in the last twenty five years. Several approaches have been implemented in the planning, design and execution stages. Implementing lean manufacturing by optimizing the process and minimizing waste is an essential step as the production system is getting larger and complex. Therefore, it is important to find an optimal, easier and faster method to implement the CMS.

There are many factors that affect the CF quality and time. One factor is the number of machines and parts. Increasing the number of machines/parts will increase the complexity of the problem. Another factor that affects the solution is the number of operations and the distribution of operations within the matrix.

The solution is used to find the optimal number of cells and assigns the parts to the corresponding cells. As the number of parts and/or machines increases, the amount of computation needed to find a solution is enormous. Also, once the combination of parts
or machines have changed, a re-evaluation of cell formation is needed to find the difference in cost.

The primary goal of this thesis is to find an optimal solution using an exact approach for different size problems. A mathematical exact method that is different in concept but comparable to other available methods that can reach the optimal solution. The model will also identify a near block diagonal structure to the data given and clearly identify cells. Other goals are to simultaneously group part families and machines cells, and to determine the optimal number of cells required. The number of cells should not be determined a priori.

### 3.2 Previous Approaches

Previous mathematical methods used different objectives to solve the CF problem such as maximizing the similarity between machine or part operations using different similarities. There were other methods that either minimized weighted sum of voids and exceptions or maximized efficacy.

Adil et al. [1996] describe a non-linear and a linear integer programming model. The linear integer program minimizes the weighted sum of exceptional elements and voids. For larger problems, Adil et al. used Simulated Annealing algorithm to reach a near optimal solution. The size of a cell is defined as the number of machines assigned to the cell times the number of parts assigned to the cell. Adil et al. show that minimizing the weighted sum of exceptional elements and voids is equivalent to minimizing the weighted sum of exceptional elements and cell sizes.

Kumar et al. [1997] developed a mathematical programming model to identify part families and machine groups simultaneously. The model considers minimization of weighted sum of voids and exceptional elements as the objective. An Iterative procedure called Assignment Allocation Algorithm (AAA) was proposed to solve the model. They noted that although the AAA provides a good solution, it is sensitive to the initial seed, the number of cells required, and the similarity and variation of the input data of the matrix. A Simulated Annealing algorithm was also developed for comparison and it was found that the later algorithm required greater more computational time than AAA. However, the quality of results was better when compared to AAA. A grouping performance measure called 'grouping measure' was used to judge the goodness of the solutions and compare them to other solutions using different methods.

Stawowy [2006] used a heuristics Evolutionary Strategy (ES) by maximizing the grouping efficacy. The ES provides new features such as new encoding/decoding mechanism for the permutation with separators representation and the concept of separators movements during mutation operator. The solution is represented by $n$ parts and $l$ separators of groups. The number of separators can be controlled by setting $l$ to a value set by the designer of the system. In the cases where the number of groups are not known, $l$ is set to $\operatorname{round}(n / 2)$. Efficacy measure was selected in the search because it is commonly used in literature and results are available for comparison. The algorithm assigns machines and parts to cells during evaluation of the fitness function and the assignment is done using the allocation rule. The algorithm can, in addition, group the parts into families and machines into cells simultaneously so visual analysis is not needed. The output of the solutions can have singleton, empty cells or a number fewer
than desired of part family. Therefore, these solutions are removed from the population during the selection phase through penalization.

Mehdavi et al. [2007] proposed a new model for cell formation based on cell utilization concept. The objective of the model is to minimize exceptional elements and voids. The objective function in the model considered minimizing the total number of voids in all cells. A minimum utilization of cells to be achieved could be specified as a constraint within the model. The nonlinear model was linearized and tested on small to medium size problems and the result of the model was compared to two other approaches, neural network and graph-neural network approach results.

### 3.3 New Approach

Previous heuristics used different objectives such as maximizing the similarity between machines or part operations using different concepts of similarity (e.g. pmedian). However, even the modified p-median by Won and Lee [2004] doesn't simultaneously group machines and parts. It does not automatically identify $E$ and $V$ elements. It just assigns machines and parts to suggested cells. The user eventually has to find out which elements are considered $E$ and which ones are $V$.

A few papers addressed exact methods where the optimal solution is not based on the seed that is fed to the method, instead the answer is always the same no matter at what point the algorithm starts. Alternatively, in a meta-heuristic method, the final solution can be obtained in a shorter period of time, yet the quality of the solution is questionable and is greatly based on the initial seed.

The objective is to find the optimal solution to various instance matrixes by minimizing $E$ and $V$. This is done by maximizing the efficacy which is a nonlinear function, therefore the function is relaxed using the Lagrangean relaxation. A trade off is observed in solving such a problem. If the objective is to form the cells by minimizing the exceptional elements, the obvious solution is to form one cell with all machines. Alternatively, if the objective is to form the cells by minimizing the voids, the obvious solution is to put every machine in a different cell. Neither solutions help to achieve the benefits of cellular manufacturing. Nor do they make planning, scheduling, and the control of operations any easier. Therefore, the ultimate objective is to find an optimal trade-off between exceptional elements and voids.

The model is designed to identify the status of every matrix element (whether it is a $E, V$ or neither one) in any feasible solution. The model also defines the status of machines and parts and the best alternative for joining a cell.

We also would like to be able to solve the matrix in one step. Simultaneous grouping of machines and parts at the same time will allow us to save time and spend less efforts. Many grouping methods consist of two or more steps to achieve the final formation.

The optimal number of groups are also vital to the solution process. Stawowy [2006] discussed that, in practice, the optimal number of groups is unknown a priori. He assumed that any optimization algorithm should be able to search for the maximum grouping efficacy value without constraining the number of permissible cells. Consequently, in the case where optimal number of groups is to be identified, the problems are much harder to solve because of the exploration space increase.

A comparison of studies for the Mathematical model is shown in Table 3.1. It can be noticed that only two of five methods can be considered exact methods. Three methods consider minimizing individual, sum, or weighted sum of voids and exceptions. The other two methods, consider maximizing efficacy if the method groups machine and parts simultaneously.

Mathematical programming to solve the machine-part problem for an improved cell formation has been introduced before, however, the concept of maximizing efficacy while using integer programming and Lagrangean relaxation to relax the non-linear objective in the proposed model is new. To bring down the complexity of nonlinear program, efficacy was relaxed through iterative search to find out the optimal solution in a reasonable time. Results show that the proposed integer program performs similarly or outperforms similar cell formation grouping methods with similar objectives in terms of goodness of grouping results.

Table 3. 1 Comparison of mathematical programming methods that min E\&V or max PM

| Year | Author | Method | Exact | Simultaneous | Min/Max |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1996 | Adil et al. | CF considering alternative <br> routing | No | Yes | Min(E,V) |
| 1997 | Kumar et al. | Minimization of weighted sum <br> of V\&E | No | Yes | Min(E,V) |
| 2006 | Stawowy | Evolutionary strategy | No | Yes | Max(Efficacy) |
| 2007 | Mehdavi et <br> al. | Nonlinear programming based <br> on utilization concept | Yes | Yes | Min(E,V) |
| 2007 | Kattan, Baki, <br> Aneja | Lagrangean and binary integer <br> programming | Yes | Yes | Max(Efficacy) |

In the case of performance measure (PM), so far there has not been any known measure which is linear while considering $\mathrm{E}, \mathrm{V}$ as a criteria of evaluation. A Linear Performance Measure (LPM) is introduced to evaluate different grouping methods. The
introduced LPM is needed to serve as a linear objective function in mathematical models. A linear objective function is faster and easier to compute than a non-linear objective. The LPM overcomes some of the drawbacks of previously compared performance measures. When compared with other known measures, the new measure is considered comparable with respect to: positive value, simplicity, discriminating power and sensitivity. In Table 3.2, a comparison of studies for PM shows that exceptions and voids are parameters considered in all performance measures except in GCI and GM . GCI is the only measure that neglected the effect of void on the performance. Alternatively, GCI and LPM are the only linear performance measures available.

Table 3.2 Comparison of Performance Measures

| Year | Author | Method | Linear | Min E, V |
| :---: | :---: | :---: | :---: | :---: |
| 1986 | Chandrasekharan and <br> Rajagopalan | Efficiency | No | Yes |
| 1990 | Kumar, Chandrasekharan | Efficacy | No | Yes |
| 1990 | Hsu | GCI | Yes | E |
| 1991 | Miltenburg and Zhang | GM | No | E |
| 1996 | Nair and Narendran | GI | No | Yes |
| 2007 | Kattan, Baki, Aneja | LPM | Yes | Yes |

## CHAPTER IV

## NEW CELL FORMATION GROUPING APPROACH

### 4.1 Introduction

Minimizing intercellular moves and maximizing machine utilization are important factors in CF that are considered in the introduced model. In this model, a mathematical programming approach is proposed to simultaneously group machines into groups and parts into families in a cellular manufacturing system. A corresponding $0-1$ integer model is formulated with an emphasis on the goodness of grouping performance measure called efficacy. A Lagrangean relaxation technique is adopted in our model to relax the nonlinear objective function. The mathematical model is discussed and illustrated using an example. The method is applied on various sizes of incidence matrices from literature. Results show that the proposed method performs similarly or outperforms similar cell formation grouping methods with similar objectives in terms of goodness of grouping.

### 4.2 Notations

The following notations will be used to formulate the problems or interpret the results:

```
\lambda= a weight factor;
G= the total number of cells formed;
m= the total number of machines in the machine-part incidence matrix;
n= the total number of parts in the machine-part incidence matrix;
i= machines index;
j= parts index;
k= cells index;
```

$g_{k}=$ a binary variable to represent the status of cell $k$ :
$g_{k}= \begin{cases}1, & \text { if cell } k \text { is formed; } \\ 0, & \text { otherwise }\end{cases}$
$a_{i j}=$ a binary variable to represent the status of elements in the matrix $\left\{a_{i j}\right\}$ :
$a_{i j}= \begin{cases}1, & \text { if part } j \text { visits machine } i \\ 0, & \text { other wise }\end{cases}$
$x_{i k}=$ a binary variable to represent the status of machine $i$ at cell $k$ :
$x_{i k}= \begin{cases}1, & \text { if machine } i \text { is assigned to cell } k \\ 0, & \text { otherwise }\end{cases}$
$y_{k j}=\mathrm{a}$ binary variable to represent the status of part $j$ at cell $k$ :
$y_{k j}= \begin{cases}1, & \text { if part } j \text { is assigned to cell } k \\ 0, & \text { otherwise }\end{cases}$
$e_{i k f}=$ a continuous variable that takes the value 0 or 1 and represents exceptions:
$e_{i k j}= \begin{cases}1, & \text { if } a_{i j}=1, \text { machine } i \text { is assigned to cell } k, \text { but part } j \text { is not assigned to cell } k \\ 0, & \text { othewise }\end{cases}$
$v_{i k j}=$ a continuous variable that takes the value 0 or 1 and represents voids:
$v_{i k j}= \begin{cases}1, & \text { if } a_{i j}=0, \text { but both machine } i \text { and part } j \text { are assigned to cell } k \\ 0, & \text { othewise }\end{cases}$

### 4.3 Mathematical Model

MAX $\frac{A-E}{A+V}$
SUBJECT TO

$$
\begin{array}{lll}
G-\sum_{k=1}^{m} g_{k} & =0 & \\
m g_{k}-\sum_{i=1}^{m} x_{i k} & \geq 0 & k=1,2, \ldots, m \\
\sum_{i=1}^{m} x_{i k}-g_{k} & \geq 0 & k=1,2, \ldots, m \\
n g_{k}-\sum_{j=1}^{n} y_{k j} & \geq 0 & k=1,2, \ldots, m \\
\sum_{j=1}^{n} y_{k j}-g_{k} & \geq 0 & k=1,2, \ldots, m \\
\sum_{k=1}^{m} x_{i k} & =1 & i=1,2, \ldots, m \\
\sum_{k=1}^{m} y_{k j} & =1 & j=1,2, \ldots, n \\
-x_{i k}+y_{k j}+e_{i k j} & \geq 0 & \forall i, j \ni a_{i j}=1, \forall k \\
e_{i k j} & =0 & \forall i, j \ni a_{i j}=0, \forall k \\
E-\sum_{i=1}^{m} \sum_{k=1}^{m} \sum_{j=1}^{n} e_{i k j} & =0 & \\
x_{i k}+y_{k j}-v_{i k j} & \leq 1 & \forall i, j \ni a_{i j}=0, \forall k \\
v_{i k j} & =0 & \forall i, j \ni a_{i j}=1, \forall k \\
V-\sum_{i=1}^{m} \sum_{k=1}^{m} \sum_{j=1}^{n} v_{i k j} & =0 & \\
e_{i, k, j}, v_{i, k, j} & \geq & \\
x_{i k}, y_{k j} & & \\
g_{k} & \in\{0,1\} & \forall i, j, k  \tag{23}\\
& \in \in 0,1\} & \forall k
\end{array}
$$

The Constraints of this model can be divided into 3 groups. Constraints (8-12) refer to cell formation, constraints (13-14) refer to the assignment of machines and parts to cells and finally, constraints (15-20) refer to the evaluation of void and exception elements.

Individual constraints can be introduced as follow: Constraint (8) sums up the number of all cells formed. Constraint (9) forces the $g_{k}$ to be assigned value of one if at least one machine is assigned to cell $k$. Constraint (10) ensures that there are one or more machine in every cell. Constraint (11) forces the $g_{k}$ to be assigned the value of one if at least one part is assigned to cell $k$. Constraint (12) ensures that there are one or more parts in every cell.

Constraints (8-12) count the number of cells. The count allowed us to use an additional constraint $G_{l} \leq G \leq G_{u}$, where $G_{l}$ and $G_{u}$ are, respectively, lower and upper limits on the number of cells. Clearly, $G_{l} \geq 1$ and $G_{u} \leq m$. It was experienced that $G_{l}=1$ and $G_{u}=m / 2$ save computational time without compromising optimality.

Constraint (13) ensures that every machine is assigned to one cell only. Constraint (14) ensures that every part is assigned to one cell only.

Constraint (15) checks if exceptional elements exist. The constraint forces $e_{i k j}$ to be one, when an operation, $a_{i j}$, exists while machine $i$ is assigned to cell $k$, and part $j$ is not. Constraint (16) assures the occurrence of inter-cellular movement only when $a_{i j}$ exists. Constraint (17) adds up the number of exceptional elements.

Constraint (18) checks if a void condition exists. The constraint forces $v_{i k j}$ to be one when an operation, $a_{i j}$, doesn't exist while both machine $i$ and part $j$ are assigned
to cell $k$. Constraint (19) assures the occurrence of a void only when $a_{i j}$ does not exist. Constraint (20) adds up the number of voids.

