COST BASED MACHINE-COMPONENT GROUPING MODEL: IN GROUP TECHNOLOGY

By

HAMID SEIFODDINI

Bachelor of Science
Arya-Mehr University of Technology
Tehran, Iran
1974

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1977

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 1984

COST BASED MACHINE-COMPONENT GROUPING
MODEL: IN GROUP TECHNOLOGY

Thesis Approved:

Philip M. Wolfe
Thesis Adviser

M. Palmer Terrell

Kenneth E. Case

Wayne Turner

Donald W. Grace

Norman N. Durham
Dean of the Graduate College

PREFACE

This research is an extension to the application of the similarity coefficient method to the machine-component grouping process. There are three major problems associated with the similarity coefficient method when applied to the machine-component grouping process. First, the algorithms based on this method do not deal with bottleneck machines in the machine-component grouping process. Secondly, the selection of a proper solution among a set of solutions given by these algorithms is not based on manufacturing related factors. Finally, the algorithms based on the similarity coefficient method do not consider the specifications of machines and the nature of the manufacturing process involved in the machine-component grouping process. These problems are overcome through the development of a new machine-component grouping model.

I am deeply indebted to many individuals for their support and encouragement. My special thanks and appreciation go to my major adviser, Dr. Philip M. Wolfe, for his inspiration, support, and encouragement during this course of study. I would also like to thank the other members of my doctoral committee, Dr. Wayne C. Turner, Dr. Kenneth E. Case, Dr. M. Palmer Terrell, and Dr. Donald W. Grace for their continuous guidance and support.

My thanks and appreciation also go to Mrs. Melanie Bayles for her excellent typing.

My love, gratitude, and appreciation go to my wife Minoo who displayed patience and understanding during the course of this study.

Finally, I would like to extend my thanks and appreciation to my parents, Mr. Abdolhossein and Mrs. Dorbebe Seifoddini, who have done all in their power to provide me with a good life and education.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION. 1

Statement of the Problem 4
Definition of Group Technology. 4
Historical Background 5
Benefits from Group Technology. 8
Benefits from Group Technology in Design 9
Benefits from Group Technology in Manufacturing. 10
Benefits from Group Technology in Inventory Control. 12
Group Technology and Management 13
Implementation of Group Technology. . . 14
Machine-Component Grouping. 15
Research Objectives. 17
Summary of Results 18

II. LITERATURE REVIEW 22

Machine-Component Grouping 22
Production Flow Analysis 23
Machine-Component Group Analysis 27
Iri's Algorithm 28
Bond Energy Algorithm 28
Rank Order Clustering Algorithm 29
Similarity Coefficient Method. 31
Single Linkage Cluster Analysis (SLCA). 32
Graph Theoretic Method. 38

III. COST BASED MACHINE-COMPONENT GROUPING MODEL . . . 40

Part-Families and Machine Cells. 41
The Clustering Algorithm. 42
Threshold Value 46
Bottleneck Machines 48
The Model. 49
Overview. 49
Assumptions of the Model. 54
Inputs to the Model 55
Output of the Model 57
Algorithmic Flow Chart of the Model . . 57
The Clustering Unit 62

Chapter Page

Assignment Unit 65
Bottleneck Unit 68
Threshold Value Unit. 79

IV. SOLUTION METHODOLOGY. 86

Data Storage and Analysis. 87
Bit-Level Data Storage. 87
Data Storage and Retrieval. 91
Data Analysis 94
Cluster Analysis Techniques. 96
Procedures for Machine-Component Grouping. . 105
Procedure for Forming Machine Cells . . 105
Procedures for Dealing With Bottleneck Machines. 112
Procedures for Selection of the Threshold Value 118

V. ANALYSIS OF RESULTS 123

Selection of the Threshold Value 126
Identification and Duplication of Bottleneck Machines. 134
Duplication Without Considering Cost Factors 138
Duplication Based on Cost Factors . . . 145
Sensitivity Analysis 155
Sensitivity to Similarity Measure . . . 155
Sensitivity to Production Volume. . . . 159
Sensitivity to Cost Coefficients. . . . 163

VI. SUMMARY AND CONCLUSIONS 168

Recommendations for Further Studies. 170

BIBLIOGRAPHY 172

APPENDIX A - OUTLINE OF THE CRAFT ALGORITHM. 177

APPENDIX B - DESCRIPTION OF VARIABLES USED IN THE COMPUTER PROGRAM FOR THE COST BASED MACHINE-COMPONENT GROUPING MODEL. 180

APPENDIX C - FORTRAN CODES FOR MACHINE-COMPONENT GROUPING MODEL. 186

LIST OF TABLES

Table Page

I. Correspondence Between Binary and Hexidecimal Numbers. 89

II. Masking Operation on IWORD 91

III. Cells and Associated Machines for Threshold Value = 0.72 127

IV. Cells and Associated Machines for Threshold Value = 0.56 127

V. Cells and Associated Machines for Threshold Value = 0.41 128

VI. Cells and Associated Machines for Threshold Value = 0.33 128

VII. Cells and Associated Machines for Threshold Value = 0.25 128

VIII. Cells and Associated Machines for Threshold Value = 0.17 128

IX. Inter-Cellular and Intra-Cellular Material Handling Costs 134

X. Cells and Associated Machines for Threshold Value = 0.17 135

XI. Bottleneck Machines and Related Inter-Cellular Moves. 137

XII. Exceptional Parts on Machines 6 and 8. 146

XIII. Cost Data for Machines 6 and 8 147

XIV. Exceptional Parts and Their Parent Cells 149

XV. Bottleneck Machine Required by Different Cells . 150

XVI. Duplication Based on Cost Factors. 151

XVII. Duplication Results for Burbidge's Solution. . . 153

Table Page

XVIII. Duplication Results for Burbidge's Soution and New Solution. 154

XX. Duplication Process for 20% Decrease in Production Volume. 161

XX. Duplication Process for 10% Decrease in Production Volume. 162

XXI. Duplication Process for 10% Increase in Production Volume. 162

XXII. Duplication Process for 20% Increase in Production Volume. 163

XXIII. Duplication Process for ACOST = 0.5. 166

LIST OF FIGURES

Figure Page

1. Two Types of Layout 6
2. Machine-Component Chart 32
3. Machine-Component Chart 34
4. The Similarity Matrix for Figure 3. 35
5. Dendogram for the Example Problem 36
6. A Machine-Component Chart 44
7. The Similarity Matrix for Data in Figure 6. 44
8. Machine-Component Chart (3, 4, 5, and 6 are Exceptional Parts). 45
9. Machine-Component Chart (3, 4, and 5 are Exceptional Parts). 45
10. Machine-Component Grouping Model. 52
11. Algorithmic Flow Chart of Clustering Unit. 58
12. Algorithmic Flow Chart of Assignment Unit. 59
13. Algorithmic Flow Chart of Bottleneck Unit. 60
14. Algorithmic Flow Chart of Threshold Value Unit. . . 61
15. Machine-Component Chart 68
16. A Machine-component Chart With Five Machines. . . . 81
17. 2 x 2 Table for Two Variables 97
18. General Similarity Matrix 100
19. Similarity Matrix With Symmetric Similarity Coefficients. 100
20. Machine-Component Chart 103

Figure Page

21. Similarity Matrix for Data in Figure 20 104

22. Revised Similarity Matrix After First Iteration . . 104

23. Revised Similarity Matrix After Second Iteration. . 104

24. Revised Similarity Matrix After Third Iteration . . 105

25. Machine Cells at a Threshold Value of 1.00. 110

26. Machine Cells at a Threshold Value of 0.50. 112

27. Machine-Component Chart 114

28. Machine-Component Chart 120

29. The Original Machine-Component Chart. 124

30. Dendogram for Test Problem. 125

31. From-to Chart for Threshold Value = 0.41. 130

32. From-to Chart for Threshold Value = 0.33. 131

33. From-to Chart for Threshold Value = 0.25. 131

34. From-to Chart for Threshold Value = 0.17. 132

35. Machine-Component Chart Before Considering Bottleneck Machines, Parts are Assigned to Cells, "One" Entries Outside the Blocks Represent Inter-Cellular Moves. 136

36. Final Machine-Component Chart (LIMIT = 0) 139

37. Final Machine-Component Chart (LIMIT = 2) 140

38. Final Machine-Component Chart (LIMIT = 10). 141

39. Burbidge's Manual Solution. 143

40. Final Machine-Component Chart Considering Economic Factors 148

41. Dendogram for Dice Similarity Coefficient 156

42. Dendogram for the Third Similarity Coefficient. . . 160

43. Final Machine-Component Chart for 20% Decrease in Production Volume 164

44. Final Machine-Component Chart for 20% Increase in

Figure Page

Production Volume 165

45. Final Machine-Component Chart (ACOST = 0.5) 167

46. A From-To Chart for a Facility With Five Departments 179

CHAPTER I

INTRODUCTION

The increasing demand for diversity in manufacturing commodities has compelled manufacturers to produce a large number of products in smaller batches. With traditional manufacturing methods, a reduction in batch size would result in higher production costs due to increased setup times. This is only one of the problems associated with traditional methods of manufacturing (11, 29, 36). Though batch-type manufacturing systems have been in operation since the beginning of the Industrial Revolution, it is becoming clear that there are major shortcomings associated with them (30, 50). Some of the more serious shortcomings are the following:

- In batch-type manufacturing systems, the three major functions : planning, coordination, and control are complex.
- Compared to the overall throughput times, setup times are long.
- Due to part travel between different departments, the transportation cost is high; and there is a need to carry large in-process inventories.

In most manufacturing orgainizations, especially in batch-type manufacturing, only a small portion of production costs can be attributed to the manufacturing process. In fact, product design, method development, process planning/ control, and tool design/manufacturing account for a major portion of the costs. Even when a part is in process, only a small portion of the production time is spent on machines; the rest is spent on such things as moving the part among work stations, waiting for a lot to be formed, or a machine to be set up. Therefore, the success of a manufacturing organization, to a great extent, depends upon the way these problems are handled. As such, technological development today calls for the introduction of new scientific principles in solving these problems (22, 51).

Group Technology is a proven technique that is extremely effective in solving many of the problems associated with batch-type manufacturing. Group Technology is a manufacturing concept or philosophy that seeks the similarity of manufacturing processes and uses it as a way to reduce production costs through reduced design costs, smaller setup times, improved process planning, reduced tooling requirements, less throughput times, and better utilization of expensive machinery.

The systematic application of Group Technology as a scientific technique is new, but the idea itself has been used for many years in one way or another and under different names. Group Technology as a technique to improve productivity and to reduce production costs was first applied in Europe (51). Japanese companies have been using Group Technology as a way of improving productivity for many years. In the United States, the concept and techniques of Group Technology have received a great deal of attention in recent years. The need for productivity improvement in the U. S. industry has led to the concentration of time and effort in the area of Computer Aided Manufacturing (CAM). This, in turn, has enhanced the interest in Group Technology as an essential part of a successfully implemented CAM program (21).

In the application of Group Technology to a production process, the basic idea is to find all components having similar manufacturing requirements and to group them in a single family. Then, machine cells are formed, such that all components belonging to one family can be processed within a single machine cell. Therefore, one of the major problems in Group Technology is machine-component grouping. Many models have been developed to carry out this job (13, 23, 31, 35).

The purpose of this dissertation is to review the existing machine-component formation algorithms and to introduce a new model which improves the machine-component formation process by considering the material handling costs and production data.

Statement of the Problem

Definition of Group Technology

A broad definition of Group Technology was given by Professor V. B. Solaja at the First International Seminar on Group Technology in Turin, Italy. (19) His introductory statement was the following:

> Group Technology is the realization that many problems are similar and that, by grouping similar problems, a single solution can be found to a set of problems, thus saving time and effort (19, p. 51).

In engineering practice, it is not unusual to repeat the whole design process for the same part or for quite similar parts, due to a lack of a classification system which can easily bring to the designer's attention the fact that such a part has already been designed. It is also quite possible for a methods engineer to prepare a route card for a part without knowing that such a route card has been prepared before. Group Technology attempts to eliminate such practices.

The definition of Group Technology for engineering purposes can be given as:

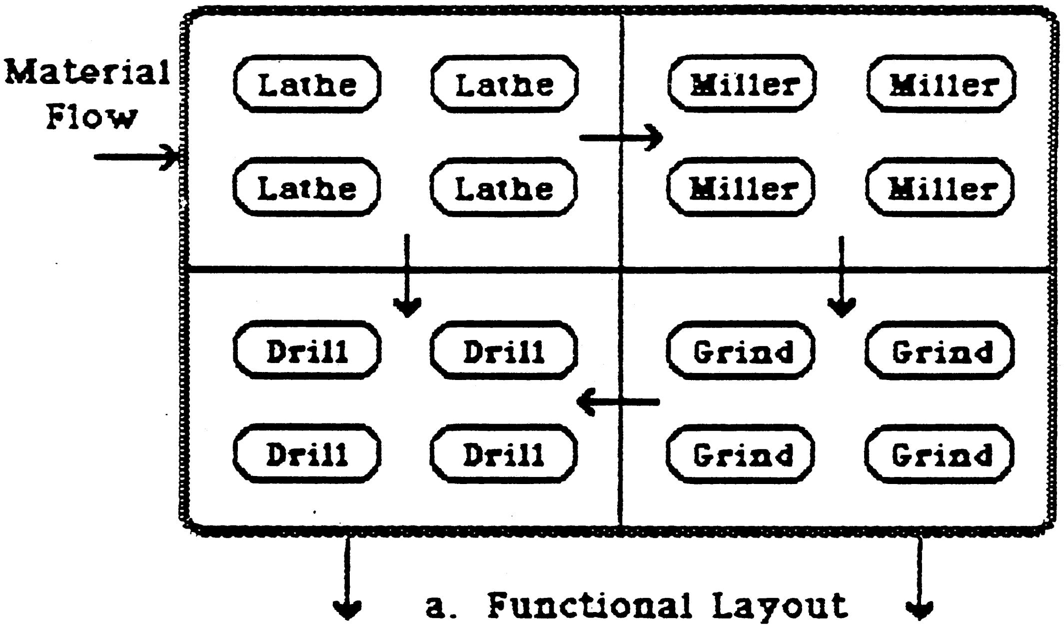
> Group Technology is the replacing of traditional jobbing shop manufacturing by the analysis and grouping of work into families, and the formation of groups of machines to manufacture these families on a flow-line principle with the objective of minimizing setting times and throughput times (19, p. 51).

In job-shop manufacturing, the arrangement of machines is based on functional layout. That is, all machines capable of doing similar operations are laid out in one place. To process a part, it is loaded on one machine for its first operation, then unloaded and moved to another machine, sometimes quite far from the first one, for its second operation. Also, in a batch-type manufacturing, due to improper sequencing of jobs, it is quite possible to set up a machine several times to process a number of similar jobs. The best way to eliminate these time delays is by the introduction of Group Technology.

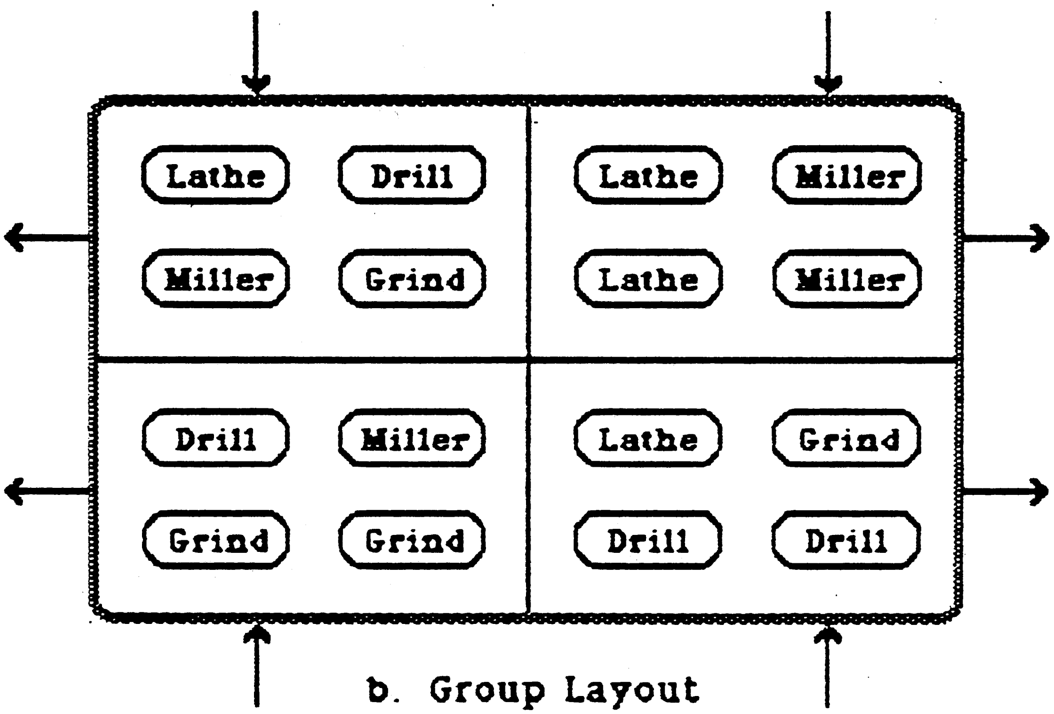
In Group Technology, similar parts are grouped together to form families of similar components (part-families). Then, within each family the components with similar manufacturing processes are processed in a proper sequence in order to reduce the setup times. The functional layout is, also, replaced by a group layout in which the groups of machines are formed in such a way that all components related to a part-family can be completely processed within a single machine cell. This saves the transportation costs, paper work, and time delays associated with moving the parts from one work station to another in a functional layout. Group Technology can be applied to many areas of production and create enormous benefits. Some of these benefits will be discussed later. Figure 1 illustrates the difference between a functional and a group layout.

Historical Background

Group Technology, as an engineering practice, has been



a. Functional Layout



b. Group Layout

Figure 1. Two Types of Layout

used for many years. There are cases in which group layouts have been employed as a way of reducing transportation costs. Group Technology has also been used in many factories to reduce tooling costs (51). However, most of these have not been done in a systematic manner. They have been introduced as solutions to isolated problems which engineers encounter at different times and places.

The first published work on the subject of Group Technology is credited to S. P. Mitrofanove (46) from Russia. In 1958, his book entitled The Scientific Principle of Group Technology discussed the subject in detail and created a great deal of interest in the field.

According to Phillips et al. (51) the first systematic implementation of Group Technology was reported by a French company named 'Forges et Ateliers de Constructions Electrignes de jemment.'

In the United Kingdom, many universities and research institutions have been working on Group Technology since the 1960's. Universities such as Birmingham, Aston, Stanford, and Manchester, have conducted many research projects on the subject. Production Engineering Research Association, PERA, has been actively involved in encouraging the application of Group Technology in the U. K. industries for a long period of time (34).

Professor Burbidge has done a considerable amount of work on the subject. His Production Flow Analysis approach (PFA) is very well known all over the world (7, 9, 10, 11).

In West Germany, one of the most remarkable works on the subject of Group Technology has been done by Professor H. Opitz at Aachen University (49). His intensive work on workpiece classification led to the development of the Opitz Classification System, which is one of the most popular classification methods in European industry.

Japan is also one of the pioneers in development and implementation of Group Technology. Japanese industries have used Group Technology as early as the 1960's. Many government sponsored institutes and industries have been working in this area and have developed several classification and coding systems for different purposes. Recently, many Japanese companies, aware of Group Technology applications in Computer Aided Manufacturing (CAM) in the United States, are concentrating on this specific area of Group Technology to cope with future demands (34).

In the United States, in the past, relatively little effort has been devoted to the development and application of Group Technology. However, with new emphasis on the need for productivity improvement in U. S. industry, and with continuous effort in the development and implementation of CAM systems, the interest in Group Technology is growing rapidly.

Benefits from Group Technology

Over the last three decades, many manufacturing organizations have been using Group Technology; and numerous benefits have been reported (30, 32, 43, 63). In the United States the Langston Division of the Harris-Intertype Corporation was one of the first factories to implement a successful Group Technology program (30). The major benefits derived from the program include 50% increase in parts produced per man-hour, a 22% decrease in floor space requirements, and a reduced throughput time from 30-45 days to 2-5 days. Generally, the following reductions have been attributed to the implementation of Group Technology programs:

- 50% in new parts design
- 10% in number of drawings
- 60% in industrial engineering time
- 20% in plant floor space requirements
- 40% in raw material stocks
- 60% in in-process inventories
- 70% in setup times
- 70% in throughput time.

In addition, Group Technology improves the work environment by humanization of work. Group Technology produces benefits in areas such as product design, manufacturing process, production planning/control, tooling, inventory control, and management (1, 3, 13, 24, 34, 35). A brief description of each follows.

Benefits from Group Technology in Design. Component variety is one of the serious problems associated with batch-type manufacturing, and is one source of extra production costs. In job-shop manufacturing, there are many similar components used in different products. The development and implementation of a classification system based on the similarity of components can easily discover such similar components; and the design engineer can eliminate unnecessary designs. Such a classification system is an integral part of Group Technology.

Design duplication is a common practice in many manufacturing organizations, especially in batch-type manufacturing, and is another source of extra production costs. Recognition of such duplications needs an effective design data retrieval system, based on the classification and coding systems mentioned earlier.

In addition to the above advantages, the development and implementation of classification and coding systems based on the concepts of Group Technology leads to the standardization of design features, the simplification of design process, and the improvement of costs estimation system.

Benefits from Group Technology in Manufacturing. The greatest cost savings can be achieved by the application of Group Technology in the areas of production planning/control, manufacturing processes, and tool design/manufacturing.

In batch-type manufacturing, the components of a specific lot or an order are moved from one work station to another for different operations. These work stations are sometimes far apart and most components must wait for a lot to be formed before moving to another work center. One way to eliminate these extra transportation costs and time delays is by the application of Group Technology. That is, when the functional layout is replaced by a group layout, all the components of a family can be processed within a single machine cell. Therefore, the need for extra transportation among different work stations is eliminated. Also, by proper sequencing of the components within each cell, it is possible to reduce the setup times, which account for a major portion of the production costs, especially in the case of expensive machinery.

Another source of cost reduction in manufacturing is in the area of group tooling. After the establishment of machine-component groups, it is quite possible to design group tools, such as group fixtures, which, combined with a special adapter, are capable of processing all components of a family or a large number of them. If this is done, the tooling costs would decrease and the setup times required by the individual fixtures would be eliminated.

In production planning and control, traditionally, each component is considered as one unit in the scheduling phase. Consequently a planner has to deal with a large number of units. Using group scheduling, this large number of components reduces to a much smaller number of families of similar components. As a result, the scheduling problem is simplified, and the sequencing of jobs within each group becomes less complicated. This is true because it is much easier to deal with scheduling and job sequencing problems within a machine cell in a group layout rather than within a plant in a functional layout.

Finally, the application of Group Technology to the manufacturing process reduces the throughput times. With smaller throughput times for products, due dates are more likely to be met. The delivery of orders on times is an important factor in the success of a manufacturing organization in a competitive market.

Benefits from Group Technology in Inventory Control. With long throughput times, it is necessary to keep a large amount of inventory to cope with the market changes and to meet the consumers' demand. A long throughput time, combined with time delays associated with batch-type manufacturing, also, calls for a huge amount of in-process inventory, which can be an important source of increased production costs in most manufacturing organizations. As was mentioned before, application of Group Technology to the manufacturing process reduces the throughput times, and, in many cases, eliminates the time delays. As a result, the required in-stock and in-process inventories decrease substantially.

Another prevalent problem associated with inventory systems is that of obsolescence. With traditional lot-size manufacturing, it is necessary to produce in large lots to reduce the setup times and the corresponding production costs. Also, with a long throughput time, a product should be placed in production well ahead of the time it is needed in stock. These two factors in the presence of fast technological change, in some cases, cause some portion of a lot to be obsolete even before it gets to the market. This problem can likewise be overcome by reduction of throughput times through the application of Group Technology (19).

Group Technology and Management

The implementation of a group layout as part of a Group Technology application changes the technical structure of the manufacturing organization. This brings about some changes in the overall structure of the organization, which, in turn, calls for changes in managerial aspects of the organization.

With group layout, a number of workers are assigned to a machine cell, in which a limited number of similar jobs are completely processed. In such an environment, the supervisor has a better control over the group and is able to do his job more efficiently. From the worker's point of view, functioning in a machine cell provides each worker with the opportunity to become familiar with the other workers' jobs and to realize how his work is related to that of others. This realization leads to higher performance and better work quality.