Constraint (21) restricts $e, v$ to be positive integer variables. Finally, constraints (22-23) force $x, y, g$ to be $0-1$ binary variables.

Optional constraints can be added to assign lower and upper bounds for the number of machines in each cell. The constraints are as follow:

$$
\begin{array}{ll}
\sum_{i=1}^{m} x_{i k} & \geq L_{m} \\
\sum_{i=1}^{m} x_{i k} & \leq U_{m} \tag{25}
\end{array}
$$

Where $L_{m}$ is the lower bound for the number of machines in cell $k$ and $U_{m}$ is the upper bound for the number of machines in cell $k$.

Similar constraints can be added to assign lower and upper bounds for the number of parts assigned to each cell. The constraints are as follow:

$$
\begin{array}{ll}
\sum_{j=1}^{m} y_{k j} & \geq L_{p} \quad k=1,2, \ldots, m \\
\sum_{j=1}^{m} y_{k j} & \leq U_{p} \quad k=1,2, \ldots, m \tag{27}
\end{array}
$$

Where $L_{p}, U_{p}$ are the lower and upper bounds for the number of parts in cell $k$.

### 4.4 Lagrangean Relaxation Algorithm

Based on the notion of key variables in efficacy measure, (i.e., number of voids and exceptions) a new integer linear programming formulation is introduced. Using the
formulation, a Lagrangean relaxation algorithm is derived. The problem of maximizing grouping efficacy is denoted by:

$$
\max \left\{\frac{A-E}{A+V}\right\}
$$

Now, we shall show that it is equivalent to $\min \{E+\lambda V\}$, where $\lambda$ is a Lagrangean improving every iteration

Let $\lambda^{*}=\max \left\{\frac{A-E}{A+V}\right\}$
$=\min \left\{\lambda: \lambda \geq \max \left\{\frac{A-E}{A+V}\right\}\right\}$
$=\min \{\lambda: \max \{(A-E)-\lambda(A+V)\} \leq 0\}$
$=\min \{\lambda: L(\lambda) \leq 0\}$
where, $L(\lambda)=\max \{(A-E)-\lambda(A+V)\}$

$$
\begin{aligned}
& =\max \{(A-\lambda A)-(E+\lambda V)\} \\
& =(A-\lambda A)-\min \{(E+\lambda V)\} \\
& \Rightarrow \min \{(E+\lambda V)\}, \text { since }(A-\lambda A) \text { is a constant }
\end{aligned}
$$

Therefore, $\lambda^{*}=\min \{\lambda: \min \{(E+\lambda V)\} \geq 0\}$
Let $\left(E^{*}, V^{*}\right)$ be the optimal solution to $\min \{E+\lambda V\}$
then $L(\lambda)=\left(A-E^{*}\right)-\lambda\left(A+V^{*}\right)$
Notice that $\lambda \geq(A-E) /(A+V) \geq 0$, since efficacy is always positive or zero

$$
\text { and } \max \{(A-E)-\lambda(A+V)\} \leq 0 \text {. }
$$

Since $\lambda^{*}=\min \{\lambda: \lambda \geq(A-E) /(A+V)\}, \lambda \leq 1$. Thus $0 \leq \lambda \leq 1$.

For each $\lambda,\left(E^{*}, V^{*}\right)$ are obtained by solving $\min \{E+\lambda V\}$.
Since $0 \leq E^{*} \leq A$ and $0 \leq V^{*} \leq A$, the function $L(\lambda)$ is a piece-wise linear, where each piece has a positive intercept $\left(A-E^{*}\right)$ and a negative slope $-\left(A+V^{*}\right)$ as illustrated in Figure 4.1. As shown, value of $L$ decreases until it reaches zero. In this case, L4 is the optimal point that satisfies the requirements.


Figure 4. 1 Iterative search in Lagrangean relaxation.

For $\lambda=0,\left(E^{*}, V^{*}\right)$ are obtained by solving $\min \{E:(4-19)\}$. An optimal solution is to put all machines in a single cell, yielding $E^{*}=0, V^{*}=m n-A$ and slope $=-m n$. As $\lambda$ increases, the weight assigned to $V$ increases, so $V^{*}$ does not increase. Therefore,
the slope $-\left(A+V^{*}\right)$ does not decrease. Thus, $L(\lambda)$ is convex. At optimal solution, $L(\lambda)=0$. So, $\lambda=\left(A-E^{*}\right) /\left(A+V^{*}\right)$.


Figure 4. 2 Lagrangean algorithm flow chart.

The following method (Figure 4.2) can be used to search for $\lambda^{*}$ :
Step 1: Take an initial trial value of $\lambda=A / m n$.
Step 2: Search for $\left(E^{*}, V^{*}\right)$ by solving $\min \{E+\lambda V\}$.
Compute $\lambda^{\prime}=\left(A-E^{*}\right) /\left(A+V^{*}\right)$. If $\lambda^{\prime}=\lambda$, then stop.
Step 3: Update $\lambda$ with $\lambda^{\prime}$ and go to step 2.

Assuming that the first trial value is $\lambda=A / m n$, then, $\left(E^{*}, V^{*}\right)$ is obtained by solving $\min \{E:(4-19)\}$. The next solution will give more weight to $V$ and therefore will decrease while $E$ is increasing. Algorithm will stop when the current value of $\lambda$ doesn't improve from previous iteration. An optimal solution will be reached by then and efficacy is the highest.

Consider the example in Figure 4.3. The incidence matrix with the following parameters, $m=4, n=6$, and $A=12$.

|  |  | Parts |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 |
| Machines | 1 | 1 |  | 1 |  | 1 |  |
|  | 2 |  | 1 |  | 1 |  | 1 |
|  | 3 |  |  |  | 1 | 1 |  |
|  | 4 |  | 1 |  | 1 |  | 1 |

Figure 4. 3 Example of unsolved problem.


Figure 4.4 Optimal solution for the problem.

A step by step process through the iterative search is illustrated in Table 4.1 and is done as follows:

## Iteration 1

Step 1: Take an initial trial value, $\lambda=A / m n=0.5$

Step 2: Solve $\min \{E+0.50 V\}$. The next solution is the one shown in Table 4.1. Assign machines 1,3 and parts 1,3 and 5 to cell 1 . Assign machines 2,4 and parts 2, 4 and 6 to cell 2. $E^{*}=1$ and $V^{*}=1$. Compute

$$
\lambda^{\prime}=\left(A-E^{*}\right) /\left(A+V^{*}\right)=(12-1) /(12+1)=0.8462 \neq \lambda .
$$

Step 3: Update $\lambda$ with $\lambda^{\prime}=0.8462$ and go to Step 2.

## Iteration 2

Step 2: Solve $\min \{E+0.8462 V:(4-19)\}$. The same solution is in the previous iteration. $E^{*}=1$ and $V^{*}=1$. Compute $\lambda^{\prime}=0.8462=\lambda$, Stopping criteria satisfied, so the algorithm stops.

Hence, an optimal solution is the one shown on Figure 4.4. Assign machines 1, 3 and parts 1,3 and 5 to cell 1 . Assign machines 2, 4 and parts 2,4 and 6 to cell 2 . The method converges very quickly for the example considered. This method requires two iterations when starting with $\lambda=A / m n=12 /(4 \times 6)=0.50$.

Table 4. 1 Step by step values when applying Lagrangean on an example.

|  | $m \times n=24$ |  |  |  |  |  |  | $A=12$ |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $E^{*}$ | $V^{*}$ | $\lambda^{\prime}$ | $\lambda$ | stop? |  |  |  |  |  |  |
| 1 | - | - | 0.05 | - |  |  |  |  |  |  |  |
| 1 | 1 | 1 | 0.84 | 0.5 | No |  |  |  |  |  |  |
| 2 | 1 | 1 | 0.84 | 0.84 | Yes |  |  |  |  |  |  |

### 4.5 Computational Analysis

The model presented in this thesis has many advantages over other models found in other literature. The advantages are:

1. It can be used to solve various size problems with comparable results.
2. The given parametric model provides flexibility by allowing one or more parameters to be modified in the model before being solved again.
3. This model groups machines and parts into cells simultaneously.
4. It determines the optimal number of cells required for the best alternative layout.
5. It allows the setting of upper and lower bounds of the number of machines or parts within each cell.
6. Solving the model results into a diagonal structure of the clustered matrices.
7. Different size clusters are solved with the best obtained value according to efficacy

The program was tested on a P 4 with 3.0 Ghz processor and 2.0 GB RAM. The linear programming model was coded on Lingo 9.0 commercial software to optimize the solution of the objective function. Since Lingo 9.0 does not have a looping function, an alternative software was needed to support Lingo in the Lagrangean relaxation iterative search. The Lagrangean requires the software to check the value of $\lambda$ in each iteration. Figure 4.5 illustrates this procedure.

To achieve the goal of iterative search and automate the algorithm, Visual Basic (VB) was used to communicate with the Lingo solver. The VB application calls the Lingo solver and passes a lambda value. Lingo solver solves the model and passes back some variables to the VB application for evaluation. The iterative procedure is carried out several times until the lambda value converges. Figure 4.6 and 4.7 shows the input and output user interface forms of the visual basic program.


Figure 4. 5 Communication procedures between Lingo 9.0 and VB application.
Several benchmark problems with different sizes, presented in Table 4.2, were collected from literature. Problems were solved in the source literature using different methods. A few aspects were considered when selecting the data sets from literature. These aspects can be listed as follow:

1. They have a single objective which is to minimize voids and exceptions.
2. Clusters should contains binary data of $(0,1)$.
3. Output solution should be diagonal.
4. No duplication of machines as part of the solution.
5. No operation sequence or process plan is set.


Figure 4. 6 User interface form used to input initial lambda.


Figure 4. 7 User interface form used to read the output.

The clustered data were collected and embedded into the proposed mathematical program as an input data. Results were compared with source approach and output were presented. When running each clustered data, the number of constraints and variables generated in the program are recorded. Theses values are considered a better index to measure the size of the problem than the multiplication of machines and parts numbers.

Data were collected into Table 4.3 after solving all 12 problems. $A, \mathrm{E}, V$ are the same as before, and $G$ denotes the optimal number of cells in the system. Since the data needed for efficacy became available, efficacy was calculated for both the proposed method and source method for the same problems. The two efficacies were compared and two different signs were marked next to the result. A "*" sign means that the efficacy of the source and proposed methods are equal and a "**" sign means that the proposed method had outperformed the source method in terms of efficacy results.

Time and efficacy are also recorded on solutions for 12 randomly generated matrices with 3 different sizes and 4 problems of each size. The number of voids, exceptions and the optimal number of cells created are recorded as well. Table 4.4 has the solutions output for the generated problems. Figure 4.8 and 4.10 show two examples of source paper solution while Figure 4.9 and 4.11 show the proposed model solution layout.

Table 4. 2 List of several benchmark problems.

| No | Size (mxn) | Reference |
| :--- | :--- | :--- |
| 1 | $5 \times 7$ | 1982, King and Nakoranchai |
| 2 | $5 \times 6$ | 1980, King |
| 3 | $5 \times 18$ | 1989, Seifoddini |
| 4 | $8 \times 20$ | 1986 b, Chandrasekharan and Rajagopalan |
| 5 | $8 \times 20$ | 1986 a, Chandrasekharan and Rajagopalan |
| 6 | $14 \times 24$ | 1987, Askin and Subramanian |
| 7 | $18 \times 24$ | 1973, Carrie |
| 8 | $23 \times 20$ | 1986, Kumar et al. |
| 9 | $20 \times 35$ | 1973, Carrie |
| 10 | $24 \times 40(1)$ | 1989, Chandrasekharan and Rajagopalan |
| 11 | $24 \times 40(2)$ | 1989, Chandrasekharan and Rajagopalan |
| 12 | $30 \times 41$ | 1987, Kumar and Vannelli |

Table 4. 3 Solution to benchmark problems using proposed method and efficacies for source and propsed methods .

| No | (A) | (E) | (V) | (G) | The proposed Method Efficacy | Source <br> Efficacy |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 14 | 0 | 3 | 2 | *0.82 | 0.82 |
| 2 | 12 | 0 | 3 | 2 | ${ }^{*} 0.80$ | 0.80 |
| 3 | 46 | 7 | 3 | 2 | ${ }^{*} 0.80$ | 0.80 |
| 4 | 61 | 9 | 0 | 3 | ${ }^{*} 0.85$ | 0.85 |
| 5 | 91 | 40 | 3 | 4 | 0.54 | N/A |
| 6 | 58 | 3 | 22 | 5 | *0.69 | 0.69 |
| 7 | 88 | 36 | 7 | 8 | 0.55 | N/A |
| 8 | 113 | 59 | 11 | 7 | ${ }^{* *} 0.44$ | 0.37 |
| 9 | 136 | 2 | 41 | 4 | ${ }^{*} 0.75$ | 0.75 |
| 10 | 131 | 0 | 0 | 7 | ${ }^{*} 1.00$ | 1.00 |
| 11 | 130 | 10 | 11 | 7 | ${ }^{*} 0.85$ | 0.85 |
| 12 | 127 | 77 | 32 | 10 | ${ }^{* *} 0.31$ | 0.27 |


|  | Tab | n to rand | rated | g pro |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | No | Size of Matrix | (A) | (E) | (V) | (G) | Time (sec) | Efficacy |
| $\underset{\stackrel{\rightharpoonup}{\sigma}}{ }$ | 1 | 5x7 | 17 | 1 | 3 | 2 | 4 | 0.80 |
|  | 2 | 5x7 | 15 | 3 | 4 | 2 | 9 | 0.63 |
|  | 3 | $5 \times 7$ | 12 | 2 | 3 | 3 | 6 | 0.67 |
|  | 4 | $5 \times 7$ | 10 | 0 | 5 | 3 | 5 | 0.67 |
|  | 5 | $8 \times 20$ | 62 | 19 | 23 | 3 | 1142 | 0.51 |
|  | 6 | $8 \times 20$ | 55 | 21 | 5 | 4 | 559 | 0.57 |
|  | 7 | $8 \times 20$ | 51 | 12 | 23 | 3 | 205 | 0.53 |
|  | 8 | $8 \times 20$ | 49 | 16 | 14 | 4 | 487 | 0.52 |
|  | 9 | 10x40 | 156 | 34 | 92 | 2 | 23958 | 0.49 |
|  | 10 | 10x40 | 135 | 43 | 60 | 3 | 16486 | 0.47 |
|  | 11 | 10x40 | 135 | 42 | 63 | 3 | 24844 | 0.47 |
|  | 12 | 10x40 | 102 | 31 | 59 | 4 | 21071 | 0.44 |


|  |  | Parts |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | ¢ | ¢ | ¢ | 訃 | 9 | 9 | N | $\stackrel{1}{\sim}$ | N | N | N | $\bar{m}$ | N | ¢ | $\stackrel{¢}{0}$ | $\cdots$ | $\underset{\sim}{\infty}$ | F | \% | \% |
|  | 5 | 4 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 6 | 0 | 0 | 1. | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 7 | 0 | U 4 | 414 | 4 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 8 | 0 | \% 1 | 1 | U 0 | 0 | \% | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 9 | 0 | 4 | 4. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 13 | , 1 | 0 | 0 | 0 | 0 | , 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 14 | 1. | W | 0. | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 17 | O. | O. | 1. | - | 0 | 0. | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1. | 4 | 1 | 1 | 1 | \% 0 | 1 | 4 | 1. | 1. | 3 | 1. | 0 |
|  | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | k. 1. | 0 | 0 | ). 1. | 0. | 1. | . 1. | 0 | 0. | 0. | , 0 | 0 | 0 |
|  | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1. | O. | 1. | 0 | \% 0 | 4. | 0. | 0 | 0. | 0 | 0 O | U4. | 1 | 1. |
|  | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0. | 0 | 4.1. | 0 | 0 | 0. | ]. | 4 | 0 | 0) | . 0. | 30 | 0 |
|  | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 | 1.15 | O | \% | 4. | \% 4 | 0 | 0. | , 1. | (0) | 0 |
|  | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 11 | ¢ 0 | \% 0 | 0 | 1 | 0. | 0 | 0. | \% 0 | 0 | 0 |
|  | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Q 0. | 4 | 0 | 0\% | 4. | 0 | 0 | 0. | H 1. | 4 | 0 | 0. | 4 |
|  | 15 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1. | 0 | 1 | 1)12 | 4t. | 0 | 1. | 1 | 4 | 0 | 1. | 0 | \% |
|  | 16 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | . 0. | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 | 9 | , 0 | 0. | 1 |
|  | 18 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | , 1 | 0. | 0. | 0 | 0 | -1 | , 1 | $\underline{4}$ | 0 | \% 4 | 0 | 0 | 0 |
|  | 19 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 20 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0. | 2. | 0 | . 0 | 1. | 14 | \% 0 | 0 | O | 0 | 0 |
|  | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20. | 11 | 0 | 0 | d | U1. | 0 | \% 0 | 0 | \#1: | - ${ }^{\text {a }}$ | 3 | 0 |
|  | 22 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 4. | [ 4 | 0 | 11 | \% 1 | , 1 | - 0 | 0 | 1. | \% 4 . | Q 0 | 0 | \% 1 |
|  | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | \% 0 | - 4 | - | - 0 | 0 | , 1 | 0 | 0 | 0 | 14 | - 0 | 3 | 0. |

Figure 4. 8 Solution to $23 \times 20$ matrix (Kumar et al. [1986])

|  |  | Parts |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $\infty$ | $\pm$ | N | N | $F$ | $\stackrel{\square}{6}$ | 앋 | - | 안 | - | + | \% | $\stackrel{\sim}{2}$ | 15 | $\omega$ | $\cdots$ | $\underset{\sim}{\infty}$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $\begin{aligned} & \mathbf{y} \\ & \underset{\mathbf{N}}{\mathbf{E}} \\ & \text { U } \\ & \mathbf{N} \end{aligned}$ | 3 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 12 | 1 | , | 1\% | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 15 | 1 | 1 | \% 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 16 | 1 | 1\% | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 5 | 0 | 0 | 0 | 1. | 1 | , 1 | T 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 14 | 0 | 0 | 0 | 4 | 0 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0. | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 1 | 1. | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 22 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 1 | \% 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|  | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 41 | 1 | 1. | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | $\underline{1}$ | 1. | 1. | 0 | 1 | 1 |
|  | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | O | 0 | 1 | 0 | 0 |
|  | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 4 | 1 | 1 | 14 | 0 | 0 | 1 |
|  | 7 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
|  | 13 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 4 | 0 | 0 |
|  | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
|  | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 11 | 1. |
|  | 18 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
|  | 20 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1.1 |

Figure 4. 9 Optimal solution to $23 \times 20$ matrix using proposed model


Figure 4. 10 Solution to $30 \times 41$ matrix (Kumar and Vannelli [1987])

|  |  | Parts |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | - |  |  |  |  | \% | N | 或的 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| $\frac{\stackrel{y}{8}}{\frac{5}{=}}$ | 24 |  |  |  | 0 | 0 | 0 | 00 | 00 | 0 | 00 | 00 |  |  | O | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 25 | 9 | 0 | 01 | 0 | - | 0 | 00 | 00 | 00 | 00 | 00 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 |  | 00 |  | 0 | 0.0 |  | 0 |  | 0 |  | 0 |  |  |  | 0 | 0 |  | 0 |
|  | 4 | 0 | 3 | 9 | 3 | 0 | , | 11 | 111 | 10 | 0 | 010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | D | 0 | 00 |  | 010 | 0 | 0 | 00 | 0 | 0 | 0 | 0 | 0 | 10 | 0 |  | 0 | 0 | 0 |  | 0 |
|  | 16 | 0 | 0 | 0 | 0 | 0 | 00 | 01 | 10 | 010 | 9) 0 | 9 | T | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 |  | 0 |  | 0 | 0.0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
|  | 6 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 00 | 00 | 0 |  | 0 | 0 | 0. | 9 |  |  | 0 | 0 | 0 | 0 | 00 |  | 1 |  | 0 | 00 | 0 | 0 |  | 0 | 0 |  |  | 0 | 0 | 0 | 0 |  | 0 |
|  | 12 | 0 | 0 | 10 | 0 | 0 | 10 | 100 | 00 | 010 | 0.0 | 010 |  | 0 |  |  |  |  |  |  |  |  |  | 00 |  | 0 |  |  | 0.0 |  |  |  |  |  |  |  |  | 10 | 0 |  |  | 0 |
|  | 15 | 0 | 0 | 0 | 0 | 1 | 00 | 00 | 00 | 00 | 00 | 010 | 0 | 0 | 0. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 0 | 01 |  | 0 | 0 | 0 | 0 |  | 0 | 0 | 0. | 0 | 0 | 0 | 0 | 0 |  | 0 |
|  | 22 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 00 | 00 | 0 | 0 | 1 | 1 |  |  | 0 | 0 |  | 0 |  | 00 |  | 0 |  | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |
|  | 23 | 0 | 0 | , | 0 | 0 | 0 | 00 | 00 | 0 | 10 | 010 | 0 | 0 | 1 | 1 |  |  | 0 | 0 |  | 0 | 0 | 00 |  | 00 |  | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | O | 0 | 0 | 0 | 0 |  |  |
|  | 19 | 0 | 0 | 00 | 0 | 0 | 010 | - 0 | 00 | 010 | 0 | 00 | 0 | 1 | 0 | 0 |  |  |  |  |  |  |  | 00 |  | 00 |  | 0 | 00 |  | 0 |  | 0 |  | 0 |  |  | 0 | 0 | 0 |  | 0 |
|  | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 00 | 0.0 | 0 | 0 | 0 | $\square$ | 0 | 0 |  |  |  |  |  | 00 |  |  |  | 0 | 00 |  |  |  | 0 |  |  |  |  | 0 | 0 | 0 |  | 0 |
|  | 30 | 0 | 0 | 0 | 0 | 0 | 07 | 00 | 00 | 010 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | , |  |  |  | 00 |  | 0.0 |  | 0 | 0 | 0 | 0 |  | 0 |  | 0. |  | 0 | 0 | 0 | 0 |  | 0 |
|  | 14 | 0 | 0 | 0 | 0 | 0 | 01 | 00 | 00 | 00 | 01 | 10 |  | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  | 010 |  | 0 |  | 0 |  |  |  |  | 0 | 0 | , |  | 0 |
|  | 18 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 0.0 | 00 | 00 | 0. | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | , | 0 | 00 |  | 0 |  | 0 | 90 |  | 10 |  | 4 |  |  | 0 | 0 | 0 | 0 | 1 |  | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 000 | 00 | 00 | 00 | 0.0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 010 |  | 00 |  | 0 | 00 |  | 10 |  | 0 |  |  |  |  |  | 0 | 0 |  | 0 |
|  | 27 | 0 | 0 | 0 | 7 | 0 | 0 | 00 | 00 | 07 | 00 | 00 |  | 0 | 0 | 0 |  | , |  |  |  |  |  | 00 |  | 00 |  |  | 0 |  | 0 |  | 0 |  |  |  |  |  |  | 0 |  | 0 |
|  | 28 | 0 | , | 0 | 0 | 0 | - | 000 | , | 0 | 00 | 00 | 0 | 0 | , | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 |  | 00 |  | 00 | 00 | 0 | 0 |  | 0 | 0 | 0 |  |  |  | 0 | 0 |  | 0 |
|  | 29 | 0 | 0 | 0 | 0. | 0 | 00 | 00 | 00 | 00 | 00 | 00 | 0 | 1. | 0 | 0 | 0 | 0 | 1 |  |  | 0 | 0 | 00 |  | 00 |  | 00 | 011 | 0 | 0 |  | 0 |  | 0 |  |  | 10 | 0 | 0 |  | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 00 | 0 O 0 | 00 | 0.0 | 00 | 00 |  | 0 | 1 | 0 | 0 | 1 |  | 0 |  | 0 |  | 0.0 |  |  |  | 0 | 0 |  | 0 | 0 |  |  |  |  |  | 0 |  |  |  | 0 |
|  | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 00 | 00 | 0.0 | 00 | 0 | 0 | 0 | 11 | , | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 01 |  | 00 |  | 0 | 0.0 | 0 | 0 |  | 1 |  |  |  |  | 0 |  |  |  | 0 |
|  | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 000 | 00 | 00 | 00 | 0. | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0. |  | 1 | 0 | 0 | 0 | 0 |  |  |  |  |  | 0 | , |  |  | 0 |
|  | 5 | - | 0 | 0 | 0 | 1 | 0 | 0 | 00 | 0 | 01 |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 |  |  |  |  |  | 0 | 00 |  |  |  | 0 |  |  |  |  |  | d |  |  | 1 |
|  | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 0.0 | 00 |  | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  | 00 |  | 0 |  | 010 |  |  | 0 |  | 0 |  |  |  |  | 0 |  |  |  | 0 |
|  | 10 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 01 | 10 | 010 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |  | 0 | 0 | 00 |  | 00 |  | 0 | 00 |  | 0 |  | 0 | 0 |  |  |  | 0 |  |  |  | 0 |
|  | 11 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 00 | 00 | 00 |  |  |  |  |  | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 00 |  |  |  | O |  |  |  |  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |  | 0 |
|  | 13 |  |  | 0 | 0 | 0 | 0 | 110 | 00 | 0 | 00 | 0 |  | 0 |  | 0 |  |  |  |  |  |  |  | 00 |  |  |  | 0 | 0 |  |  |  | 0 |  |  |  |  | 0 |  |  |  | 0 |
|  | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 00 | 00 | 0 | 00 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 |  | 00 |  | 0 | 0.0 | 0 | 0 |  | 0 | 0 |  |  | 0 | 0 | 0 | 0 |  |  |
|  | 17. | 0 | 1 | 0. | 0 | 0 | 00 | 010 | 00 | 00 | 00 | 0. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 00 |  | 00 |  | 0 | 010 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 01 |  | I |
|  | 21. | 0 | 0 |  | 0 | 0 | 0 | 00 | 00 | 00 | 0 | 0. |  | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  | 0 | 0 |  |  |  | 0 | 0 | 0 | 0 |  |  |  |
|  | 26 |  | 0 | 0 | 0 | 0 | 010 | 1010 | 00 | 00 | 010 | 00 |  | 0 | 0 | 10 |  |  | 0 |  |  |  |  |  |  |  |  |  | 010 | 10 | 10 | 0 | 0 | 0 |  | 0 | 10 | 10 |  |  |  |  |

Figure 4. 11 Optimal solution to $30 \times 41$ matrix using proposed model

An experimentation is conducted to improve the result quality or time of the model. A rearrangement of rows and columns is performed on a matrix to see if the starting seed (the arrangement of 0 and 1) would affect the solution time or quality. The solution before and after the rearrangement is found to be the same

Another experimentation is conducted to reduce the computational time without altering the final solution through reducing the maximum number of groups created to less than half of the number of machines (if number of machines are odd then less than half of number of machines +1 is used instead). In an example, if the number of machines $m=10$, then the maximum number of groups allowed is $G_{u} \leq 5$. The experimentation was successful and the time to solve the problem was reduced significantly. Table 4.5 presents some examples of the time to solve the same problems before and after reducing the upper limit of cells.

Table 4. 5 Difference in time when number of cells $G<n / 2$ vs. $G<n$.

| No | Size (literature case) | $\mathrm{G}<\mathrm{n} / 2$ | Time $1(\mathrm{sec})$ | $\mathrm{G}<\mathrm{n}$ | Time 2 (sec) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | $5 \times 18(3)$ | 3 cells or less | 29 | 5 cells or less | 154 |
| 2 | $8 \times 20(4)$ | 4 cells or less | 159 | 8 cells or less | 2030 |

IP relaxation was considered assuming all binary variables as continuous. When running the program, the solution to a small problem assigned values of 0.5 or 0 to most variables. No variables were assigned a value close to 0 or close to 1 . Therefore, the solution was not useful and was not considered.

It can be realized from the model that the number of variables and constraints as a function of number of machines and parts is shown below as:

$$
\# \text { binary variables }=m(1+m+n) \Rightarrow=O(m n)
$$

\# constraints $=3+n+m(m+n+4 m n+6+A) \Rightarrow=O\left(m^{2} n\right)$

Therefore, problem complexity increases dramatically with the increase of $m$ and $n$.

Table 4. 6 Effect of assigning upper and lower limits on Efficacy value.

|  | (Literature case \#5: 8x20) |  |  |
| :---: | :---: | :---: | :---: |
|  | Not set | Set 1 | Set 2 |
| Um | - | 2 | 3 |
| Lm | - | 2 | 2 |
| Up | - | 5 | 7 |
| Lp | - | 4 | 5 |
| Efficacy | 0.85 | 0.51 | 0.51 |

By assigning an upper and lower limits to the number of machines and parts within any cell, the solution will vary and thus the efficacy value will change. In Table 4.6, an example from literature was solved initially without any limits assigned, and then was solved twice with different limits assigned. Efficacy and the number of cells were recorded for all cases.

### 4.6 Discussion

As it can be observed from Table 4.3 that solutions to most of the problems using the proposed integer model give equal efficacy to solutions given by literature methods. Problems 5 and 7 did not have the solutions in the source literature. Problems 8 and 12 proposed method solutions gave a better efficacy rate than the solutions reported in source papers.