Again, workers in a machine cell are able to identify with their work. They can see their role in completing a job. This implies that, in contrast to the traditional layout, workers do not feel that they are doing a meaningless and trivial job. Therefore, job satisfaction in the case of group layout, is expected to be much higher than in the case of functional layout. Group morale is also improved by implementation of group layout, which, in turn, results in higher productivity.

Implementation of Group Technology

Group Technology may be used as a solution to an isolated problem encountered by a firm, or it may be implemented as an overall approach to productivity improvement of an organization. In each case, the implementation of Group Technology should be based upon a complete analysis of the costs and benefits involved. There are two approaches to Group Technology implementation: implementation throughout the factory at once or gradual implementation. Since the throughout implementation involves a high risk and requires a large number of specialized personnel besides a huge amount of preparatory work, the second approach is more popular in practice. In most cases, at first, a pilot cell is established, then, based on the experience gained, additional cells are gradually introduced until all the components are produced in a cellular manufacturing system.

Machine-Component Grouping

The very first step in the application of Group Technology to the manufacturing process is to identify the families of similar components (part-families) and to form the associated machine cells to manufacture these part-families. This process is referred to as "machine-component grouping". Machine-component grouping is the core of Group Technology application to the manufacturing process. For this reason, many research groups and interested individuals have been involved in research and development in this area. As a result, many methods have been developed to find part-families and form the associated machine-component groups. In the next chapter some of these methods will be presented and discussed.

However, before getting into further details of machine-component grouping, some related terms which are very common in the literature need to be defined:

Part-family: A set of components which have some number of operations in common and are grouped together to be processed in a single machine cell.

Machine Cell: A set of machines capable of processing most, if not all, of the operations required to manufacture one or more part-families.

Machine-Component Chart: An M x N matrix the elements of which are either zero or one. M and N represent the number of machines and parts respectively. If the entity in row i and column j of the matrix is one, it indicates that part j has an operation on machine i; if the entry is zero it does not.

Similarity Coefficient: For two machines, this is the number of parts visiting both machines, divided by the number of parts visiting at least one of the two machines.

Similarity Matrix: An M x M matrix, containing all pairwise similarity coefficients between elements to be clustered.

Threshold Value: A similarity coefficient that indicates the similarity level at which two machines or two groups of machines should be joined together. Pairwise similarity coefficients between machines or groups of machines are calculated, and those machines or groups of machines which have a similarity coefficient greater than the threshold value are grouped together.

Inter-Cellular Moves: The number of part types transported between cells.

Intra-Cellular Moves: The number of part types transported between machines within cells.

Inter-Cellular Trips: Inter-cellular moves weighted by the number of parts to be produced and by the number of moves each part makes

Intra-Cellular Trips: Intra-cellular moves weighted by the number of parts to be produced and by the number of moves each part makes

Bottleneck Machine: A machine which is required by a large number of parts from different cells. A bottleneck machine creates a great deal of inter-cellular moves.

Exceptional Part: A part which has operations in more than one cell.

Duplication: Assignment of a bottleneck machine to several cells.

Block Diagonal Form: A form of machine-component charts in which "one" entries are concentrated in blocks along the diagonal of the chart.

Generally, all machine-component grouping algorithms fall into two major categories: (a) algorithms utilizing machine-component group analysis and (b) those using the similarity coefficient method. Most algorithms in the first category form the machine-component groups by exchange of rows and columns of the machine-component chart in an iterative process. A brief description of some of these algorithms will be given in Chapter II. The algorithms based on the similarity coefficient method use the pairwise similarity coefficients of machines to group machines into cells. A detailed discussion of the similarity coefficient method is given in Chapter II.

Research Objectives

The overall objective of this research was to develop a machine-component grouping model that considers costs and similarity coefficients. The model was designed to improve the machine-component grouping process by considering the material handling cost as a basis for selection of the threshold value, by dealing with the problem of bottleneck machines, and by considering the modification of the production cells. The primary objective of the research was to overcome the three major problems associated with the similarity coefficient method:

(1) The bottleneck machines problem,
(2) cost-benefit analysis of the duplication process, and
(3) determination of a proper threshold value.

Summary of Results

The model developed here provides a practical basis for application of the similarity coefficient method to solve the machine-component grouping problem. The model usus the similarity coefficient method and gives an analytical solution to the problem of clustering together similar machines. At the same time, it deals with the bottleneck machines, duplication process, and selection of a threshold value.

The following new features have been built into the model:

(1) Material handling costs among, and within, cells are determined and used as a basis for selection of a threshold value. The material handling costs are associated with inter-cellular and intra-cellular trips.

(2) Production data are incorporated into the

model. The following items are considered:
- production volume in terms ot number ot parts to be produced.
- processing times of parts on machines.
- production hours per day.

(3) Data related to machines are incorporated into the model. The following items are considered:
- price and installation cost.
- useful life and salvage value.
- operating and maintenance costs.
- use factors.

(4) Bottleneck machines are identified and duplicated, if economically justified, to reduce the inter-cellular moves. The production data and machine specifications are used as a basis for economic analysis of the duplication process.

(5) Model validation is performed by using the existing solution of a machine-component grouping problem.

(6) Sensitivity of results to similsrity measures is analyzed by considering several different similarity coefficients.

(7) Sensitivity of results to production volume and cost coefficients of material handling is analyzed.

The development of the model has led to the following findings.

(1) The results obtained by the model show that the problems of bottleneck machines and machine duplication can be effectively solved when the similarity coefficient method is used to form machine-component groups.

(2) The material handling costs associated with inter-cellular and intra-cellular moves in machine-component groups can be used as a basis for choosing a proper threshold value.

(3) The economic analysis of costs and benefits associated with duplication of bottleneck machines provides a logical basis for the decision about the purchase of additional machines. These additional machines, in some cases, are necessary to reduce the inter-cellular moves caused by the bottleneck machines.

(4) The solution procedures employed are very effective and efficient. Application of data storage and analysis techniques such as bit-level storage and bit manipulation techniques produce numerous benefits including:

  - The computer storage required for storing the data in the machine-component chart is reduced substantially.
  - The computation of similarity coefficients becomes easier.
  - The identification and duplication of bottleneck

machines are facilitated.

- The determination of inter-cellular and intra-cellular moves is done in a less complicated manner.

The development and use of the cost based machine-component grouping model provides the practitioners with an effective tool for forming machine cells. The detailed discussion of the model is given in Chapter III.

CHAPTER II

LITERATURE REVIEW

Machine-Component Grouping

Application of Group Technology to the manufacturing process starts with finding part-families and forming the associated machine cells. There are different approaches to the problem. Some are based on the data in route cards, while others use the machine-component chart to do this. Those methods of machine component grouping which rely on the systematic analysis of route card data are mostly derived from a production flow analysis approach developed by Burbidge (7). There are two approaches which use the data in the machine-component chart to cluster machines into cells. One of these approaches involves a permutation of the rows and columns of the machine-component chart. This approach is referred to as "machine-component group analysis." The second approach, called "similarity coefficient method" uses the pairwise similarity coefficients between machines to form the machine cells. A brief description of some of the algorithms based on these approaches is given in the subsequent sections of this chapter.

Production Flow Analysis

The first systematic approach to the problem of machine-component grouping was developed by Professor John L. Burbidge (7). The method named "Production Flow Analysis" uses the information in the route cards to find part-families and the associated groups of machines. As described by Burbidge, Production Flow Analysis is concerned with the manufacturing method only rather than with the design features and shape of the components. It is simple to understand and easy to implement (8, 40).

Production Flow Analysis consists of three phases. The first phase, termed "Factory Flow Analysis," deals with the assignment of parts and plants (factories) to major departments. In this phase, all parts having major differences in their manufacturing processes are separated and assigned to different departments. Varying plants are also divided among these departments. The aim of this phase is to divide parts and plants among different departments in such a way that all parts in one department can be completely processed within that department, so that there exists only one plant of a particular type in each department. In practice, this reduces to processing the maximum number of parts within one department, and having a minimum number of plants of the same type in each department.

The next step in Factory Flow Analysis is to prepare a Process Route Number (PRN) for each part. The PRN for each part is formed by putting together in proper sequence the number of departments visited by the part.

After PRN's are formed, the number of route cards for each PRN is determined, and a PRN frequency chart is developed. A PRN frequency chart simply shows how many route cards exist for each PRN. Based on the information gathered so far, a flow chart system, which shows the flow of materials among the departments, is drawn. The number of parts flowing along each path is obtained from the PRN frequency chart and is depicted on the materials flow chart.

The last step in this phase is to find the exceptional parts and to modify the material flow chart to achieve a simplified flow of materials among the departments. The exceptional parts are those parts having PRN's which are not compatible with the majority of components. These exceptional parts are usually a small percentage of the total production, and by modification of their processes or elimination of them from the manufacturing schedule, a more simplified flow of materials will be achieved (7). After the elimination of exceptions, some of the departments will be combined in successive steps to simplify the flow of materials.

The second phase of Production Flow Analysis is termed "Group Analysis." In this phase, the components with similar processing requirements are grouped together to form the families of similar components. Then, the machines capable of processing these components are assigned to these families to form the production cells. The main purpose is to form production cells in such a way that all components of a family can be completely processed within a single cell, and so that one machine of a particular type exists in each cell.

As in the case of Factory Flow Analysis, a PRN for each part is developed, but for this case they are formed by putting together the machine numbers visited by each part, rather than the department numbers. The final result of this step is a machine-component chart, which shows the machines or the workstations required by each component. Burbidge has explained how, by manual manipulation of the information in a machine-component chart, the family of similar components and the associated machines can be found. Since the time he suggested his manual method, many computer algorithms based on row and column permutations have been developed. Some of these algorithms will be discussed later.

The last phase of Production Flow Analysis is "Line Analysis." In this phase, the flow of materials and the sequence of the operations for each group are analyzed to find the best layout for each group of machines.

The underlying assumption for applying Production Flow Analysis is that the majority of components and machines in a factory belong to well established and clearly defined families and groups, and the problem is to find these families and groups. In practice, however, Production Flow

Analysis does not always result in the formation of such clearly defined families and groups. There are some components that cannot be processed completely within a single production cell (exceptional parts).

There are also some machines which are used by a large number of components and are needed by more than one group (bottleneck machines). As mentioned by J. R. King in his 1982 Survey of Group Technology (40), the application of Production Flow Analysis in general, and the problem of exceptional parts, in particular, need a good deal of subjective evaluations, which, in turn, requires a great deal of knowledge about the details of the process to which Production Flow Analysis is applied. Also, the manual evaluation of the machine-component chart, as suggested by Burbidge, becomes increasingly difficult as the number of components and machines increases.

Burbidge, in his paper in 1973 (8), mentioned some of these difficulties and suggested a method which he believed to be the most effective way of forming machine-component groups. This method is called "Nuclear Synthesis" and is based on forming the initial cells or nuclei by choosing machines used by a few components, and then successively adding to them those machines which have the smallest number of components assigned to other groups. After the process is completed and the groups are formed, some of the groups are modified, and others combined, until the required number of groups is obtained. One of the serious questions about this method is the way in which the required number of groups is defined. This number is determined based upon a factor called "Sociologically Accepted Size," but there is actually no concrete basis for such a factor.

There are two other important approaches to machine-component formation based upon the Production Flow Analysis concept. One is Component Flow Analysis and the other is Production Flow Synthesis.

Component Flow Analysis, like Production Flow Analysis, has three phases of analysis. In the first phase, all components are classified and sorted according to their manufacturing requirements; then rough groups of machine-components are formed by using the sorted list of components and taking into consideration the nature of manufacturing processes. Finally, by detailed analysis of loading and flow of materials and by making the required adjustment, a group layout is designed (40).

Production Flow Synthesis has been developed by De Beer and De Witte (16) to extend the concept of Production Flow Analysis to consider the problem of machine duplication and different machine characteristics. The major difference between this method and the other two methods discussed earlier is that in the former case, many components require more than one cell to be processed.

Machine-Component Group Analysis

As mentioned before, there are many clustering algorithms which, by analysis of the machine-component chart, find the part-families and form the associated machine cells. These algorithms, basically use heuristic techniques and convert the machine-component chart to a block diagonal form by exchanging the rows and columns of the related matrix in an iterative process. Some of these algorithms are briefly described below.