It can also be observed from Table 4.4 that not only the size of the matrix can affect the time needed to solve the problem, but also the number of ones within the matrix. However, that is not the case always. In some cases of similar size matrices, two different instances with similar number of operations could take various times to solve. It was found that the sparser the matrix, the greater the time needed to solve the problem. In other words, when the similarity between parts' operations decreases, it takes more time to find the solution.

Clearly, a significant computational time could be saved without compromising with optimality by reducing the maximum number of groups created to less than half of the number of machines (or \# machines +1 if odd number of machines). In Table 4.5, the first example showed a time saving of almost 2 minutes, The other example showed a saving of almost 31 minutes. Additionally, starting seed does not influence the solution. Solution quality and time for the same problem were the same whether the program starts searching from a local solution close or far to global optimal.

Upper and lower limits of parts and machines within each cell can give great flexibility to the design of the system and can fulfill the requirement and limitation given, however, Table 4.6, explains how the quality of the solution is altered by adding more constraints to the problem.

### 4.7 Review and Elaboration

A proposed binary integer mathematical model to reduce intercellular moves and improve utilization of machines was introduced. The model aimed at improving the result
through using efficacy measure as an objective function. Since the efficacy is nonlinear, a Lagrangean relaxation was used.

The model was applied on different types of matrices, randomly generated problems and benchmark problems from previous literature. Results show that the proposed method performs similarly or outperforms similar cell formation grouping methods with similar objectives in terms of goodness of grouping.

Experimentations were also conducted to reduce the time without altering the quality of the solutions. Results show that it is possible to reduce the time by reducing the search space and limiting the number of cells created. Since this is a parametric model, it can be modified according to the need of the user. Upper and lower bounds on number of machines or parts within each cell can be added if desired.

## CHAPTER V LINEAR PERFORMANCE MEASURE

### 5.1 Introduction

In the absence of performance measures (PM), comparing different grouping methods in CMS is an uneasy task. Therefore, the evaluation of goodness of CF is essential. Several measures were developed with different requirements and suitability according to the criteria being tested. In this chapter, a Linear Performance Measure (LPM) is introduced.

So far, there has not been any known PM which is linear in $E, V$ exists. The introduced LPM is designed to serve as a linear objective function in mathematical models. A linear objective function is faster and easier to compute than a non-linear objective. When compared with other known measure, the new measure considered comparable with respect to: positive value, simplicity, discriminating power and sensitivity. Results show that the proposed measure is comparable to other known measures.

### 5.2 New Performance Measure

In previous section, it was explained how different methods have been developed to obtain the best block diagonal form, and one way to find out which method is performing the best and would be perfectly suitable to obtain a better grouping is by using a PM .

There are several elements that are important to include so that the PM could be considered a comprehensive measure. These are:

- The number of machines and parts that will represent the matrix. The multiplication of number of machines with the number of parts gives us the matrix size. By knowing the size, the proportion of operations to the matrix could be estimated. A higher number or proportion will result in a better value in PM.
- The number of operations within the matrix, or in other words, the number of ones in the matrix. This number is always smaller than the matrix size.
- The number of voids in the diagonal blocks is important because it represents the underutilized machines and is the cause for less production of parts. A low number is preferred in the cells.
- The number of exceptions in the diagonal blocks is important because it represents the intercellular movements of parts and a lower value is desirable.

After considering all of these factors, the new proposed linear performance measure is :

$$
\begin{equation*}
\Omega=1-\left\{\frac{E}{A}+\frac{V}{m n-A}\right\} / 2 \tag{28}
\end{equation*}
$$

Where $E$ is number of exceptions in the solved machine part matrix, $A$ is the number of operations or ones in the matrix, $V$ is the number of voids in the solved matrix, and $m n$ is the multiplication of number of machines and parts in the matrix.

### 5.3 Analysis of the New Performance Measure

The new measure is composed of two significant parts. The first part measures the efficiency of the off-diagonal block which is the ratio of exceptions to the number of operations. The second part measures efficiency of the diagonal block which is the ratio of number of voids to matrix size minus the number of operations.

One important goal to consider in the linear measure is to give equal weights to diagonal and off-diagonal blocks while keeping the measure in linear form to use as an objective function or constraint in mathematical programming models.

When taking a closer look at the LPM, a few characteristics are observed. First, it is noticed that as the matrix size gets larger, the effect of number of exceptions becomes greater than the effect of the number of voids on the measure, and hence, a higher number of operations is required to enhance the performance measure.

Second, it can be noticed that the greater the number of exceptions, the lower the performance measure, and this also applies in case of greater number of voids but to a lower extent.

Third, the LPM measure value is positive and ranges between 0 and 1 . Theorem 1 provides a proof and information on the bounds of the measures

Fourth, this measure is linear. It can be observed that the denominators in both parts of the measure are constant values which must be known a priori in every instance matrix. The number of operations $A$ and the value of $m n$ minus $A$ is also known a priori.

Lastly, this performance measure has no weighting factor or any other factor that should be determined by an expert. The weighting factor value is usually hard to determine and could pose a problem if it was wrongly used.

### 5.4 Comparison with Other Measures

To observe how the new measure performs, a comparison with other measures was conducted. Three known and commonly used measures which are efficiency, efficacy, and CGI are selected to perform a pair wise comparison.

The new measure was compared to other PM, with respect to: positive value, simplicity, discriminating power and sensitivity and then the result were reported. For each criteria, an analysis was used to compare the new LPM with other PMs. Mathematical and statistical tools were used to evaluate the data collected and the measures were compared with each other. The criteria are explained as follows:

1. Simplicity: a PM is simple if it is easy to compute.
2. Discriminating power (Dispersion measure): a PM has a high discriminating power if the values of the PM for various solutions are dispersed over a wider range (Sarker and Khan [2001]).
3. Sensitivity (Response measure): a PM is sensitive if it responds to changes in its parameters
4. Positive: a PM is positive if the value of the PM is always positive in every possible scenario.

To clarify the difference between discriminating power and sensitivity, an example of two PMs with 4 values are used, $\mathrm{PM1}=\{0.1,0.1,0.9,0.9\}$ and PM2 $=\{0.2,0.4,0.6,0.8\}$. Suppose that voids or exceptions are varying in descending order. PM1 covers higher range between 0 and 1 and therefore it has a higher
discriminating power, however, PM2's values varies more within the 0 and 1 range and therefore PM2 is more sensitive to changes in its variables.

### 5.5 Computational Analysis

To conduct a comparison, several statistical tests were used such as: standard deviation, correlation, Analyses of Variance (ANOVA), and paired $t$-test. Also, the reaction of all performance measures to changing variables was simulated to evaluate the sensitivity of each measure.

### 5.5.1 Simplicity

To measure the simplicity factor, two issues where taken into considerations:

1. If the measure is linear
2. Free from parameters that needs to be estimated or obtain.

Table 5. 1 Simplicity factor of different measures

| Feature | Efficiency | Efficacy | GCI | LPM |
| :--- | :---: | :---: | :---: | :---: |
| Linear <br> Free from estimated <br> parameter | 0 | 0 | 1 | 1 |
| Total | 0 | 1 | 1 | 1 |
| Rank | $\mathbf{0}$ | $\mathbf{1}$ | $\mathbf{2}$ | $\mathbf{2}$ |

Table 5.1 shows a ranking scheme to measure the simplicity of the measure. If the feature specified in each row applies, the measure is given a value of 1 , otherwise, a
value of 0 is given. The measure with higher ranking denotes strong representation of simplicity.

It can be realized that efficiency was the least simple measure with value of 0 due to the fact that it is non-linear and it has a $q$ factor that is hard to predict. LPM and GCI were considered simple with value of 2 and efficacy ranked in the middle with value of 1 .

### 5.5.2 Discriminating Power (Dispersion)

Discriminating power is evaluated using the standard deviation of data for each measure, the larger the standard deviation the better the measure is performing. For the factor of discriminating power, a 100 feasible solutions were randomly generated for different size matrices (Appendix B- 100 randomly generated feasible solutions for different size matrices ). The solutions were used to assess the response of each performance measure. The values of $m, n, A, E$, and $V$ were randomly generated according to the following assumptions:

- $n \leq A \leq m n$
- $0 \leq E \leq A$
- $0 \leq V \leq m n-A$

To conduct certain statistical analysis, data should fall under the normal distribution for best result and accurate conclusion. To test the normality of the data, a program known as Stat-fit was used. The data of each measure for the 100 problems were inserted and tested. The program would either accept or reject if the data falls under the required distribution. Result showed a satisfactory result and would not reject the normality of the data for all performance measures test.

To test discriminating power, the four measures were computed for the 100 problems and the mean and standard deviation for data was obtained. Table 5.2 shows the comparison of different performance measures.

Table 5. 2 Comparison of mean and standard deviation for different performance measure

|  | Efficiency | Efficacy | $G C I$ | LPM |
| :--- | :---: | :---: | :---: | :---: |
| Count | 100 | 100 | 100 | 100 |
| Average | 0.63 | 0.50 | 0.76 | 0.63 |
| Variance | 0.03 | 0.05 | 0.02 | 0.03 |
| Std. Dev. | 0.17 | 0.23 | 0.15 | 0.17 |
| Rank |  | $\mathbf{2}$ | $\mathbf{1}$ | $\mathbf{3}$ |



Figure 5. 1 Discriminating power increases as standard deviation increases.

Standard deviation is the best data dispersion measure available, and therefore it was selected for our comparison (Figure 5.1). The tabular results show that the standard deviation of efficacy is the highest, which means that there were more variation in the measure, therefore it had the most discriminating power. Efficiency and LPM came second with the same value of 0.17 and lastly GCI with the lowest value of 0.15 .

### 5.5.3 Sensitivity (Response)

One way to analyze the criteria of sensitivity is by changing one variable at a time and looking to see if the measure is sensitive to changes. Also whether the change is in all cases or there are some conditions where the measure doesn't respond to changes in its parameters.

A response or sensitivity evaluation was performed to observe any reaction on performance measure by setting $m, n$ and $A$ to fixed values, and changing $E$ and $V$. This analysis was conducted on a hundred solution cases to the same size matrix " $40 \times 100$ " (Appendix C- $40 \times 100$ generated matrix). These cases are constructed in a special way to assess the response of each measure to changes in its variables. The hundred cases in Table 5.3 are divided into 4 sections, where each section has 25 cases with two corresponding values, exceptions and voids. If one value is set fixed, then the other value is set to increase or decrease throughout the 25 cases. The fixed number of exceptions or voids is usually set to a value either near the lowest possible value (small) or near the highest possible value (large).

Table 5.3 Status of $E$ and $V$ for set of cases

| Cases | $\mathbf{E}$ | V |
| :--- | :--- | :--- |
| $1-25$ | Large | Changing |
| $25-50$ | Changing | Small |
| $50-75$ | Small | Changing |
| $75-100$ | Changing | Large |



Figure 5. 2 Sensitivity evaluation on $40 \times 100$ matrix

Reaction of each measure in correspondence to the variability of $E$ and $V$ is recorded and plotted in Figures 5.2.

To measure the criteria of sensitivity, the following algorithm is used:

1. Let $y_{i}$ be the performance measure of solution $i$. Arrange the solutions in ascending order $y_{i}$, So, $y_{1} \leq y_{2} \leq y_{3} \ldots \leq y_{n}$
2. Compute the differences, $\delta_{i}=y_{i+1}-y_{i}$
3. Compute the standard deviation, $\sigma_{\delta}$ of differences $\delta_{1}, \delta_{2}, \delta_{3}, \ldots, \delta_{n-1}$.

If there are variations in PM values, $\sigma_{\delta}$ of the differences is high and PM is less sensitive. If otherwise, PM is more sensitive.

Table 5.4 Standard deviation of all measures

|  | Efficiency | Efficacy | GCI | LPM |
| :--- | :---: | :---: | :---: | :---: |
| Std Dev of difference | 0.0057 | 0.0089 | 0.0213 | 0.0052 |
|  |  |  |  |  |
| Rank | 2 | 3 | 4 | $\mathbf{1}$ |

For this criteria, the standard deviations of the difference of values were recorded for the hundred solution cases in Table 5.4. LPM values were more sensitive to changes than any other measure. Efficiency and efficacy came second and third respectively and GCI came last as the least sensitive measure. Although, efficacy and GCI have greater slopes in some cases, they were considered underperforming due to lack of response in the area where $E$ is large and $V$ is changing. GCI also underperformed when $E$ is small and $V$ is changing. LPM correlated more to efficiency than other measures in all cases.

### 5.5.4 Positive Measure

Finally positive value is evaluated by observing the limits of the measure when using best and worst case scenarios. To show that the LPM is always positive, Theorem 1 was used to provide information on lower and upper bound of the measure.

Theorem 1. If $\Omega b$ and $\Omega w$ are the linear performance measure in a best and worst case scenario, respectively, and $\Omega w \leq \Omega \leq \Omega b$, then $0 \leq \Omega \leq 1$.

Proof. For a perfect block-diagonal solution, $E=0$ and $V=0$. So
$\Omega b=1-\left\{\frac{E}{A}+\frac{V}{m n-A}\right\} / 2$

$$
\begin{aligned}
& =1-\left\{\frac{0}{A}+\frac{0}{m n-A}\right\} / 2 \\
& =1-0 / 2 \\
& =1
\end{aligned}
$$

Now, for a worst block-diagonal solution, $E=A$ and $V=m n-A$. So

$$
\begin{aligned}
\Omega w & =1-\left\{\frac{E}{A}+\frac{V}{m n-A}\right\} / 2 \\
& =1-\left\{\frac{A}{A}+\frac{m n-A}{m n-A}\right\} / 2 \\
& =1-\{1+1\} / 2 \\
& =1-\{2\} / 2 \\
& =1-1 \\
& =0
\end{aligned}
$$

Hence, this leads to the conclusion that $0 \leq \Omega \leq 1$ and is always positive.

### 5.5.5 Other Comparison Analysis

Some other analysis were implemented to see if the output values of LPM is consistent with all other measures. The objective of this type of analysis is to test for how different is the LPM from the other compared measure. Three tests were used, correlation, ANOVA, and paired t-test.

ANOVA test can be used to test if there is a significant difference among means. If there is a difference then a paired $t$-test will help to compare two means at a time. The pair wise comparison tells which two means are not significantly different.

A test of variance allows us to see if there is a significant difference among the data of all measures. Hypothesis 1 and Table 5.5 illustrates the result of the test.

## Hypothesis 1:

$H_{0}: \mu_{1}=\mu_{2}=\mu_{3}=\mu_{4}$
$H_{1}$ : At least one mean is different from the others.