Iri's Algorithm

One of the simplest of such algorithms was developed by M. Iri in 1968 (39). The method begins with any arbitrary row in the machine-component matrix and masks all the columns which have an entry in this row, then continues the process by masking all the rows having entries in these columns. This process is repeated until there is no further increase in the number of rows and columns. The masked rows and columns constitute a machine-component group. If the matrix cannot be divided into clearly defined groups, it will be masked as one group. The major limitation of this algorithm is that it cannot deal with the exceptional components, which are encountered in most real world situations.

Bond Energy Algorithm

McCromic et al. (45) have developed a general cluster analysis technique called the Bond Energy Algorithm (BEA). The algorithm seeks to maximize a measure of effectiveness called Bond Energy by rearrangement of the rows and columns of the machine-component matrix.

The Bond Energy of a matrix of m rows and n columns is defined as:

$$\text{B.E.} = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{i,j}[d_{i,j+1} + d_{i,j-1} + d_{i+1,j} + d_{i-1,j}]$$

where, $d_{i,j}$ is an entry (0 or 1) in the ith row and jth column, and $d_{o,j} = d_{m+1,j} = d_{i,o} = d_{i,n+1} = 0$

The algorithm selects an arbitrary column as a starting point and places each of the remaining columns to the left and to the right of this column and measures the incremental contribution, in terms of Bond Energy, for each of them. Then, the column with the largest incremental contribution is selected as the next entry. The process continues for all the columns and is repeated for all the rows in the machine-component chart. When the process is complete, a block diagonal form is achieved, if one exists.

Rank Order Clustering Algorithm

Another clustering technique called Rank Order Clustering algorithm (ROC) has been developed by J. R. King in 1980 (39). The ROC algorithm works as follows: the row entries (0,1) of the machine-component chart are treated as binary numbers, and are ranked according to their binary values in descending order. Then, the rows are rearranged according to their associated ranks. The same process is repeated for the columns of the chart. At the end, a block diagonal is formed. The major limitation of the algorithm is the fact that most computers cannot process a number larger than $2^{48} - 1$; therefore, the maximum number of rows or columns that can be handled by the algorithm is limited to 47.

To overcome the problem of the limit on the matrix size, King has suggested that the rows and columns be ranked based upon entry-by-entry comparison. To compare two rows, one begins from the left and compares the first entry of one row with the same entry of the second row, and continues until the entry in one row is different from the related entry in the second row. The comparison of the two rows is then ended, and the row with the larger entry is ranked higher than the other one. This process is repeated for all rows and columns. In most cases, only a few comparisons are required to determine which of the two rows has a higher rank. Though the procedure overcomes the problem of large binary numbers, it has a computational complexity of cubic order (40).

King has developed a new ROC algorithm which employs special data structure techniques, such as linked list and hash tables to overcome some of the problems associated with the previous ROC algorithm. The new algorithm is more suited for dealing with sparse matrices which are very common in group technology applications. The new algorithm also reduces the computer storage required as well as the computational complexity of the previous one. The detailed discussion of the new algorithm is given by King in his recent paper (40).

Similarity Coefficient Method

The similarity coefficient method has been developed in the field of numerical taxonomy.[1] It was introduced to the area of production, for the first time, by John McAuley, who applied this method to the problem of machine-component grouping (44). Since then, it has been used by other researchers and has proved to be an effective tool in machine-component grouping applications (14, 16, 18, 54). The method is based on the similarity coefficients between two machines. This similarity coefficient is defined by McAuley as "the number of components which visit both machines divided by the sum of components which visit one or the other of the machines." To illustrate the point (Figure 2), the similarity coefficient between machines A and B is 4/5 = 0.8; between B and C is 1/8 = 0.125; and between B and D is zero. To form the machine cells, a similarity matrix containing the pairwise similarity coefficients of all machines is constructed. Then, this matrix is used by the clustering algorithm to group similar machines into cells.

McAuley has used a single linkage cluster analysis

---

[1] Numerical Taxonomy means: "The theoretical study of classification, including its bases, principles, procedures, and rules" (Simpson, 1961, p. 11).

algorithm to form machine-component groups. This is one of several algorithms based on clustering techniques. The cluster analysis technique will be discussed in Chapter IV.

		Components							
		1	2	3	4	5	6	7	8
Machines	A	1	1	1	1	1			
	B	1	1	1	1				
	C				1	1	1	1	1
	D					1	1	1	

Figure 2. Machine-Component Chart

Single Linkage Cluster Analysis (SLCA)

This method was developed by Sneath in 1957 (58) and is the simplest of all clustering methods. In this method, the similarity coefficient between two groups is the similarity coefficient between the two most similar machines in these groups. Most algorithms based on SLCA work as follows:

1. A similarity matrix containing all the similarity coefficients associated with all pairs of the data set is constructed and stored in the primary storage of the computer. (In some algorithms the data set is stored and the similarity coefficients are calculated on request by the clustering algorithm. This saves some computer storage at the cost of more computations. In

some cases this trade-off is necessary, since, as the number of entries in the data matrix increases, the size of the similarity matrix increases exponentially.)

2. The similarity matrix is scanned by the clustering algorithm, and the pair with the highest similarity coefficient is chosen as the initial cluster.
3. The similarity level is lowered and all the clusters or entities with a similarity coefficient larger than this level are grouped together.
4. Steps two and three are repeated until all clusters join together and make a single cluster containing all the entities.

At each similarity level there are a number of clusters in which each member has a similarity coefficient with at least one other member of the cluster greater than or equal to the similarity level. These clusters will join together in subsequent steps to make clusters at lower similarity levels.

The results of a clustering algorithm can be best illustrated by a type of tree diagram called "dendogram". At the lowest level of a dendogram, each branch represents one entity (machine). Moving up toward the root of the dendogram, the branches merge into new ones representing clusters with larger sizes. The root of the tree represents a cluster encompassing all the entities. A similarity scale can be used with the dendogram to show the clusters associated with each similarity level.

The application of SLCA to machine-component grouping can be best illustrated by an example: for this purpose, consider the machine-component chart in Figure 3.

		Components											
		1	2	3	4	5	6	7	8	9	10	11	12
Machines	1	1	1	0	0	0	0	0	0	0	0	0	0
	2	1	1	0	0	0	0	0	0	0	0	0	1
	3	0	0	1	1	0	0	0	1	0	0	1	0
	4	0	0	1	1	0	0	0	1	1	0	1	0
	5	0	0	0	0	0	0	1	0	1	1	0	1
	6	0	0	1	1	0	0	0	1	0	0	1	0
	7	0	0	0	0	0	0	1	0	1	1	0	0
	8	0	0	1	1	0	0	0	1	1	0	0	0
	9	0	0	0	0	0	0	1	0	1	1	0	0
	10	1	1	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	1	1	0	0	0	0	0	0
	12	0	0	0	0	1	1	0	0	0	0	0	0

Figure 3. Machine-Component Chart

Based on the data in Figure 3 the similarity matrix can be constructed as in Figure 4.

By examining the similarity matrix, the clusters formed at each similarity level are found (Figure 5). At the 100 percent similarity level, machines 3 and 6 form the first

	Components											
Machines	1	2	3	4	5	6	7	8	9	10	11	12
1	-											
2	.67	-										
3	0	0	-									
4	0	0	.8	-								
5	0	.17	0	.13	-							
6	0	0	1	.80	0	-						
7	0	0	0	.14	.75	0	-					
8	0	0	.6	.80	.14	.60	.17	-				
9	0	0	0	.14	.75	0	1	.17	-			
10	1	.67	0	0	0	0	0	0	0	-		
11	0	0	0	0	0	0	0	0	0	0	-	
12	0	0	0	0	0	0	0	0	0	0	1	-

Figure 4. The Similarity Matrix for Figure 3

cluster; machines 1 and 10 form the second cluster; machines 7 and 9 form the third cluster; and machines 11 and 12 form the fourth cluster. At this similarity level, all other machines remain single. At the 75 percent similarity level, machines 4 and 8 join the first cluster, and machine 5 joins the third cluster. At the 50 percent similarity level machine 2 joins the second cluster. At the 25 percent similarity level the clusters remain unchanged, and at zero similarity level all clusters join together and make a single cluster encompassing all machines. The dendogram for this clustering problem is illustrated in Figure 5.

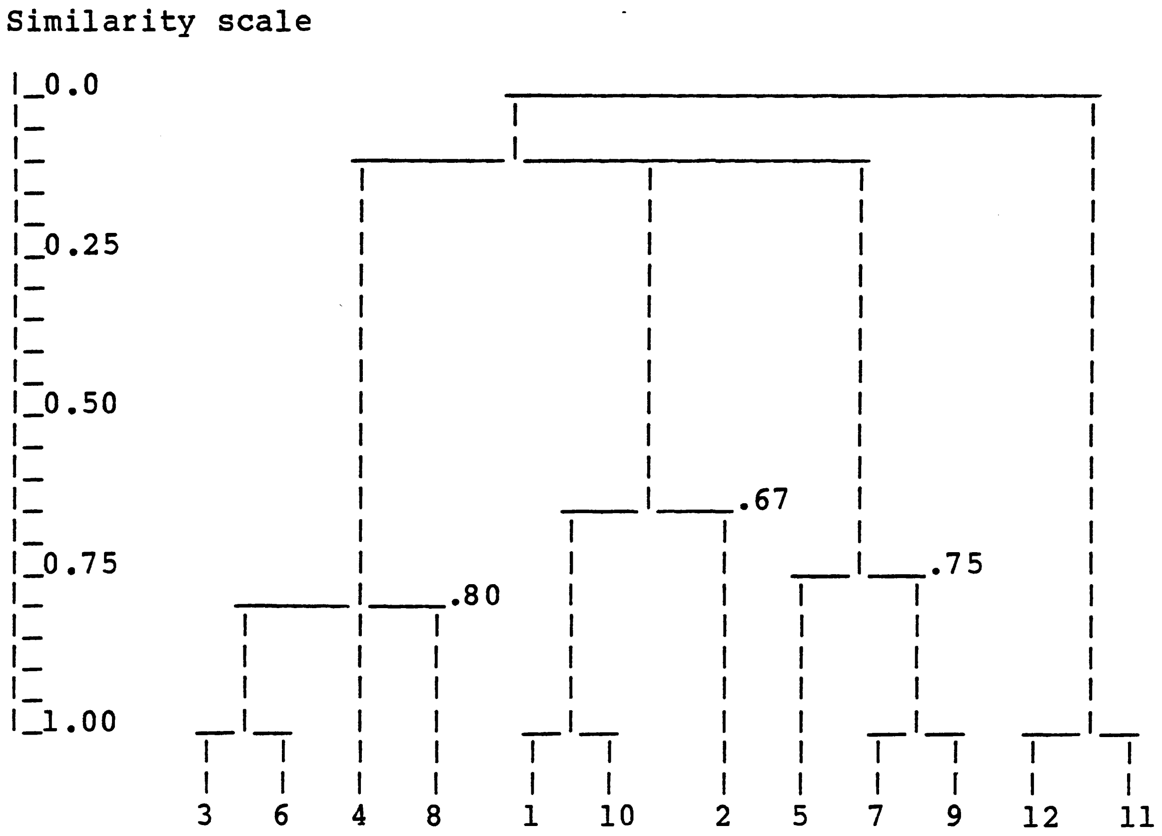


Figure 5. Dendogram for the Example Problem

The dendogram shows that at the 100 percent similarity level there are eight clusters: (3,6), (4), (8), (1,10), (2), (5), (7,9), (12,11). If the similarity is lowered to 75 percent, there are five clusters: (3,6,4,8), (1,10), (2), (5,7,9), and (12,11). At the 50 percent similarity level, the number of clusters reduces to four; and at the zero level, there is only one group. This means we have a set of solutions rather than a single one. Here, five similarity levels have been considered, and five different solutions have been obtained (this is one of the problems associated with the similarity coefficient method).

As argued by John McAuley (44), finding the best solution from a set of given solutions in this type of problem is not an easy task. It requires some criteria, based upon which the number of groups can be determined. The number of groups, of course, depends upon the similarity level called "the threshold value." Many factors affect the choice of the threshold value. Some of these factors are: the number of inter-group/intra-group movements, the percent of machine utilization, machine duplication (assignment of one machine type to several groups), planning and control problems, as well as managerial considerations. Not all these factors can be quantified and used in the calculation of the threshold value; only the most concrete and important factors are considered. The problem of finding the optimum number of groups will be discussed in the next chapter.