Table 5.5 ANOVA test of variance where alpha=0.05

| Source of <br> Variation | SS |  |  |  |  |  |  | Df | MS | F | P-value | F critical |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Between Groups | 3.41 | 3.00 | 1.14 | 33.95 | 0.00 | 2.63 |  |  |  |  |  |  |
| Within Groups | 13.27 | 396.00 | 0.03 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| Total | 16.68 | 399.00 |  |  |  |  |  |  |  |  |  |  |

Decision: Reject the null hypothesis since F $>$ F critical, and conclude that there is a significant difference between all measures and at least one mean is different from the others.

Three t-paired tests were conducted and used to compare if there is any significant difference between the LPM and each of the three other measures. Paired $t$-test is calculated by finding the difference of the values of the two compared items. Hypothesis 2 and Table 5.6 illustrate the result of the test.

## Hypothesis 2:

$H_{0}: \bar{D}=0$
$H_{1}: \bar{D}>0$
Where t critical $=1.66$

Table 5. 6 Paired t-test where alpha=0.05

|  |  |  |  |
| :---: | :---: | :---: | :---: |
| LPM vs. | Efficiency | Efficacy | GCI |
| Mean | -0.00547 | -0.13298 | 0.128241 |
| St dev | 0.065401 | 0.176417 | 0.169715 |
| T | -0.72391 | -6.5282 | 6.543946 |
| P | 0.202612 | 0 | 0 |

Decision (LPM vs. Efficiency): Accept the null hypothesis since $|t|<t$ critical, and conclude that there is no significant difference between LPM and Efficiency.

Decision (LPM vs. Efficacy): Reject the null hypothesis since $|t| \geq t$ critical, and conclude that there is a significant difference between LPM and Efficacy.

Decision (LPM vs. GCI): Reject the null hypothesis since $|\mathrm{t}| \geq \mathrm{t}$ critical, and conclude that there is a significant difference between LPM and GCI

Null hypothesis is either true or false and it represent two judgments based on the evidence presented. The hypothesis was rejected twice and accepted once in the three tests. Decisions show that LPM values are the closest to the values of efficiency. However, the values of LPM are greatly different from efficacy and GCI.

Now a correlation test will also be used to test LPM against other measures. Correlation test will help to understand the significance and nature of the relationship between independent parameters. A correlation value of one means there is strong positive relationship between any two parameters, however, a value of zero means there is no relationship.

The correlation test between all measures revealed that there is a strong correlation between efficiency and LPM while the lowest correlation is between efficacy
and GCI. The correlation between LPM and Efficacy was moderate. Table 5.7 explains the results.

Table 5.7 Correlation test between measures

|  | Efficiency | Efficacy | GCI | LPM |
| :--- | ---: | ---: | ---: | ---: |
| Efficiency | 1 |  |  |  |
| Efficacy | 0.66027 | 1 |  |  |
| GCI | 0.51477 | 0.33583 | 1 |  |
| LPM | 0.92368 | 0.54396 | 0.50346 | 1 |

### 5.5.6 Testing LPM

Eight different size instance matrices were selected from Table 4.8 to compare the run time of the model when using efficacy vs. LPM as an objective function. For each problem, run time for LPM was recorded first and efficacy was recorded second.

Table 5. 8 The difference in runtime of the model when using Efficacy vs. LPM

| No. <br> (literature case) | Size (mxn) | Run Time <br> (LPM) | Run time <br> (Efficacy) |
| :---: | :---: | :---: | :---: |
| $1(1)$ | $5 \times 7$ | 4 | 5 |
| $2(2)$ | $5 \times 6$ | 3 | 4 |
| $3(3)$ | $5 \times 18$ | 4 | 13 |
| $4(4)$ | $8 \times 20$ | 76 | 159 |
| $5(5)$ | $8 \times 20$ | 237 | 582 |
| $6(6)$ | $14 \times 24$ | 6208 | 10430 |
| $7(9)$ | $20 \times 35$ | 1395 | 3913 |
| $8(10)$ | $24 \times 40(1)$ | 893 | 1879 |

Table 5.8 showed that when running small problems, the difference in run time was not significant. As the problem increased in size the time difference was increasing significantly. In some cases it took twice the time to solve the same problem, and in some other cases it took more than twice.

Table 5. 9 Solutions to 50 randomly generated problems on the model when using Efficacy vs. LPM


Randomly generated problems were also used to compare quality of solution and runtime of LPM. The Integer programming model was tested on 50 different problems. For each problem, either LPM or efficacy was used as an objective function. The result were recorded in Table 5.9. Results show that there is no relation between the two runtimes, however, LPM runtime was always less than efficacy. Results also show that in 37 cases, solutions were the same even though the objective was different. The other 13 cases have solutions that are different in term of the number of exceptions, voids. Efficacy is still used to evaluate the results of the two runs.

### 5.6 Discussion and Review

Table 5.10 shows a ranking scheme designed to summarize the comparison among all measures in regards to all factors. Each measure was ranked from 1 to 4 , where 1 denotes a strong representation of the factor, and 4 denotes a weak representation.

Table 5. 10 Summary of comparison of different factors between measures

| Measure | Simplicity | Discriminating <br> power | Sensitivity | Positive |
| :--- | :---: | :---: | :---: | :---: |
| Efficiency | 4 | 2 | 1 | 1 |
| Efficacy | 3 | 1 | 3 | 1 |
| GCI | 1 | 3 | 4 | 1 |
| LPM | 2 | 2 | 1 | 1 |

GCI is given a ranking of 1 for simplicity, while efficiency was given 4 for being the least simple measure. Efficacy was ranked the best in terms of discriminating power, while efficiency and LPM were tied which means that both measure have close discriminating power. In sensitivity, LPM and efficiency were more sensitive to changes
in $E, V$ while GCI was the least sensitive. All measures were positive and were given the same rank

When comparing LPM to efficiency, $q$ in efficiency is hard to find, and could pose a problem in large incidence matrix. LPM is more sensitive than efficacy to changes in its variables when the number of exception is large. LPM is also linear and with only two variable, $E$ and $V$ and three constants $m, n$ and $A$. Therefore it is simple to compute.

The performance measure can measure the goodness of the solution for various grouping CF methods. It determines how perfect is to produce a standard block diagonal matrix. Different factors can contribute to the goodness of the results or the result value of the measure. Some are a result of the measure used and some others are given inputs.

A new LPM was introduced, and the purpose of the measure was clarified. The LPM was compared to other existing measures. The comparative study showed that the new measure is comparable to existing measures. The analysis also showed that the LPM has a higher sensitivity, and positive value and is simple to use for computation.

The improvement of this measure over other measures that it is linear and it gave equal weight to in diagonal and off-diagonal block efficiency. It also gave reasonable values of performance when tested on solved incidence matrices that are comparable to other existing PMs.

## CHAPTER VI

## CONCLUSIONS AND RECOMMENDATIONS

### 6.1 This Work

In the past few decades, there has been an increasing demand on minimizing waste and efficient productivity. New ways of operations had immerged to improve the work place environment with factors such as team work and job satisfaction. Flexibilities in production and management decentralization are also becoming important for the survival of complex and large systems. CMS has come out as a promising technique to satisfy the need for more efficient manufacturing systems. The first step in developing CMS is by forming the cells using CF methodology. The contributions of this research lie in developing a mathematical model to form cells, and improve the solution performance by introducing a new performance measure and comparing it to previously used measures

Throughout our review of existing work in this field, there is little effort to solve the machine part CF problem optimally, and rather they concentrated on solving large size problems with good feasible solution. Additionally, previous work tended to solve the problem in two or more steps rather than considering all factors when solving in one step. Finally, although the various mostly used goodness of grouping measure are good and reliable, they have some drawbacks besides their advantages.

When designing a CMS, there are few factors that ought to be considered. Two important factors are machine utilizations and the movement of parts. The proposed CF methodology considers these two factors and offer optimal solutions to systems with different sizes. The developed model is capable of forming cells into groups and parts into families simultaneously using mathematical programming. It also finds the optimal
number of cells required for an efficient system. The model is designed so that the performance measure, efficacy, is considered as an objective function.

The solution of the model will result in a similar or higher value than other methods according to major performance measures. Therefore, our method is comparable to major cell formation problem methodologies with similar objectives when it comes to obtaining a global optimal solution.

The mixed integer programming model presented in this research provides the user with a valuable CF design tool. The model can be used in real-life scenarios and can be modified by adding or removing constraints.

The model was tested on several problems with various size matrices, some were previously solved using different methods. Despite the fact that the data used were of certain size, the model can accept various sizes of data. The results from our experiments performed prove that the model can successfully be applied as a design tool.

In the second part a linear performance measure is presented along with comparison of mostly used measures. The new measure overcomes some drawbacks of previous measures. It also can be used as an objective function in the model to reduce the time to solve the problem

The LPM was compared to other measures using four criteria: sensitivity, discriminating power, simplicity, and if positive. Sensitivity was tested by generating data with various variable's values and observing the effect of the change on all performance measures. Discriminating power was tested by generating 100 solutions to different size problems, and evaluating the computed performance measures for the 100 problems. Statistical tests such as ANOVA, paired t -test and correlation were used to
evaluate the data and compare difference in means Taking the limits was used to find the boundaries of the measure and examining if the measure is positive

Results show that the new performance measure is comparable to other measures in terms of the four criteria. The results also show that LPM is closest to efficiency and there is a strong correlation between the two. The LPM should improve the performance of the integer programming model if used as an objective function

### 6.2 Scope of future work

A future extension of this work would be to consider the next phase of CM design. The second phase consider the design of each individual cell. Therefore, typical issues in this problem are: job routing and sequencing jobs within the cells, machine scheduling, set up time, and operation cost, and machine capacity.

However, it is believed that the consideration of the second phase is beyond the scope of this thesis and it will require substantial time and cannot be achieved within the time frame of this thesis. For this reason, the second phase is considered as a future research work.

## APPENDICES

## Appendix A

## LINDO CODE

```
! code written by: Talal Kattan;
! Machine-part cell formation problem;
! Date 28 Sep, 2006;
```



```
MODEL:
SETS:
cell/g1..g3/:gk--->!;change !gk=1 if cell formed, 0 otherwise;
machine/m1..m5--->!;/change
m1 means machine 1;
part/p1..p7--->!;/change;
Matx (machine, part): Mx ;
x (machine, cell): xik; !xik=1 if machine i assigned to
cell k, 0 otherwise;
y (cell, part): ykj ; !ykj=1 if part j assigned to cell
k, 0 otherwise;
exc (machine, cell, part): eikj; !eikj=1 if exeption, 0 otherwise ;
vid (machine, cell, part): vikj; !vikj=1 if void, o otherwise;
ENDSETS
```

```
lobjective function;
```

lobjective function;
]OBJECTIVE [MIN) = E+lamda*V;(

```
]OBJECTIVE [MIN) = E+lamda*V;(
```

!initialize variables;
@FOR)cell@ :BIN)gk; (;
@FOR) machine@:FOR)cell@:BIN)xik; (; (
@FOR) part@ : FOR) cell@ :BIN)ykj; (; ( (
Ae - SUM) Matx (I,J) : Mx(I,J) \#EQ\#1)=0;
!cell constraints;
G@ - SUM) cell(K): $\mathrm{gk}(\mathrm{K}))=0$;
@FOR) cell(K): m*gk(K@ - (SUM) machine(I): xik(I,K))>=0; (
@FOR) cell(K@ : (SUM) machine(I): xik(I, K)) - gk(K)>=0; (
@FOR) cell(K): $n * g k(K @-(S U M) \operatorname{part}(J): y k j(K, J))>=0$; (
@FOR)cell(K@ : (SUM) part(J): ykj(K,J)) - gk(K) >=0; (
!machine constraints;
@FOR) machine (I@ : (SUM) cell (K):xik(I,K))=1; (
!part constraints;
@FOR) part (J@ :(SUM) cell (K):ykj(K,J))=1; (
!exception constraints;
@FOR)cell(K@:(FOR)Matx(I,J)| Mx(I,J) \#EQ\#1: -
$\operatorname{xik}(I, K)+y k j(K, J)+e i k j(I, K, J)>=0 ;(($
@FOR) cell(K@:(FOR)Matx(I, J)| Mx(I, J) \#EQ\#0: eikj(I, K, J)=0; ( (
E@ -SUM) cell(K@ :(SUM) machine(I@ :(SUM)part(J): eikj(I,K,J));))=0;

```
!void constraints;
@FOR)cell(K@ :(FOR)Matx(I,J)| Mx(I,J) #EQ#0: xik(I,K)+ykj(K,J)-
vikj(I,K,J)<=1;((
@FOR)cell(K@ :(FOR)Matx(I,J)| Mx(I,J)#EQ#1: vikj(I,K,J)=0;((
V@ -SUM)cell(K@ :(SUM)machine(I@ :(SUM)part(J): vikj(I,K,J));))=0;
!genaral;
@FOR)cell(K@ :(FOR)machine(I@ :(FOR)part(J): eikj(I,K,J)>=0;((;)
@FOR)cell(K@ :(FOR)machine(I@ :(FOR)part(J): vikj(I,K,J)>=0;((;)
DATA:
```

```
    lamda@ = POINTER;(1)
@ POINTER = (2)E;
@ POINTER = (3)V;
@ POINTER = (4)OBJECTIVE ;
@ POINTER@ = (5)STATUS; ()
@ POINTER = (6)A;
(d POINTER = (7)G;
!given constants;
m=5--->! ;change # of machine;
n=7--->! ; change # of part;
```

!Given Matrix;
$M \mathrm{x}=\begin{array}{lllllll}0 & 1 & 0 & 1 & 1 & 1 & 0\end{array}$
$\begin{array}{lllllll}1 & 0 & 1 & 0 & 0 & 0 & 0\end{array}$
$\begin{array}{lllllll}1 & 0 & 1 & 0 & 0 & 0 & 1\end{array}$
01011010
$1000001 ;-->$ ! change;
ENDDATA
END

```
VB Script
' code written by: Talal Kattan;
' Machine-part cell formation problem;
' Date 28 Sep, 2006;
| =========================================%;
Private Sub solve_Click()
' Calls the LINGO DLL to solve the machine part problem
' model in MACHINE.LNG. lamda is taken from the user
'Stores start time in variable "StartTime"
Dim StartTime As Double, EndTime As Double
StartTime = Timer
' Get lamda from the dialog box
Dim varLamda As Double
varLamda = lamda.Text
' Create the LINGO environment object
    Dim pLINGO As Long
    pLINGO = LScreateEnvLng()
    If pLINGO = 0 Then
        MsgBox ("Unable to create LINGO Environment.")
        GoTo FinalExit
    End If
' Open LINGO's log file
    Dim nError As Long
    nError = LSopenLogFileLng(pLINGO, "\LINGO.log")
    If nError <> 0 Then GoTo ErrorExit
' Pass memory transfer pointers to LINGO
    Dim varExcept As Double, varVoid As Double
    Dim varObject As Double, dStatus As Double
    Dim varOper As Double, varNoCell As Double
' Build LINGO's command script (commands
' are terminated with an ASCII 10
    Dim cScript As String
Dim Counter As Integer
Counter = 0
' Loop for best lamda
'---------------------
    Do
    ' Counter
    If Counter <> 0 Then
                varLamda = lamdanod
```

```
    End If
    Counter = Counter + 1
    cScript = ""
' @POINTER(1)
    nError = LSsetPointerLng(pLINGO, varLamda, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POINTER(2)
    nError = LSsetPointerLng(pLINGO, varExcept, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POINTER(3)
    nError = LSsetPointerLng(pLINGO, varVoid, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POINTER(4)
    nError = LSsetPointerLng(pLINGO, varObject, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POTNTER(5)
    nError = LSsetPointerLng(pLINGO, dStatus, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POINTER(6)
    nError = LSsetPointerLng(pLINGO, varOper, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
- @POINTER(7)
    nError = LSsetPointerLng(pLINGO, varNoCell, nPointersNow)
    If nError <> 0 Then GoTo ErrorExit
        ' Causes LINGO to echo input
            cScript = "SET ECHOIN 1" & Chr(10)
            ' Read in the model file
                    cScript = cScript &
                    "TAKE \LINGO9\Samplès\machine.lng" & Chr(10)
            r Solve the model
                cScript = cScript & "GO" & Chr(10)
            ' Quit LINGO DLL
            cScript = cScript & "QUIT" & Chr(10)
            ' Mark end of script with a null byte
```

```
    cScript = cScript & Chr(0)
    ' Run the script
        dStatus = -1#
    nError = LSexecuteScriptLng(pLINGO, cScript)
    ' Close the log file
    LScloseLogFileLng (pLINGO)
    - Problems?
        If nError > 0 Or dStatus <> 0 Then
                MsgBox ("Unable to solve!")
                GoTo ErrorExit
    End If
    ' Place Start values in dialog box
        exception.Caption = varExcept
    ' Place On Duty values in dialog box
        void.Caption = varVoid
    ' Place best Lamda value in dialog box
        mtrxSize.Caption = Counter
    ' Place On Duty values in dialog box
        operation.Caption = varOper
    ' Place On Duty values in dialog box
        noCell.Caption = varNoCell
    ' Place Objective Value in dialog box
        objective.Caption = Format(varObject, "00.00")
    ' Place best Lamda value in dialog box
        bstLamda.Caption = Format(lamdanod, "0.00")
    - Step 2 in the algorithm
        lamdanod = (varOper - varExcept) / (varOper + varVoid)
    Loop Until lamdanod = varLamda
    '------------------
    ' End of lamda loop
    LSdeleteEnvLng (pLINGO)
    GoTo FinalExit:
ErrorExit:
    MsgBox ("LINGO Error Code: " & nError&)
    LSdeleteEnvLng (pIINGO)
FinalExit:
```

```
'Stores end time in variable "EndTime"
EndTime = Timer
'time.Caption = Format$(EndTime - StartTime, "0000.00")
time.Caption = Format$((EndTime - StartTime) / 86400#, "hh:mm:ss")
End Sub
Private Sub exit Click()
    End
End Sub
```


## Appendix B

100 randomly generated feasible solutions for different size matrices

| $\eta$ | $\Gamma$ | GCI | LPM | Efficienc vs <br> New PFM | Efficacy vs New PFM | GCI vs New PFM |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0.39 | 0.22 | 0.52 | 0.39 | 0.00 | -0.17 | 0.13 |
| 0.39 | 0.49 | 0.59 | 0.33 | 0.05 | 0.16 | 0.26 |
| 0.60 | 0.39 | 0.87 | 0.58 | 0.03 | -0.19 | 0.29 |
| 0.69 | 0.86 | 0.88 | 0.81 | -0.12 | 0.05 | 0.08 |
| 0.84 | 0.82 | 0.99 | 0.57 | 0.27 | 0.25 | 0.42 |
| 0.74 | 0.50 | 0.72 | 0.76 | -0.02 | -0.26 | -0.04 |
| 0.53 | 0.08 | 0.70 | 0.70 | -0.16 | -0.62 | 0.00 |
| 0.32 | 0.11 | 0.51 | 0.32 | 0.00 | -0.21 | 0.19 |
| 0.61 | 0.62 | 0.69 | 0.65 | -0.05 | -0.03 | 0.03 |
| 0.85 | 0.93 | 0.96 | 0.86 | -0.01 | 0.07 | 0.10 |
| 0.87 | 0.76 | 0.97 | 0.83 | 0.03 | -0.08 | 0.13 |
| 0.77 | 0.84 | 0.89 | 0.81 | -0.04 | 0.03 | 0.08 |
| 0.36 | 0.21 | 0.55 | 0.37 | -0.01 | -0.16 | 0.18 |
| 0.70 | 0.47 | 0.84 | 0.73 | -0.03 | -0.27 | 0.11 |
| 0.85 | 0.50 | 0.57 | 0.77 | 0.08 | -0.26 | -0.20 |
| 0.78 | 0.85 | 0.98 | 0.61 | 0.17 | 0.24 | 0.37 |
| 0.41 | 0.59 | 0.75 | 0.41 | -0.01 | 0.18 | 0.34 |
| 0.68 | 0.49 | 0.67 | 0.68 | 0.00 | -0.19 | -0.01 |
| 0.43 | 0.15 | 0.66 | 0.42 | 0.01 | -0.27 | 0.23 |
| 0.81 | 0.85 | 0.88 | 0.87 | -0.06 | -0.02 | 0.01 |
| 0.29 | 0.32 | 0.53 | 0.32 | -0.02 | 0.00 | 0.21 |
| 0.62 | 0.31 | 0.92 | 0.62 | 0.00 | -0.31 | 0.30 |
| 0.49 | 0.07 | 0.70 | 0.48 | 0.02 | -0.41 | 0.23 |
| 0.74 | 0.45 | 0.59 | 0.73 | 0.01 | -0.28 | -0.14 |
| 0.53 | 0.26 | 0.69 | 0.54 | -0.01 | -0.28 | 0.15 |
| 0.62 | 0.24 | 0.91 | 0.81 | -0.19 | -0.57 | 0.10 |
| 0.19 | 0.29 | 0.58 | 0.30 | -0.10 | -0.01 | 0.29 |
| 0.69 | 0.47 | 0.95 | 0.64 | 0.06 | -0.17 | 0.31 |
| 0.40 | 0.22 | 0.89 | 0.47 | -0.07 | -0.25 | 0.42 |
| 0.74 | 0.53 | 0.55 | 0.74 | 0.00 | -0.21 | -0.19 |
| 0.53 | 0.28 | 0.69 | 0.54 | 0.00 | -0.26 | 0.16 |
| 0.71 | 0.62 | 0.91 | 0.65 | 0.07 | -0.02 | 0.26 |
| 0.71 | 0.59 | 0.89 | 0.67 | 0.05 | -0.08 | 0.23 |
| 0.45 | 0.11 | 0.64 | 0.42 | 0.03 | -0.31 | 0.22 |
| 0.61 | 0.38 | 0.53 | 0.61 | 0.00 | -0.23 | -0.08 |
| 0.80 | 0.68 | 0.97 | 0.67 | 0.13 | 0.01 | 0.30 |
| 0.58 | 0.54 | 0.73 | 0.58 | 0.01 | -0.03 | 0.16 |
| 0.51 | 0.39 | 0.68 | 0.51 | 0.00 | -0.11 | 0.17 |
| 0.64 | 0.29 | 0.67 | 0.75 | -0.10 | -0.46 | -0.08 |
| 0.67 | 0.52 | 0.62 | 0.68 | 0.00 | -0.15 | -0.05 |
| 0.52 | 0.74 | 0.94 | 0.51 | 0.01 | 0.23 | 0.43 |


| 0.42 | 0.13 | 0.69 | 0.42 | 0.00 | -0.29 | 0.28 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0.79 | 0.57 | 1.00 | 0.82 | -0.04 | -0.25 | 0.18 |
| 0.73 | 0.87 | 0.89 | 0.85 | -0.12 | 0.01 | 0.03 |
| 0.62 | 0.39 | 0.69 | 0.63 | -0.01 | -0.24 | 0.05 |
| 0.82 | 0.64 | 0.99 | 0.76 | 0.05 | -0.12 | 0.23 |
| 0.77 | 0.69 | 0.95 | 0.68 | 0.09 | 0.01 | 0.27 |
| 0.46 | 0.47 | 0.82 | 0.48 | -0.02 | -0.01 | 0.34 |
| 0.77 | 0.76 | 0.81 | 0.81 | -0.04 | -0.05 | 0.00 |
| 0.71 | 0.57 | 0.73 | 0.71 | 0.00 | -0.14 | 0.03 |
| 0.50 | 0.30 | 0.99 | 0.50 | 0.00 | -0.20 | 0.49 |
| 0.56 | 0.54 | 0.61 | 0.59 | -0.03 | -0.04 | 0.02 |
| 0.53 | 0.27 | 0.82 | 0.52 | 0.00 | -0.25 | 0.30 |
| 0.54 | 0.48 | 0.86 | 0.52 | 0.02 | -0.03 | 0.34 |
| 0.38 | 0.15 | 0.70 | 0.40 | -0.03 | -0.25 | 0.29 |
| 0.88 | 0.92 | 0.93 | 0.95 | -0.07 | -0.03 | -0.02 |
| 0.71 | 0.39 | 0.70 | 0.79 | -0.08 | -0.40 | -0.09 |
| 0.77 | 0.47 | 0.51 | 0.72 | 0.05 | -0.25 | -0.21 |
| 0.95 | 0.94 | 0.96 | 0.96 | -0.01 | -0.02 | 0.00 |
| 0.69 | 0.76 | 0.97 | 0.57 | 0.13 | 0.19 | 0.40 |
| 0.65 | 0.40 | 0.96 | 0.57 | 0.08 | -0.17 | 0.39 |
| 0.39 | 0.35 | 0.75 | 0.43 | -0.04 | -0.08 | 0.31 |
| 0.61 | 0.52 | 0.63 | 0.62 | -0.01 | -0.10 | 0.00 |
| 0.78 | 0.72 | 0.73 | 0.84 | -0.07 | -0.13 | -0.11 |
| 0.82 | 0.69 | 0.94 | 0.79 | 0.03 | -0.10 | 0.15 |
| 0.91 | 0.96 | 0.98 | 0.92 | -0.01 | 0.04 | 0.06 |
| 0.62 | 0.43 | 0.77 | 0.61 | 0.01 | -0.18 | 0.16 |
| 0.53 | 0.50 | 0.54 | 0.55 | -0.02 | -0.05 | -0.01 |
| 0.58 | 0.24 | 0.75 | 0.63 | -0.05 | -0.39 | 0.12 |
| 0.58 | 0.19 | 0.90 | 0.66 | -0.08 | -0.46 | 0.25 |
| 0.55 | 0.57 | 0.72 | 0.55 | 0.00 | 0.01 | 0.16 |
| 0.60 | 0.23 | 0.73 | 0.72 | -0.12 | -0.49 | 0.02 |
| 0.41 | 0.43 | 0.52 | 0.36 | 0.05 | 0.08 | 0.16 |
| 0.81 | 0.54 | 0.74 | 0.83 | -0.02 | -0.29 | -0.09 |
| 0.60 | 0.62 | 0.95 | 0.52 | 0.07 | 0.10 | 0.43 |
| 0.69 | 0.58 | 0.61 | 0.74 | -0.04 | -0.16 | -0.13 |
| 0.63 | 0.40 | 0.57 | 0.63 | 0.00 | -0.23 | -0.06 |
| 0.76 | 0.54 | 0.64 | 0.75 | 0.02 | -0.20 | -0.11 |
| 0.94 | 0.92 | 0.98 | 0.91 | 0.03 | 0.01 | 0.07 |
| 0.73 | 0.69 | 0.92 | 0.65 | 0.08 | 0.04 | 0.28 |
| 0.45 | 0.30 | 0.72 | 0.47 | -0.01 | -0.16 | 0.26 |
| 0.21 | 0.32 | 0.58 | 0.29 | -0.09 | 0.02 | 0.29 |
| 0.53 | 0.23 | 0.59 | 0.54 | -0.01 | -0.31 | 0.05 |
| 0.72 | 0.72 | 0.73 | 0.83 | -0.11 | -0.11 | -0.10 |
| 0.84 | 0.80 | 0.94 | 0.79 | 0.05 | 0.01 | 0.15 |
| 0.76 | 0.61 | 0.74 | 0.76 | 0.00 | -0.15 | -0.02 |
| 0.72 | 0.33 | 0.52 | 0.74 | -0.02 | -0.41 | -0.22 |
| 0.92 | 0.63 | 0.69 | 0.84 | 0.08 | -0.21 | -0.15 |
| 0.87 | 0.90 | 0.92 | 0.92 | -0.05 | -0.02 | 0.00 |
| 0.58 | 0.69 | 0.74 | 0.64 | -0.06 | 0.05 | 0.10 |
| 0.75 | 0.50 | 0.60 | 0.73 | 0.02 | -0.23 | -0.13 |


| 0.47 | 0.52 | 0.58 | 0.44 | 0.03 | 0.07 | 0.13 |
| :--- | :--- | :--- | :--- | :---: | :---: | :---: |
| 0.59 | 0.21 | 0.93 | 0.62 | -0.03 | -0.40 | 0.31 |
| 0.53 | 0.34 | 0.56 | 0.53 | 0.00 | -0.19 | 0.03 |
| 0.74 | 0.50 | 0.78 | 0.78 | -0.04 | -0.29 | 0.00 |
| 0.76 | 0.76 | 0.91 | 0.71 | 0.05 | 0.05 | 0.20 |
| 0.36 | 0.49 | 0.63 | 0.34 | 0.02 | 0.14 | 0.29 |
| 0.23 | 0.24 | 0.63 | 0.34 | -0.11 | -0.10 | 0.29 |
| 0.78 | 0.82 | 0.86 | 0.84 | -0.06 | -0.03 | 0.01 |
| 0.48 | 0.20 | 0.72 | 0.48 | 0.00 | -0.27 | 0.25 |