Graph Theoretic Method

Before ending this chapter, it is worthwhile to mention some other clustering methods based upon the similarity coefficient. One such method is a graph-theoretic method developed by Rajagopalan and Batra (54). In this method, each machine is represented by a vertex; and the similarity between two machines is represented by an arc. Maximal collection of vertices, in which each pair is connected by an edge, is called a "clique." The graph-partitioning approach is employed to classify the vertices.

The method uses the similarity coefficient introduced by McAuley. Based upon this similarity coefficient, a similarity matrix is constructed. The machine graph is drawn by examining the similarity matrix, and connecting the pairs of vertices having a similarity coefficient greater than the threshold value.

As in the case of previous clustering method, the process of choosing the threshold value is a complicated one. In the discussion by Rajagopalan and Batra, if the threshold value is too large, the related graph is sparsed. This implies that only the effect of a few machines has been considered in the process of machine-component formation. On the other hand, if the threshold value is too small, there will be a very dense graph in which the effects of the majority of machines are included.

Another consideration in choosing the threshold value is the sensitivity of the solution to the variations in the input data. To consider this sensitivity, variations in the number of edges are plotted against the threshold value. Then, the threshold value related to the flat part of the graph, or part of the graph with minimum slope, is chosen. The major limitation on the application of the graph-theoretic method is that as the number of vertices increases the number of cliques increases exponentially and the partitioning process becomes very complicated.

Another clustering method based on the similarity coefficient has been developed by De Beer et al. (16); and De Beer and De Witte (18). They distinguish three types of machines: primary, secondary, and tertiary. Then they develop three kinds of similarity coefficients related to these three types of machines. The arcs between the vertices are drawn based upon these three coefficients. The procedure is similar to the method developed by Rajagopalan and Batra.

CHAPTER III

COST BASED MACHINE-COMPONENT GROUPING MODEL

In the previous chapter the machine-component group analysis and similarity coefficient method were discussed as the two major approaches to the machine-component grouping process. Due to the heuristic nature of the algorithms based on machine-component group analysis, the final block diagonal form produced by them may well depend upon the initial arrangement of rows and columns in the machine-component chart. That means, several solutions may be obtained by changing the initial arrangement of the chart. Also, if there are some exceptional parts, which require operations in more than one cell, these algorithms fail to form any block diagonal form before dealing with these exceptional parts. These two problems can be overcome by using the similarity coefficient method.

The similarity coefficient method forms the machine cells based on pairwise similarity coefficients of the machines involved, and gives an analytical solution which depends on the initial machine-component chart (not on the arrangement of rows and columns) and on the type of similarity coefficient used. The algorithms based on the similarity coefficient method can form the machine cells prior to a consideration of any exceptional part.

Although the similarity coefficient method can overcome the two problems mentioned before, it has its own limitations. The clustering algorithms based on the similarity coefficient method have been developed in the field of numerical taxonomy and then adopted by engineers and applied to the machine-component formation process. These algorithms, basically, look for patterns in data sets and cluster together the closely related data elements. However, in manufacturing, the data elements are machines with different processing capabilities and prices. A production cell is not merely a collection of similar machines, but a workstation capable of processing a set of components at a certain cost. As a result, clustering algorithms lack many capabilities, which, if they existed, would improve the machine-component grouping process to a great extent. Basically, the clustering algorithms applied to the machine-component grouping process must be capable of dealing with machines, parts, and the production system; this should be the major concern in the development of any new model.

Part-Families and Machine Cells

The purpose of the machine-component grouping process in Group Technology is to form a set of mutually exclusive machine cells, each capable of processing all operations of one or more part-families assigned to it. In this work, three major problems will be carefully studied:

(1) The clustering algorithm

(2) The threshold value

(3) The bottleneck machines.

The Clustering Algorithm

Clustering algorithms are used to bring similar machines together. There are different clustering algorithms capable of doing this job. McAuley, who has introduced the clustering method to the field of production, uses a SLCA algorithm to form machine cells.

As discussed earlier, SLCA uses a single linkage to cluster together similar machines or groups of machines. In this approach, the similarity coefficient between two groups is the similarity coefficient between the two closest members of the two groups. As a result, the groups may join together merely because two of their members are similar. This problem is referred to as "chaining", and in some cases, where the groups are not well separated, may create serious dilemmas. Due to the chaining effect, while two groups may join together on the basis of a single linkage, the majority of their members may be quite far apart in terms of similarity. This is the major drawback of the SLCA method. Therefore, SLCA is not a reliable choice for the machine-component grouping process. A better choice is a method which uses the overall similarity between all members of two groups as a basis for calculating the similarity coefficient between them. The Average Linkage Clustering (ALC) method has this property. In this method, the similarity coefficient between two groups is defined as the average of the similarity coefficients of all pairs involved. The new model uses a clustering algorithm based on the average linkage clustering method to form the machine cells. A detailed discussion of this method is given in Chapter IV.

Another problem, which may be encountered when the similarity coefficient method is used, is that some machines may not be assigned to the cell whose members have the largest number of common operations with them. Let us call this problem "improper machine assignment," and use an example to clarify it. Consider the machine-component chart in Figure 6. The similarity matrix for this machine-component chart is constructed and given in Figure 7. From Figure 7, the similarity coefficient between machines B and C is 0.5, while between C and D it is 0.44, though C has more common parts with D than with B. Based on this similarity matrix, with the threshold value of 0.5, machines A, B, and C form the first group, while machines D and E form the second. With this arrangement, there are four inter-cellular moves relating to parts 3, 4, 5, and 6, which have operations in both cells. However, with the reassignment of machine C to the second cell, the number of inter-cellular moves can be reduced to three, which are caused by parts 3, 4, and 5. The machine-component charts of the two cases are shown in Figures 8 and 9. An asterisk has been used to show the operations which create inter-cellular moves.

		Components										
		1	2	3	4	5	6	7	8	9	10	11
Machines	B	1	1	1	1	1						
	A	1	1	1								
	C			1	1	1	1					
	D			1	1	1	1	1	1	1	1	1
	E				1	1	1	1	1	1		

Figure 6. A Machine-Component Chart

		Machines				
		A	B	C	D	E
Machines	A	-				
	B	.60	-			
	C	.17	.50	-		
	D	.09	.28	.44	-	
	E	0	.22	.43	.67	-

Figure 7. The Similarity Matrix for Data in Figure 6

		Components										
		1	2	3	4	5	6	7	8	9	10	11
Machines	B	1	1	1	1	1						
	A	1	1	1								
	C			1	1	1	1					
	D			*	*	*	*	1	1	1	1	1
	E				*	*	*	1	1	1		

Figure 8. Machine-Component Chart (3, 4, 5, and 6 are Exceptional Parts)

		Components										
		1	2	3	4	5	6	7	8	9	10	11
Machines	B	1	1	*	*	*						
	A	1	1	*								
	C			1	1	1	1					
	D			1	1	1	1	1	1	1	1	1
	E				1	1	1	1	1	1		

Figure 9. Machine-Component Chart (3, 4, and 5 are Exceptional Parts)

As can be seen, the machine-component grouping based on the similarity coefficient method in some exceptional cases does not produce a satisfactory result. This fact has been considered in the development of the new model. By using a clustering algorithm based on ALC, the chance of occurrence of such exceptional cases is reduced substantially. This is true because all similarity coefficients between machines in two groups are used to determine the average similarity coefficient. When several similarity coefficients are involved, it is possible that a few individual similarity coefficients result in improper assignments. But it is quite unlikely that the average similarity method will terminate with improper assignment of machines to groups involved.

In the instance where ALC results in improper assignment of machines, they can easily be detected when the bottleneck machines are identified. This is due to the fact that only bottleneck machines create inter-cellular moves and only these machines can be improperly assigned. When the bottleneck machines are identified, it is possible to determine how many exceptional parts from each cell are processed on a specific bottleneck machine, and to assign this machine to the cell which has more exceptional parts than any other cell.

Threshold Value

Due to the nature of clustering algorithms, the number and size of the machine cells formed in the machine-component grouping process depend upon the similarity level (threshold value) used to form the cells. If the threshold value is high, there will be a large number of cells of small size. On the other hand, if the value is low, there will be only few groups of large size. In all the previous models based on the similarity coefficient method, the selection of the threshold value is, to some extent, arbitrary (40). In practice, however, there are several factors affecting the size and number of the machine cells. Some of these factors are: the number of inter-cellular trips, the number of intra-cellular trips, planning/control problems, and managerial considerations. Ideally, it is desirable to construct a model in which all these factors are incorporated; practically, however, not all of them can be quantified and used in one model.

Two of the most important elements affecting the size of the machine cells are the number of inter-cellular and intra-cellular trips. As discussed before, in most cases, it is not possible to process all components of a part-family within a single cell. As a result, there are a number of parts requiring movement from one cell to another for different operations. As a cell grows in size, a larger number of components can be processed within it, and fewer parts require to be processed in more than one cell. In the extreme case, where all machines are assigned to a single cell, no inter-cellular trips exist. On the other hand, as a cell increases in size, the number of intra-cellular trips increases. Therefore, there should be a kind of costs trade-off between inter-cellular and intra-cellular trips which can be used in determination of the threshold value.

Dr. McAuley has tried to use this type of cost trade-off as a basis for determination of the number and size of machine cells. In his model, he calculates the number of inter-cellular and intra-cellular moves. Then, he assigns different costs to each type of move and calculates the total cost. Finally, a total cost for each threshold value is calculated; and the threshold value associated with the minimum total cost is selected. It can be seen that the model is not specific regarding the type of moves taking place, distances travelled, and cost per unit of distance. In fact, the model does not consider the material handling costs; it merely uses two different weights for the two types of moves involved. In the model developed here a from-to chart containing all inter-cellular trips is prepared. This from-to chart and a move-cost chart are used to determine the material handling costs for the machine cells associated with each threshold value.

Bottleneck Machines

A bottleneck machine is a machine which is required for a large number of parts from different cells. This machine creates a large number of undesirable inter-cellular moves. In practice, usually such machines do exist and require a special treatment in the process of machine-component formation.

Theoretically, it is possible to eliminate all the inter-cellular moves by assigning the required number of bottleneck machines to the cells which require them (duplication). If this happens, each cell will have all the machines required for the processing of all operations of part-families assigned to it. This is possible if the same number of machines required by conventional manufacturing can be used to form a set of mutually exclusive machine cells. In practice, however, in some cases a new machine must be bought. In doing so, an effective cost-benefit analysis of all related factors must be made prior to any decision regarding the assignment of a new machine to a cell.

Most of the problems discussed here were not considered by the previous models. The model presented in this chapter uses all the necessary information to identify the bottleneck machines, to duplicate them wherever warranted, and to provide an economic basis for each decision regarding the assignment of new machines to cells.

The Model

Overview

The cost based machine-component grouping model is a model designed to form the machine cells and assign to them the associated part-families in a Group Technology environment. It incorporates more realism into previous models by considering such factors as: the chaining problem of SLCA, improper machine assignment, bottleneck machines, and the selection of a proper threshold value.

The model employs the similarity coefficient method and seeks an analytical solution to the clustering phase of the machine-component grouping process. It also deals with the problem of bottleneck machines. This provides a practical basis for the application of the similarity coefficient method to the machine-component grouping process.

The model makes an economic analysis of the factors involved in the duplication process. This provides the user with information regarding the costs associated with acquisition of a new machine, and the reductions in material handling costs due to duplication of the bottleneck machine.

Finally, the model chooses several threshold values and forms the related machine cells. Then, for each of them it calculates the material handling costs of inter-cellular and intra-cellular moves. These costs can be used as a basis for selection of a threshold value which results in minimum total cost.

In addition, since the model uses the average clustering method to form the cells, the chaining problem of SLCA will be eliminated, and improper machine assignment will become less likely. The model also has the capability of detecting the improper assignments, and reassigning the related machines, if necessary.

To carry out the machine-component formation process while considering all the factors mentioned, a computer model composed of 30 routines has been developed. The model performs four major functions: (1) clusters together machines to form cells, (2) assigns part-families to the cells, (3) deals with bottleneck machines, and (4) calculates the material handling costs associated with each threshold value. The routines forming the computer program can be classified under four major units:

1. Clustering unit
2. Assignment unit
3. Bottleneck unit
4. Threshold value unit

A schematic representation of the four units is depicted in Figure 10.

The clustering unit uses the information in the machine-component chart to calculate the pairwise similarity coefficients of all machines. These similarity coefficients are stored in a table called "similarity matrix." The clustering algorithm uses the similarity matrix and groups together similar machines in an iterative process. When all machines are clustered, a dendogram is developed which shows the cells and the associated machines for each threshold value.

The second unit uses the information incorporated in the dendogram to identify the machine cells. The assignment unit uses the threshold values as an input and determines the cells and their associated machines. After the cells for a given threshold value are identified, the number of operations performed on each part in different cells is determined. Next, the part is assigned to the cell which is capable of processing a larger number of its operation than

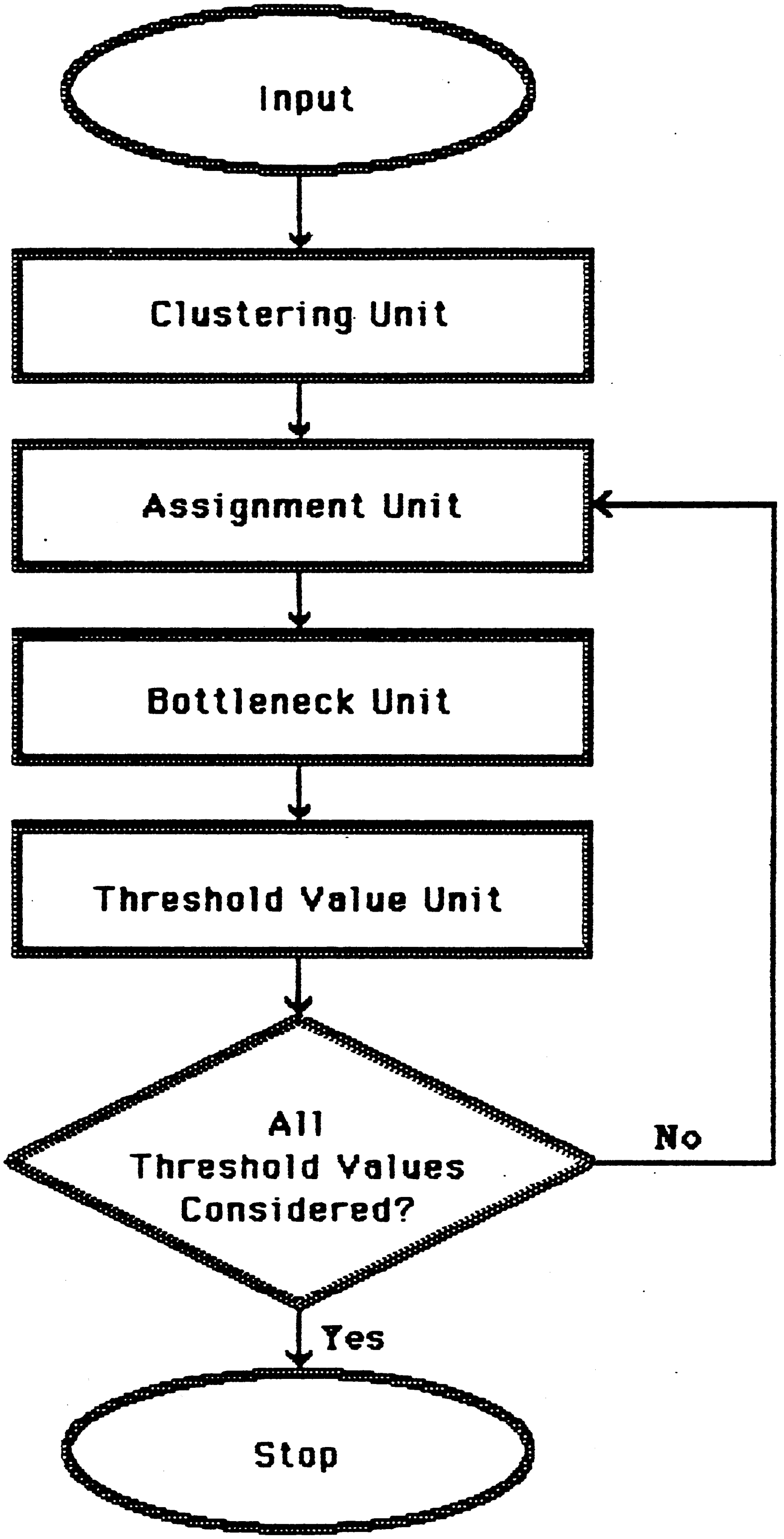


Figure 10. Machine-Component Grouping Model

any other cell. This unit, also, identifies the exceptional parts, and determines the number of such parts visiting each machine. This last step is crucial for identification of the bottleneck machines.

The bottleneck unit uses the information regarding the exceptional parts (obtained in the second unit) to identify the bottleneck machines. In fact, any machine which processes an exceptional part is a "bottleneck" machine. After the bottleneck machines are determined, the duplication process takes place. There are two cases in the duplication process. First, the bottleneck machines can be assigned to all cells requiring them, without buying any new equipment. Secondly, some additional machines should be acquired in order to complete all contemplated duplications. The latter case requires an economic analysis of all factors involved. For a given threshold value, the arrangement of machine cells is finalized at this step.

The fourth unit deals, mainly, with the selection of a proper threshold value. The output of the third unit is a threshold value with the associated machine cells. Unit four develops a from-to chart for these machine cells, and uses a facilities design algorithm such as CRAFT to determine the material handling cost of inter-cellular trips. For this purpose a move-cost chart containing the cost per unit distance travelled between different cells should also be input to the model. This unit also determines the intra-cellular costs. The material handling costs obtained in this unit can be used as a basis for determination of a proper threshold value. This is done by the selection of the threshold value associated with the minimum material handling cost.

Assumptions of the Model

The model is designed to form the machine cells which can be used in a cellular manufacturing system. This type of manufacturing is more appropriate where small lots are produced in a batch-type manufacturing system. In this type of system, reduction in setup time, time delays associated with part-travel and lot formation, and material handling cost are of major importance and concern. Since the result of the machine-component grouping process is used to set up a cellular manufacturing system, it is assumed that all data related to production schedules, machine requirements, and manufacturing processes are available. A summary of these assumptions is given here:

(1) Existing facilities meet the requirements of the production schedule.

(2) Information regarding the machine types, part types, and manufacturing of parts on machines can be obtained, i.e., the machine-component chart can be constructed.

(3) Production data are available. In this respect, the following items are important:

  (a) Production volume in terms of the number of each part to be produced over a specific period of time.

   (b) Processing time of each part on each machine.
   (c) Use factor for each machine.
   (d) Production hours per day.

(4) Specifications of the machinery are known. This includes:
   (a) Price and installation costs.
   (b) Useful life and the salvage value.
   (c) Operating and maintenance costs.

Taxes have not been included in the economic analysis.

The whole machine-component formation process depends upon the information in machine-component charts. Therefore, the second assumption must hold if any cell is going to be formed at all. The first, third, and fourth assumptions relate to duplication of bottleneck machines. If these assumptions are not valid, the duplication can still be done, but no economic justification can be provided to support such a duplication.

Inputs to the Model

The model performs three major functions to complete the machine-component formation process: clusters machines into cells, duplicates the bottleneck machines, and determines the material handling cost of inter-cellular and intra-cellular trips for a given threshold value. Each of these functions has its own input requirements. The following data are required for the clustering function:

(1) Number of machine types.

(2) Number of part types.

(3) Processing requirements of each part.

(4) Initial threshold value - the similarity level that is used to form a set of machine-component groups.

(5) Number of similarity levels. This number is necessary for the construction of the dendogram. The levels of the dendogram and the step size by which the threshold value is reduced depends upon this number. Since the range of a similarity coefficient is between 0 and 1, the following expression can be written:

$$STEP = (1-0)/NSTEP \qquad (3-1)$$

where

STEP = the step size and
NSTEP = the number of similarity levels.

The duplication process requires the following data items:

(1) Number of each part to be produced on each machine.

(2) Processing time of each part on each machine.

(3) Use factor for each machine.

(4) Working hours per day.

(5) Price, installation cost, useful life, salvage value, and the required rate of return on investment.

(6) Operating and maintenance costs of machinery.

Finally, to calculate the material handling costs of inter-cellular moves, a facilities design algorithm (like CRAFT) is used. For this, a move-cost chart and the area requirements of cells in the initial layout should be input requirements of cells in the initial layout should be input to the model. An outline of CRAFT is given in Appendix A.

Outputs of the Model

The model provides a solution that can be used to set up a group layout in a cellular manufacturing system. For this purpose, the following items are prepared:

(1) Threshold value associated with the minimum total material handling cost.
(2) Number of machine cells.
(3) Number of machines in each cell.
(4) Material handling cost.
(5) Lists of machines in each cell.
(6) Lists of parts in each cell.
(7) Number and list of additional machines required for cell formation process.

Algorithmic Flow Chart of the Model

The major functions of the model can be depicted in an algorithmic flow chart as in Figures 11-a, 11-b, 11-c, and 11-d. The flow chart shows the major functions of the model and the sequence in which they are performed. A computer program consisting of 30 routines has been developed to perform these functions. Each function is represented by one block in the flow chart and relates to one or more routines in the computer program. The flow chart can be