## Appendix C

## $40 \times 100$ generated matrix

| No | $\mathrm{e}(\mathrm{A})$ | $\mathrm{e} 0(\mathrm{E})$ | $\mathrm{ev}(\mathrm{V})$ | $\mathrm{m} * \mathrm{n}$ | q | Efficiency | Efficacy | GCI | LPM |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 1016 | 1000 | 2400 | 4000 | 0.5 | 0.19 | 0.00 | 0.02 | 0.11 |
| 2 | 1016 | 1000 | 2300 | 4000 | 0.5 | 0.21 | 0.00 | 0.02 | 0.12 |
| 3 | 1016 | 1000 | 2200 | 4000 | 0.5 | 0.22 | 0.00 | 0.02 | 0.14 |
| 4 | 1016 | 1000 | 2100 | 4000 | 0.5 | 0.24 | 0.01 | 0.02 | 0.16 |
| 5 | 1016 | 1000 | 2000 | 4000 | 0.5 | 0.25 | 0.01 | 0.02 | 0.17 |
| 6 | 1016 | 1000 | 1900 | 4000 | 0.5 | 0.26 | 0.01 | 0.02 | 0.19 |
| 7 | 1016 | 1000 | 1800 | 4000 | 0.5 | 0.28 | 0.01 | 0.02 | 0.21 |
| 8 | 1016 | 1000 | 1700 | 4000 | 0.5 | 0.29 | 0.01 | 0.02 | 0.22 |
| 9 | 1016 | 1000 | 1600 | 4000 | 0.5 | 0.30 | 0.01 | 0.02 | 0.24 |
| 10 | 1016 | 1000 | 1500 | 4000 | 0.5 | 0.30 | 0.01 | 0.02 | 0.26 |
| 11 | 1016 | 1000 | 1400 | 4000 | 0.5 | 0.31 | 0.01 | 0.02 | 0.27 |
| 12 | 1016 | 1000 | 1300 | 4000 | 0.5 | 0.32 | 0.01 | 0.02 | 0.29 |
| 13 | 1016 | 1000 | 1200 | 4000 | 0.5 | 0.33 | 0.01 | 0.02 | 0.31 |
| 14 | 1016 | 1000 | 1100 | 4000 | 0.5 | 0.33 | 0.01 | 0.02 | 0.32 |
| 15 | 1016 | 1000 | 1000 | 4000 | 0.5 | 0.34 | 0.01 | 0.02 | 0.34 |
| 16 | 1016 | 1000 | 900 | 4000 | 0.5 | 0.35 | 0.01 | 0.02 | 0.36 |
| 17 | 1016 | 1000 | 800 | 4000 | 0.5 | 0.35 | 0.01 | 0.02 | 0.37 |
| 18 | 1016 | 1000 | 700 | 4000 | 0.5 | 0.36 | 0.01 | 0.02 | 0.39 |
| 19 | 1016 | 1000 | 600 | 4000 | 0.5 | 0.37 | 0.01 | 0.02 | 0.41 |
| 20 | 1016 | 1000 | 500 | 4000 | 0.5 | 0.37 | 0.01 | 0.02 | 0.42 |
| 21 | 1016 | 1000 | 400 | 4000 | 0.5 | 0.38 | 0.01 | 0.02 | 0.44 |
| 22 | 1016 | 1000 | 300 | 4000 | 0.5 | 0.39 | 0.01 | 0.02 | 0.46 |
| 23 | 1016 | 1000 | 200 | 4000 | 0.5 | 0.40 | 0.01 | 0.02 | 0.47 |
| 24 | 1016 | 1000 | 100 | 4000 | 0.5 | 0.44 | 0.01 | 0.02 | 0.49 |
| 25 | 1016 | 1000 | 0 | 4000 | 0.5 | 0.87 | 0.02 | 0.02 | 0.51 |
| 26 | 1016 | 1000 | 0 | 4000 | 0.5 | 0.87 | 0.02 | 0.02 | 0.51 |
| 27 | 1016 | 950 | 0 | 4000 | 0.5 | 0.88 | 0.06 | 0.06 | 0.53 |
| 28 | 1016 | 900 | 0 | 4000 | 0.5 | 0.88 | 0.11 | 0.11 | 0.56 |
| 29 | 1016 | 850 | 0 | 4000 | 0.5 | 0.89 | 0.16 | 0.16 | 0.58 |
| 30 | 1016 | 800 | 0 | 4000 | 0.5 | 0.89 | 0.21 | 0.21 | 0.61 |
| 31 | 1016 | 750 | 0 | 4000 | 0.5 | 0.90 | 0.26 | 0.26 | 0.63 |
| 32 | 1016 | 700 | 0 | 4000 | 0.5 | 0.90 | 0.31 | 0.31 | 0.66 |
| 33 | 1016 | 600 | 0 | 4000 | 0.5 | 0.92 | 0.41 | 0.41 | 0.70 |
| 34 | 1016 | 500 | 0 | 4000 | 0.5 | 0.93 | 0.51 | 0.51 | 0.75 |
| 35 | 1016 | 400 | 0 | 4000 | 0.5 | 0.94 | 0.61 | 0.61 | 0.80 |
| 36 | 1016 | 350 | 0 | 4000 | 0.5 | 0.95 | 0.66 | 0.66 | 0.83 |
| 37 | 1016 | 325 | 0 | 4000 | 0.5 | 0.95 | 0.68 | 0.68 | 0.84 |
| 38 | 1016 | 300 | 0 | 4000 | 0.5 | 0.95 | 0.70 | 0.70 | 0.85 |
| 39 | 1016 | 275 | 0 | 4000 | 0.5 | 0.96 | 0.73 | 0.73 | 0.86 |
| 40 | 1016 | 250 | 0 | 4000 | 0.5 | 0.96 | 0.75 | 0.75 | 0.88 |
| 41 | 1016 | 225 | 0 | 4000 | 0.5 | 0.96 | 0.78 | 0.78 | 0.89 |
| 42 | 1016 | 200 | 0 | 4000 | 0.5 | 0.97 | 0.80 | 0.80 | 0.90 |


| 43 | 1016 | 175 | 0 | 4000 | 0.5 | 0.97 | 0.83 | 0.83 | 0.91 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 44 | 1016 | 150 | 0 | 4000 | 0.5 | 0.98 | 0.85 | 0.85 | 0.93 |
| 45 | 1016 | 125 | 0 | 4000 | 0.5 | 0.98 | 0.88 | 0.88 | 0.94 |
| 46 | 1016 | 100 | 0 | 4000 | 0.5 | 0.98 | 0.90 | 0.90 | 0.95 |
| 47 | 1016 | 75 | 0 | 4000 | 0.5 | 0.99 | 0.93 | 0.93 | 0.96 |
| 48 | 1016 | 50 | 0 | 4000 | 0.5 | 0.99 | 0.95 | 0.95 | 0.98 |
| 49 | 1016 | 25 | 0 | 4000 | 0.5 | 1.00 | 0.98 | 0.98 | 0.99 |
| 50 | 1016 | 0 | 0 | 4000 | 0.5 | 1.00 | 1.00 | 1.00 | 1.00 |
| 51 | 1016 | 0 | 0 | 4000 | 0.5 | 1.00 | 1.00 | 1.00 | 1.00 |
| 52 | 1016 | 0 | 100 | 4000 | 0.5 | 0.96 | 0.91 | 1.00 | 0.98 |
| 53 | 1016 | 0 | 200 | 4000 | 0.5 | 0.92 | 0.84 | 1.00 | 0.97 |
| 54 | 1016 | 0 | 300 | 4000 | 0.5 | 0.89 | 0.77 | 1.00 | 0.95 |
| 55 | 1016 | 0 | 400 | 4000 | 0.5 | 0.86 | 0.72 | 1.00 | 0.93 |
| 56 | 1016 | 0 | 500 | 4000 | 0.5 | 0.84 | 0.67 | 1.00 | 0.92 |
| 57 | 1016 | 0 | 600 | 4000 | 0.5 | 0.81 | 0.63 | 1.00 | 0.90 |
| 58 | 1016 | 0 | 700 | 4000 | 0.5 | 0.80 | 0.59 | 1.00 | 0.88 |
| 59 | 1016 | 0 | 800 | 4000 | 0.5 | 0.78 | 0.56 | 1.00 | 0.87 |
| 60 | 1016 | 0 | 900 | 4000 | 0.5 | 0.77 | 0.53 | 1.00 | 0.85 |
| 61 | 1016 | 0 | 1000 | 4000 | 0.5 | 0.75 | 0.50 | 1.00 | 0.83 |
| 62 | 1016 | 0 | 1100 | 4000 | 0.5 | 0.74 | 0.48 | 1.00 | 0.82 |
| 63 | 1016 | 0 | 1200 | 4000 | 0.5 | 0.73 | 0.46 | 1.00 | 0.80 |
| 64 | 1016 | 0 | 1300 | 4000 | 0.5 | 0.72 | 0.44 | 1.00 | 0.78 |
| 65 | 1016 | 0 | 1400 | 4000 | 0.5 | 0.71 | 0.42 | $\cdot 1.00$ | 0.77 |
| 66 | 1016 | 0 | 1500 | 4000 | 0.5 | 0.70 | 0.40 | 1.00 | 0.75 |
| 67 | 1016 | 0 | 1600 | 4000 | 0.5 | 0.69 | 0.39 | 1.00 | 0.73 |
| 68 | 1016 | 0 | 1700 | 4000 | 0.5 | 0.69 | 0.37 | 1.00 | 0.72 |
| 69 | 1016 | 0 | 1800 | 4000 | 0.5 | 0.68 | 0.36 | 1.00 | 0.70 |
| 70 | 1016 | 0 | 1900 | 4000 | 0.5 | 0.67 | 0.35 | 1.00 | 0.68 |
| 71 | 1016 | 0 | 2000 | 4000 | 0.5 | 0.67 | 0.34 | 1.00 | 0.66 |
| 72 | 1016 | 0 | 2100 | 4000 | 0.5 | 0.66 | 0.33 | 1.00 | 0.65 |
| 73 | 1016 | 0 | 2200 | 4000 | 0.5 | 0.66 | 0.32 | 1.00 | 0.63 |
| 74 | 1016 | 0 | 2300 | 4000 | 0.5 | 0.65 | 0.31 | 1.00 | 0.61 |
| 75 | 1016 | 0 | 2400 | 4000 | 0.5 | 0.65 | 0.30 | 1.00 | 0.60 |
| 76 | 1016 | 0 | 2400 | 4000 | 0.5 | 0.65 | 0.30 | 1.00 | 0.60 |
| 77 | 1016 | 25 | 2400 | 4000 | 0.5 | 0.63 | 0.29 | 0.98 | 0.59 |
| 78 | 1016 | 50 | 2400 | 4000 | 0.5 | 0.60 | 0.28 | 0.95 | 0.57 |
| 79 | 1016 | 75 | 2400 | 4000 | 0.5 | 0.58 | 0.28 | 0.93 | 0.56 |
| 80 | 1016 | 100 | 2400 | 4000 | 0.5 | 0.57 | 0.27 | 0.90 | 0.55 |
| 81 | 1016 | 125 | 2400 | 4000 | 0.5 | 0.55 | 0.26 | 0.88 | 0.54 |
| 82 | 1016 | 150 | 2400 | 4000 | 0.5 | 0.53 | 0.25 | 0.85 | 0.52 |
| 83 | 1016 | 175 | 2400 | 4000 | 0.5 | 0.51 | 0.25 | 0.83 | 0.51 |
| 84 | 1016 | 200 | 2400 | 4000 | 0.5 | 0.50 | 0.24 | 0.80 | 0.50 |
| 85 | 1016 | 225 | 2400 | 4000 | 0.5 | 0.48 | 0.23 | 0.78 | 0.49 |
| 86 | 1016 | 250 | 2400 | 4000 | 0.5 | 0.47 | 0.22 | 0.75 | 0.47 |
| 87 | 1016 | 275 | 2400 | 4000 | 0.5 | 0.46 | 0.22 | 0.73 | 0.46 |
| 88 | 1016 | 300 | 2400 | 4000 | 0.5 | 0.45 | 0.21 | 0.70 | 0.45 |
| 89 | 1016 | 325 | 2400 | 4000 | 0.5 | 0.43 | 0.20 | 0.68 | 0.44 |
| 90 | 1016 | 350 | 2400 | 4000 | 0.5 | 0.42 | 0.19 | 0.66 | 0.43 |
| 91 | 1016 | 400 | 2400 | 4000 | 0.5 | 0.40 | 0.18 | 0.61 | 0.40 |
| 92 | 1016 | 500 | 2400 | 4000 | 0.5 | 0.36 | 0.15 | 0.51 | 0.35 |


| 93 | 1016 | 600 | 2400 | 4000 | 0.5 | 0.32 | 0.12 | 0.41 | 0.30 |
| :---: | :---: | :---: | :---: | :---: | :---: | :--- | :--- | :--- | :--- |
| 94 | 1016 | 700 | 2400 | 4000 | 0.5 | 0.29 | 0.09 | 0.31 | 0.25 |
| 95 | 1016 | 750 | 2400 | 4000 | 0.5 | 0.27 | 0.08 | 0.26 | 0.23 |
| 96 | 1016 | 800 | 2400 | 4000 | 0.5 | 0.25 | 0.06 | 0.21 | 0.20 |
| 97 | 1016 | 850 | 2400 | 4000 | 0.5 | 0.24 | 0.05 | 0.16 | 0.18 |
| 98 | 1016 | 900 | 2400 | 4000 | 0.5 | 0.22 | 0.03 | 0.11 | 0.15 |
| 99 | 1016 | 950 | 2400 | 4000 | 0.5 | 0.20 | 0.02 | 0.06 | 0.13 |
| 100 | 1016 | 1000 | 2400 | 4000 | 0.5 | 0.19 | 0.00 | 0.02 | 0.11 |