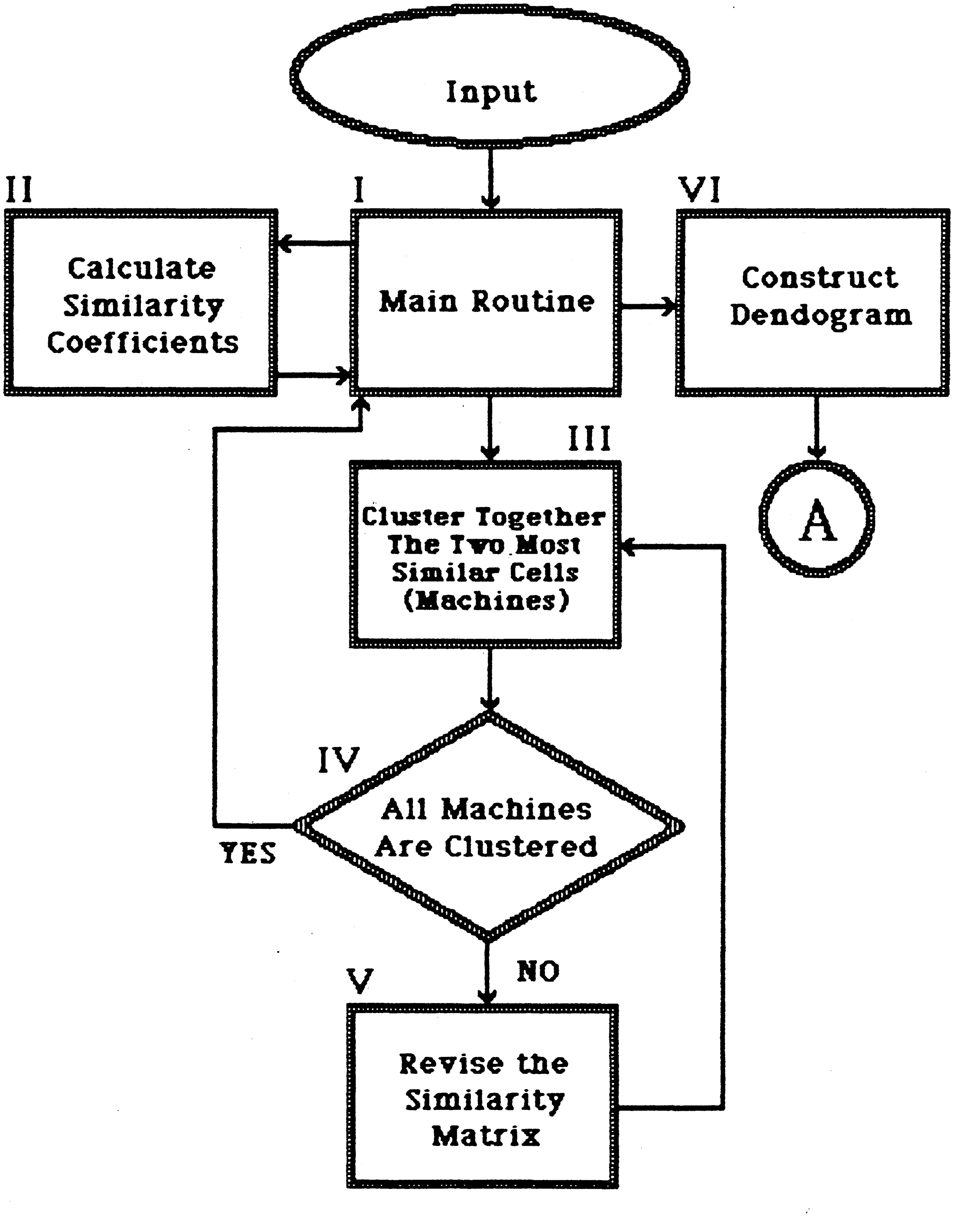


Figure 11. Algorithmic Flow Chart of Clustering Unit

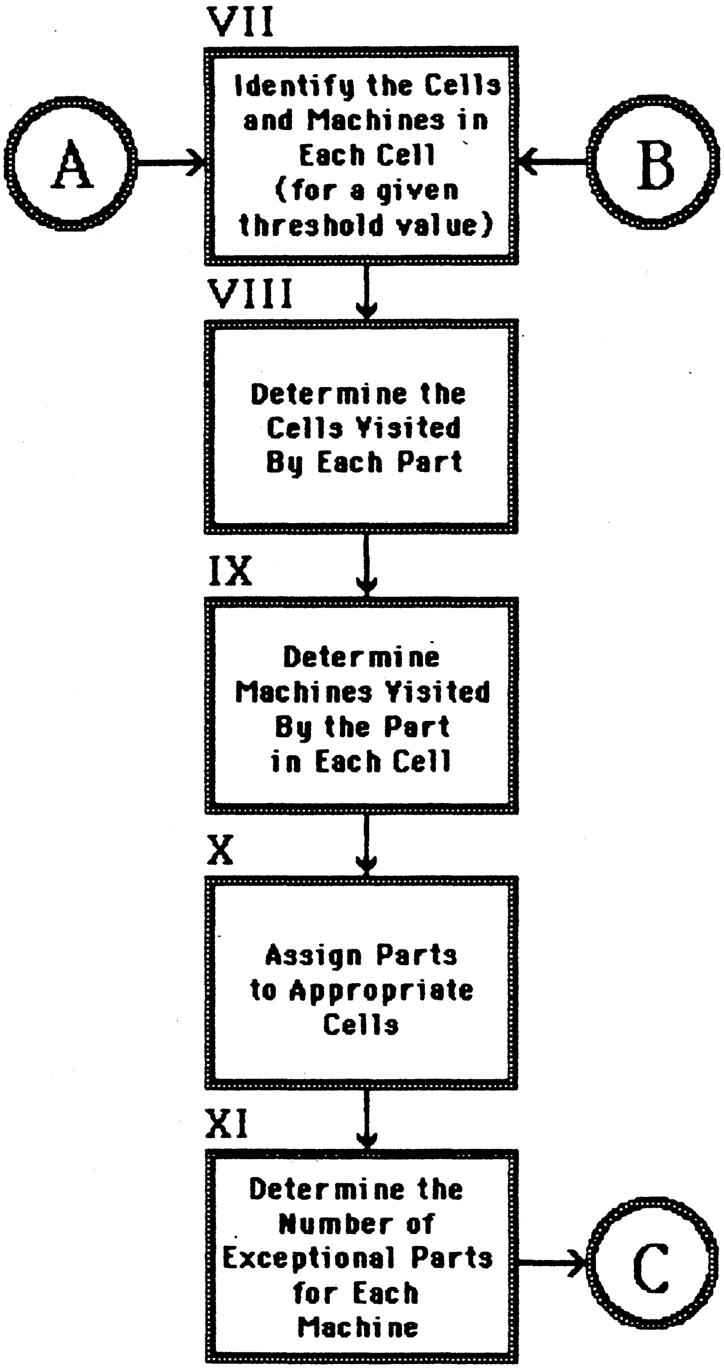


Figure 12. Algorithimic Flow Chart of Assignment Unit

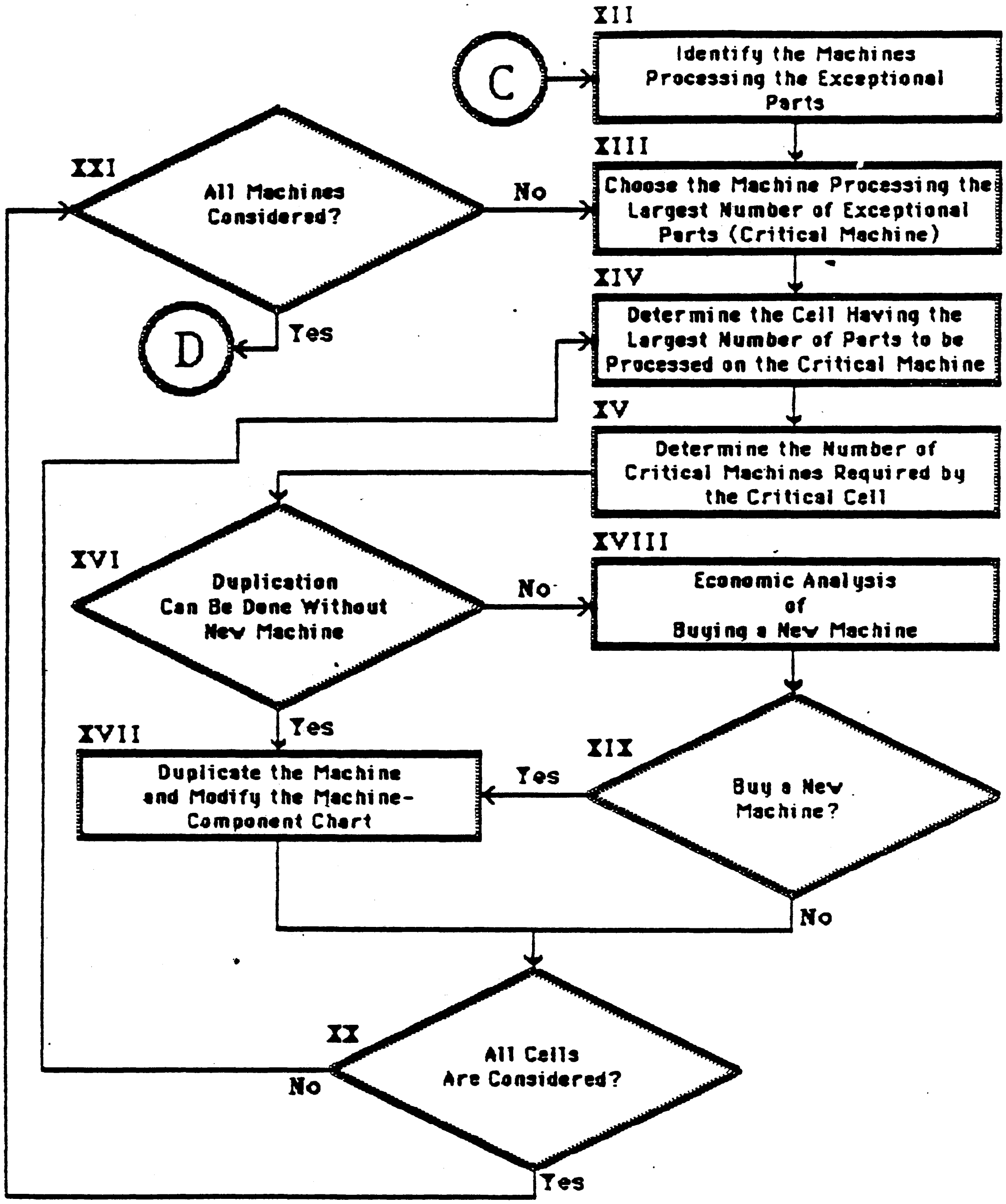


Figure 13. Algorithmic Flow Chart of Bottleneck Unit

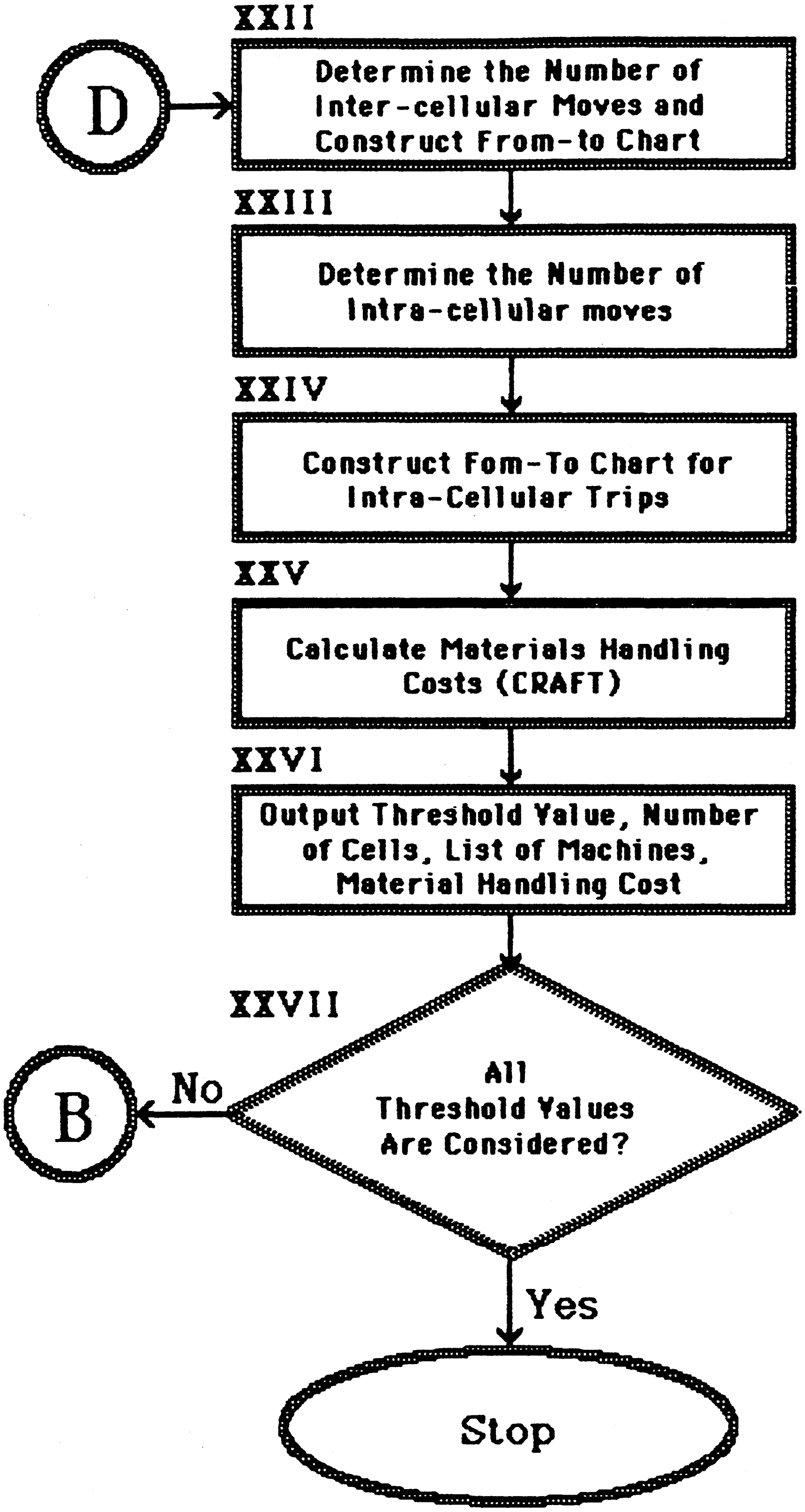


Figure 14. Algorithmic Flow Chart of Threshold Value Unit

divided into four major divisions in relation to the four major units of the model discussed earlier (Figure 10). The glossary of variables and Fortran codes of the program are given in Appendixes B and C. A brief description of each function follows.

The Clustering Unit

The four major functions of the clustering unit have been represented by blocks II, III, V, and VI in Figure 11. The main inputs to this unit are: an initial threshold value, the number of similarity levels, and the machine-component chart. The output of the unit is a dendogram showing the machine cells formed at each similarity level. The dendogram contains all the information needed to determine the machine cells and list the machines in each cell.