## REFERENCES

[1] Adil, G.K., Rajamani, D., and Strong, D. (1996), Cell formation considering alternate routings, International Journal of Production Research, 34(5), 1361-1380
[2] Agarwal, A. and Sarkis, J. (1998), A review and analysis of comparative performance studies on functional and cellular manufacturing layouts, Computers and Industrial Engineering, 34(1),77-89
[3] Albadawi, Z., Bashir, H.A., and Chen, M. (2005), A mathematical approach for the formation of manufacturing cells, Computers and Industrial Engineering, 48(1), 3-21
[4] Anderberg, M.R. (1973), Cluster analysis for applications, Academic Press, New York
[5] Askin, R.G. and Subramanian, S.P. (1987), A cost based heuristic for grouping technology configuration, International Journal of Production Research, 25(1), 101-113
[6] Baykasoglu, A. (2004), A meta-heuristic algorithm to solve quadratic assignment formulations of cell formation problems without presetting number of cells, Journal of Intelligent Manufacturing, 15(6), 753-759
[7] Black, J.T., (1991), The Design of the Factory with a Future, New York: McGraw-Hill,
[8] Black, J.T., (2002), Design for System Success, Journal of Manufacturing Systems, 20(6),712-721
[9] Boctor, F.F. (1991), A linear formulation of the machine-part cell formation problem, International Journal of Production Research, 29(2), 343-356
[10] Burbidge, J.L. (1971), Production flow analysis, The Production Engineer, 42(12), 742752
[11] Cao, D. and Chen, M. (2004), Using penalty function and tabu search to solve cell formation problems with fixed cell cost, Computers and Operations Research, 31(1), 2137
[12] Carrie, A.S. (1973), Numerical taxonomy applied to group technology and plant layout, International Journal of Production Research, 11(4), 399-416
[13] Chandrasekharan, M.P. and Rajagopalan, R. (1986a), An ideal seed non-hierarchical clustering algorithm for cellular manufacturing, International Journal of Production Research, 24(2), 451-464
[14] Chandrasekharan, M.P. and Rajagopalan, R. (1986b), MODROC: an extension of rank order clustering for group technology, International Journal of Production Research, 24(5), 1221-1233
[15] Chandrasekharan, M.P. and Rajagopalan, R. (1987), ZODIAC-an algorithm for concurrent formation of part-families and machine-cells, International Journal of Production Research, 25(6), 835-850
[16] Chandrasekharan, M.P. and Rajagopalan, R. (1989), Groupability: an analysis of the properties of binary data matrices for group technology, International Journal of Production Research, 27(6), 1035-1052
[17] Chen, M. and Cao, D. (2004), Coordinating production planning in cellular manufacturing environment using Tabu search, Computers and Industrial Engineering, 46(3), 571-588
[18] Cheng, C.H., Kumar, A., and Motwani, J (1994), A comparative examination of selected cellular manufacturing clustering algorithms, International Journal of Operations and Production Management, 15(12), 86-97
[19] Chu, C.H. (1989), Cluster Analysis in Manufacturing Cellular Formation, International Journal of Management Science, 17(3),289-295
[20] Chu, C.H. and Tsai, M. (1990), A comparison of three array-based clustering techniques for manufacturing cell formation, International Journal of Production Research, 28(8),1417-33
[21] Defersha F.M. and Chen, M. (2006), A comprehensive mathematical model for the design of cellular manufacturing systems, International Journal of Production Economics, 103(2), 767-783
[22] Dimpoulos, C. and Mort, N. (2000), Evolving Similarity Coefficients for the Solution of Cellular Manufacturing Problems, Proceedings of the IEEE Conference on Evolutionary Computation, 1, 617-624
[23] Dos Santos, N. and De Araujo L. (2003), Computational system for group technology PFA case study, Integrated Manufacturing Systems, 14(2), 138-152
[24] Gunasingh, K.R. and Lashkari, R.S. (1989), Machine grouping problem in cellular manufacturing systems- an integer programming approach, International Journal of Production Research, 27(9), 1465-1473
[25] Gunasingh, K.R. and Lashkari, R.S. (1991), Simultaneous grouping of parts and machines in cellular manufacturing systems. An integer programming approach, Computer and Industrial Engineering, 20(1), 111-117
[26] Gupta, T. and Seifoddini, H (1990), Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system, International Journal of Production Research, 28(7),1247-69
[27] Heragu, S.S. (1994), Group Technology and cellular manufacturing, IEEE Transactions on Systems, Man, and cypernetics, 24(2), 203-215
[28] Hsu, C.P. (1990), Similarity coefficient approaches to machine component cell formation in cellular manufacturing: a comparative study, Ph.D. Dissertation, Department of Industrial and Systems Engineering, University of Wisconsin,
[29] Hwang, H. and Hui O. (2003), Another similarity coefficient for the p-median model in group technology, International Journal of Manufacturing Technology and Management, 5(1-2) 38-45
[30] Irani, S.A. (1999), Handbook of cellular manufacturing systems, New York: Wiley \& Sons,
[31] James, T.L., Brown, E.C., and Keeling, K.B. (2007), A hybrid grouping genetic algorithm for the cell formation problem, Computers and Operations Research, 34(7), 2059-2079
[32] Josien, K. and Liao, T.W.R. (2002), Simultaneous grouping of parts and machines with an integrated fuzzy clustering method, Fuzzy Sets and Systems, 126(1), 1-21
[33] Kamal, S. and Burke, L.I. (1996), FACT: a new neural network-based clustering algorithm for group technology, International Journal of Production Research, 34(4),919-946
[34] Kaparthi, S. and Suresh, N.C. (1993), An improved neural network leader algorithm for part-machine grouping in group technology, European Journal of Operational Research, 69(3),342-356
[35] Kattan T., Baki, M.F., and Aneja, Y. (2007), Cell formation considering Efficacy- A Lagrangean relaxation to mathematical programming approach, Manuscript, 1-12
[36] Keeling, K.B., Brown, E.C., and James, T.L. (2007), Grouping efficiency measures and their impact on factory measures for the machine-part cell formation problem: A simulation study, Engineering Applications of Artificial Intelligence, 20(1), 63-78
[37] King, J.R. (1980), Machine component grouping in production flow analysis, International Journal of Production Research, 20(1), 213-237
[38] King, J.R. and Nakornchai, V. (1982), Machine-component group formation in group technology: review and extension, International Journal of Production Research, 20(1), 117-133
[39] Kizil, M., Ozbayrak, M. (2004), A tradeoff analysis between process plan selection and cell formation in cellular manufacturing, International Journal of Advanced Manufacturing Technology, 23(7-8), 501-506
[40] Kumar, A.G., Rajamani, D., and Strong, D. (1997), Assignment allocation and simulated annealing algorithms for cell formation, IIE Transactions, 29(1), 53-67
[41] Kumar, C.S. and Chandrasekharan, M.P. (1990), Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International Journal of Production Research, 28(2), 233-243
[42] Kumar, K.R. and Vannelli, A. (1987), Strategic subcontracting for efficient disaggregated manufacturing, International Journal of Production Research, 25(4), 17151728
[43] Kumar, K.R., Kusiak, A., and Vannelli, A. (1986), Grouping of parts and components in flexible manufacturing systems, European Journal of Operational Research, 24, 387-397
[44] Kusiak, A (1987), A generalized group technology concept, International Journal of Production Research, 25(4), 561-569
[45] Lashkari, R.S. and Gunasingh, K.R. (1990), A langrangean relaxation approach to machine allocation in cellular manufacturing systems, Computers and Industrial Engineering, 19(1-4), 442-446
[46] Lei, D. and Wu, Z. (2005), Tabu search-based approach to multi-objective machine-part cell formation, International Journal of Production Research, 43(24), 5241-5252
[47] Logendran, R. (1993), Binary integer programming approach for simultaneous machinepart grouping in cellular manufacturing systems, Computer and Industrial Engineering, 24(3), 329-336
[48] Macqueen, J.B. (1967), Some methods for classification and analysis of multivariate observations, Proceedings of the fifth symposium on mathematical statistics and probability, University of California at Berkeley
[49] Mahdavi, I., Javadi, B., Fallah-Alipour, K., and Slomp, J. (2007), Designing a new mathematical model for cellular manufacturing system based on cell utilization, Applied Mathematics and Computation, 190(1), 662-670
[50] Mahesh, O. and Srinivasan, G. (2002), Incremental cell formation considering alternative machines, International Journal of Production Research, 40(14),3291-3310
[51] McAuley, D. (1972), Machine grouping for efficient production, The Production Engineer, 51, 53-57
[52] McCormic, J.W.T., Schweitzer, P.J., and White, T.W. (1972), Problem decomposition and data reorganization by a clustering technique, Operations Research, 22, 993-1008
[53] Miltenburg, J. and Zhang, W. (1991), A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology, Journal of Operations Management, 10(1), 44-72
[54] Mosier, C.T. (1989), An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell formation problem, International Journal of Production Research, 27(10), 1811-1835
[55] Mukattash, A., Abbasi, G., Tahboub, K., and Adil, M. (2007), A modified revised Pmedian approach to cell formation, International Journal of Industrial and Systems Engineering, 2(1), 51-60
[56] Nair, G.J.K. and Narendran, T.T. (1996), Grouping index: a new quantitative criterion for goodness of block-diagonal forms in group technology, International Journal of Production Research, 34(10), 2767-2782
[57] Ozturk, G., Ozturk, Z.K., and Islier, A.A. (2006), A comparison of competitive neural network with other AI techniques in manufacturing cell formation, Advances in Natural Computation, ICNC 2006, Proceedings, 1(4221), 575-583
[58] Pai, P., Chang P., and Lee, S. (2005), Part-machine family formation using genetic algorithms in a fuzzy environment, International Journal of Advanced Manufacturing Technology, 25(11-12), 1175-1179
[59] Pierreval, H., Caux, C., Paris, J.L., and Viguier, F. (2003), Evolutionary approaches to the design and organization of manufacturing systems, Computers and Industrial Engineering, 44(3),339-364
[60] Rajagopalan, R. and Batra, J.L. (1975), Design of cellular production systems. A graphtheoretic approach, International Journal of Production Research, 13(6), 567-579
[61] Rajagopalan, R. and Fonseca, D. (2006), A genetic algorithm approach for machine cell formation, Journal of Advanced Manufacturing Systems, 5(1), 27-44
[62] Saidi-Mehrabad and Safaei (2007), A new model of dynamic cell formation by a neural approach, International Journal of Advanced Manufacturing Technology, 33(9-10) 10011009
[63] Sarker, B.R. (2001), Measure of grouping efficiency in cellular manufacturing systems, European Journal of Operational Research, 130, 588-611
[64] Sarker, B.R. and Khan, M. (2001), A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure, IIE Transactions, 33, 11-27
[65] Sarker, B.R. and Mondal, S. (1999), Grouping efficiency measures in cellular manufacturing: a survey and critical review, International Journal of Production Research, 37(2),285-314
[66] Seifoddini, H. (1989), A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications, International Journal of Production Research, 27(7), 1161-1165
[67] Shambu, G. and Suresh, N.C. (2000), Performance of hybrid cellular manufacturing systems: A computer simulation investigation, European Journal of Operational Research, 120,436-458
[68] Shargal, M., Shekhar, S., and Irani, S.A. (1995), Evaluation of search algorithms and clustering efficiency measures for machine-part matrix clustering, IIE Transactions, 27, 43-59
[69] Shunk, D.L. (1985), Group technology provides organized approach to realizing benefits of CIMS, Industrial Engineering, 17(4),74-80
[70] Slomp, J., Chowdary, B.V., and Suresh, N.C. (2005), Design of virtual manufacturing cells: a mathematical programming approach, Robotics and Computer-Integrated Manufacturing, 21(3), 273-288
[71] Snead, C. (1989), Group Technology: Foundation for competitive Manufacturing, Van Nostrand Reinhold,
[72] Solimanpur, M., Vrat, P., and Shankar, R. (2004), A multi-objective genetic algorithm approach to the design of cellular manufacturing systems, International Journal of Production Research, 42(7), 1419-1441
[73] Srinivasan, G., and Narendran, T.T. (1991), GRAFICS-a nonhierarchical clustering algorithm for group technology, International Journal of Production Research, 29(3),463-478
[74] Srinivasan, G., Narendran, T.T., and Mahadevan, B. (1990), An assignment model for the part-families problem in group technology, International Journal of Production Research, 28(1),145-152
[75] Srinivasan,G. (1994), A clustering algorithm for machine cell formation in group technology using minimum spanning trees, International Journal of Production Research, 32(9), 2149-2158
[76] Stawowy, A. (2006), Evolutionary strategy for manufacturing cell design, The International Journal of Management Science, 34(1), 1-18
[77] Steudel, H.J. and A. Ballakur (1987), A dynamic programming based heuristic for machine grouping in manufacturing cell formation, Computers and Industrial Engineering, 12(3), 215-222
[78] Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., and Safaei, N. (2007), Design of a facility layout problem in cellular manufacturing systems with stochastic demands, Applied Mathematics and Computation, 184(2), 721-728.
[79] Torkul, O., Cedimoglu, I.H., and Geyik, A.K. (2006), An application of fuzzy clustering to manufacturing cell design, Journal of Intelligent and Fuzzy Systems, 17(2), 173-181
[80] Venugopal, V. and Narenderan, T. (1992), Genetic Algorithm Approach to the MachineComponent Grouping Problem with Multiple Objectives, Computers and Industrial Engineering, 22(4), 469-480
[81] Venugopal, V. and Narendran, T.T. (1994), Machine-cell formation through neural network models, International Journal of Production Research, 32(9), 2105-2116
[82] Wang, J. and Roze, C. (1994), Formation of machine cells and part families in cellular manufacturing: a linear integer programming approach, Proceedings of the IEEE International Conference on Industrial Technology, 350-4
[83] Wang, J. and Roze, C. (1997), Formation of machine cells and part families: A modified p-median model and a comparative study, International Journal of Production Research, 35(5), 1259-1286
[84] Wei, J.C and Kern, G.M. (1989), Commonality analysis: a linear cell clustering algorithm for group technology, International Journal of Production Research, 27(12), 2053-2062
[85] Wemmerlov, U. and Hayer, N.L. (1989), Cellular manufacturing practices, Manufacturing Engineering, 102(3), 79-82
[86] Won, Y. and Currie, K. (2006), An Effective P-median Model Considering Production Factors in Machine Cell/Part Family Formation, Journal of Manufacturing Systems, 25(1), 58-64
[87] Won, Y. and Lee, K.C. (2004), Modified p-median approach for efficient GT cell formation, Computer and Industrial Engineering, 46(1), 495-510
[88] Xambre, A.R. and Vilarinho, P.M. (2003), A simulated annealing approach for manufacturing cell formation with multiple identical machines, European Journal of Operational Research, 151(2), 434-446
[89] Xu, H. and Wang, H.P. (1989), Part family formation for GT applications based on fuzzy mathematics, International Journal of Production Research, 27(9), 1637-1651

## VITA AUCTORIS

## Name: Talal Kattan

## Citizenship: Canadian

YOB: 1975

## Eduation:

- 2007, University of Windsor - M.A.Sc.

Major: Industrial \& Manufacturing Systems Engineering.

- 1999, Colorado State University at Pueblo, B.Sc.

Major: Industrial Engineering
Minor: Computer Information System

## Honors and Awards:

Graduate student tuition waiver award for academic standing 2006-2007
Graduate student Research Assistantship award, Sep 2005- Sep 2007
Conference travel award, Graduate Studies, University of Windsor, May 2007
Educational Experience:
Spring 06, Graduate Assistant, Class of "Health and Safety", University of Windsor.
Summer 06, Graduate Assistant, "Human Factors", University of Windsor.
Fall 06, Graduate Assistant, "Human Factors", University of Windsor.
Spring 07, Teaching Assistant, "Production Management", University of Windsor.
Summer 07, Graduate Assistant, "Human Factors", University of Windsor.