The first block in this unit relates to the main routine of the computer program. This routine is an administrative routine which establishes lines of communication among other routines of the computer program. It receives the external inputs as well as the outputs of different routines and provides the necessary information required by other units. Most routines of the clustering unit have been developed based on existing clustering algorithms (2).

Block II of this unit relates to the SMLTY routine in the computer program. This routine calculates all pairwise similarity coefficients between machines in the machine-component chart. For this purpose, the similarity coefficient defined by McAuley is employed. The similarity coefficients calculated by SMLTY are stored in the similarity matrix and are accessed by the clustering routine to form the machine cells. SMLTY receives the machine-component chart as its input and prepares a similarity matrix as its output. The detailed discussion of similarity coefficients is given in Chapter IV.

Block III in Figure 11 relates to the CLSTR routine of the computer program. This routine uses the similarity matrix prepared by SMLTY as its input and clusters together the two most similar machines as the first cell. Since the average linkage clustering method is used, it is necessary, to revise the whole similarity matrix and recalculate the new similarity coefficients between the existing cells (machines) and the newly formed cell. The similarity matrix is revised by routine REVIS. The revised similarity matrix is again searched by routine CLSTR and the two most similar cells (machines) are grouped together to make a new cell. The revision of the similarity matrix and clustering process continue until all machines are grouped into machine cells. Since at each iteration two cells join together, then M machines will merge in M-1 iterations. The clustering algorithm records all information related to each iteration for each pair. This includes the iteration at which the merge occurs, the cells (machines) involved, the similarity coefficient of the merge, the last iteration at which any of the two cells were involved in a merge, and the next iteration at which the new cell will merge. All these data are necessary for construction of the dendogram which is the only output of the TREE routine.

Block V of the clustering unit deals with revision of the similarity matrix. The revision process is done by the REVIS routine. Any time two cells merge and a new cell is formed, it is necessary to recalculate the new similarity coefficients between the new cell and all other existing cells and enter them into the similarity matrix. Based on the average linkage clustering method, the similarity coefficient between two groups i and j is calculated as

$$S_{ij} = s_{ij}/(N_i \cdot N_j) \qquad (3.2)$$

where

S_{ij} = the similarity coefficient between groups i and j

s_{ij} = the sum of all pairwise similarity coefficients between machines in groups i and j

N_i, N_j = the number of machines in groups i and j, respectively.

The average linkage clustering method will be discussed in more detail in chapter IV.

Block VI in the clustering unit relates to the TREE routine of the computer program. This routine uses the outputs of CLSTR and prepares a dendogram which shows the machine cells formed at each similarity level. The

(missing)

in any given cell. After the dendogram is constructed, for a given threshold value, the machine cells and the associated machines in each cell can be identified. But still there are three more steps to be taken before the arrangement of the machine cells can be finalized. First, the lists of machine cells and machines in each cell should be prepared; and part-families should be assigned to the associated cells. Secondly, the bottleneck machines should be determined and the duplication process be performed. Finally, the material handling costs associated with different arrangements of the machine cells (for different threshold values) should be determined. Unit assignment, bottleneck, and threshold value relate to these three steps.

Assignment Unit

The major functions of the assignment unit are presented in blocks VII, VIII, IX, X, and XI in Figure 11-b. Block VII of this unit relates to the CELLS routine of the computer program. This routine uses the information provided by the dendogram and prepares the list of machine cells for each similarity level. It also identifies all machines associated with each cell. Routine CELLS prepares, analyzes, and stores a lot of information regarding machine cells and their associated machines. This information is very crucial for dealing with bottleneck machines and choosing a proper threshold value. This routine stores the list of all machines and their associated cells. It also keeps a record of the machines in each cell. As a result, for each cell, the list of machines assigned to it can easily be prepared. It is also possible to determine the cell to which a specific machine belongs. These two pieces of information are necessary for determining the intercellular moves created by the machine requirements of the exceptional parts. The procedures used in this routine will be discussed in Chapter IV.

Block VIII of this unit relates to routines ASSGN and BOTLK of the computer program. Routine ASSGN identifies the cells visited by each part by determining the cells which process the part for some of its operations. Routine BOTLK receives its inputs from CELLS and passes them to ASSGN. The detailed discussion of the procedures used in these routines will be given in Chapter IV.

After the cells visited by a part are identified, ASSGN determines all machines in each cell which are required for the processing of that part. The number of such machines for each part, NOP, is determined and recorded (block IX).

Block X in Figure 12 represents the part assignment function. The assignment of parts to machines would have been very easy if there were not any parts requiring operations in more than one cell. In practice, however, the possibility of existence of exceptional parts cannot be dismissed. As a result, special care is needed in the assignment process. Any exceptional part should visit all the cells in which it has an operation. It seems such a part can be assigned to any of these cells, because it should visit all the cells any way. For the purpose of machine duplication, however, the assignment should be done in such a way that it simplifies the duplication process. For this reason, each part should be assigned to the cell which can perform a larger number of its operations than any other cell. In this way, since the part has fewer operations in any other cell than in its own cell, fewer machines are required to be duplicated in order to reduce the inter-cellular moves. To illustrate this point, consider the machine-component chart in Figure 15. In this chart machines A, B, and C belong to the first group; while machines D and E belong to the second group. In this chart part 4 is an exceptional part having three operations on machines A, B, and C in the first group and one operation on machine D in the second group. If this part is assigned to the first group, machine D must be duplicated in this group to eliminate the inter-cellular moves between the two groups. However, if it is assigned to the second group, three machines A, B, and C should be duplicated in the second group to achieve the same result.

To avoid the problem mentioned above, the number of operations of each exceptional part in different cells (NOP) should be used as a criterion for the assignment process. By using NOP, the cell which performs a larger portion of the operations of a part than any other cell is identified, then the part is assigned to this cell. The assignment process is carried out by the ASSGN routine.

Machines		Components									
		1	2	3	4	5	6	7	8	9	10
Cell I	A	1	1	1	1						
	B	1	1	1	1						
	C		1	1	1						
Cell II	D				1	1	1	1	1	1	1
	E					1	1	1	1	1	1

Figure 15. Machine Component Chart

Block XI represents the last function of the assignment unit. This includes the calculation of the number of exceptional parts visiting each bottleneck machine, NBTLK. NBTLK's are determined by ASSGN and used to determine the bottleneck machines. Since processing of any exceptional part on a machine involves an inter-cellular move, the information regarding the number of exceptional parts, visiting a machine is used to identify the bottleneck machines.

Bottleneck Unit

The major functions of this unit are represented by blocks XII, XIII, XIV, XV, XVII, and XVIII. The bottleneck unit, generally, deals with the problem of bottleneck machines and does the following:

1. Identifies the bottleneck machines and chooses the one which creates the largest number of inter-cellular moves - the critical machine.
2. Determines the cell whose parts have the largest number of operations on the critical machine - the critical cell.
3. Decides about the duplication of the critical machine in the critical cell.
4. Duplicates the bottleneck machine (if necessary) and modifies the machine-component chart accordingly.

This unit uses the output of the two previous units and determines the final form of machine cells for a given threshold value.

The first function of this routine is represented by block XII (Figure 13) and relates to the determination of bottleneck machines. Since NBTLK's for all machines are determined by ASSGN, a bottleneck machine can be easily identified. In fact, any machine with NBTLK > 0 is a bottleneck machine.

Block XIII in the bottleneck unit relates to the identification of the critical machines. A critical machine, among the bottleneck machines, is the one which creates the largest number of inter-cellular moves. For the purpose of reducing the inter-cellular moves, it is logical to choose such a machine as the best candidate for duplication. To identify a critical machine, NBTLK's for all bottleneck machines are compared and the machine with the largest NBTLK is selected as a critical one. This machine will be dropped from the list of bottleneck machines to avoid any further consideration of it. This process is done by the BOTLK routine.

The next function of this unit is to choose the cell whose components have the largest number of operations on the critical machine - the critical cell. This function is represented by block XIV in Figure 13 and relates to routine DUPLT in the computer program. To identify the critical cell, DUPLT does the following:

1. For each cell, it identifies the parts which have some operations on the critical machine.
2. It calculates the number of trips taking place between each cell and the cell containing the critical machine. The number of trips, NTRIP, for each cell is simply the summation of the number of moves related to parts, NPRTS, in the cell having an operation on the critical machine.

$$\mathrm{NTRIP}_j = \sum_{i=1}^{K_j} \mathrm{NPRTS}_i * M_i \quad \text{for } j = 1, 2, \ldots, \mathrm{NCELLS} \tag{3.3}$$

where,

NTRIP_j = the number of trips between cell j and the critical machine

K_j = the number of parts from cell j which have an operation on the critical machine

NPRTS_i = the number of part i to be produced.

M_i = the number of times part i moves between the cell and the critical machine

NCELLS = the number of cells.

3. NTRIP's of different cells are compared and the cell with the largest NTRIP is chosen as the critical cell.

Block XV of this unit deals with a limited treatment of machine requirements analysis and load balancing of the individual cells. This block relates to the NAVAL routine of the computer program. This routine receives the information related to parts in the critical cell (which have operations on the bottleneck machines) from routine DUPLT. Data items such as the processing time of each part on the bottleneck machine , the sequence of operations of parts on bottleneck machines,and the number of each part required are inputs to NAVAL.

The purpose of the machine-component grouping process is to form a set of mutually exclusive machine cells such that all parts in each cell can be entirely processed within that cell. If the nature of the production system is such that this purpose can be served, no load balancing for the individual cells will be necessary. This is true because of the assumption that the machine requirements of the manufacturing process has been already provided for, so the transfer from conventional manufacturing to cellular manufacturing can be done without any additional machine requirements. However, if some machine types are required by several machine cells, a new but limited machine requirements analysis is necessary. The analysis is limited because only a few cells require it and a limited number of bottleneck machines are involved.

The basic idea is to determine the number of bottleneck machines required by the critical cell to process all the exceptional parts within that cell. If only a fraction of a machine is required or several machines plus a fraction are required, then the problem of load balancing arises. In such a case it should be decided whether it is more economical to buy an additional machine and assign it to the critical cell, or to reroute the parts to another cell which already has that machine.

The machine requirement of the critical cell can be determined as follows (56):

$$N = \sum_{i=1}^{n} \frac{T_i P_i}{HC} \qquad (3.4)$$

where,

N = number of machines required

n = number of parts having an operation on the bottleneck machine

T_i = processing time of part i on the bottleneck machine

P_i = number of part i required

H = production hours per day

C = use factor of the bottleneck machine

The number of parts requiring the bottleneck machine, n, is determined by routine DUPLT. T, P, H, and C are input data. For the processing time of each part on the bottleneck machine, T predetermined standard data or past records can be used. H is usually a constant; it depends on the number of working shifts.

The total production, P is the sum of accepted (P_a) and rejected (P_r) parts. This can be presented by

$$P = P_a + P_r$$

Since the same amount of time is spent on rejected parts as on accepted ones, the rejected parts should be considered in the determination of machine requirements. Therefore, to calculate the required product, calculations start with the final operation. For this operation the required product is equal to the expected sales estimated by the sales department. For the next operation, which is the operation immediately before the final operation, the required product is equal to:

required product of the final operation +
expected rejected product of the final operation

In the same manner, product requirement for each operation is determined. For determination of the product requirement, past data are necessary. If the process is new an analysis of a similar process can be useful.

Another important factor in machine requirements analysis is the use factor. This factor is the ratio of the maximum expected machine availability to actual production hours. The use factor depends upon machine type, percentage of utilization, and the effectiveness of maintenance program. For a simple standard machine, a use factor as high as 0.95 is not unusual, while for a complex machine designed for specialized operations, this ratio can be as low as 0.60 or even less.

A complete analysis of machine requirements needs a lot of data and is beyond the scope of this work. However, a limited analysis of a few cells and a small number of bottleneck machines is very useful in arriving at a decision regarding duplication of bottleneck machines.

Block XVII of the bottleneck unit deals with the duplication process. Originally, there is only one machine of each type in the machine-component chart. This machine is assigned to the cell whose members are closely related to it. If parts in another cell(s) require the same type of machine, some additional machines of this type should be acquired and assigned to that cell(s). This process is referred to as the "duplication process." When a bottleneck machine is duplicated in a cell which requires it, the machine should be added to the list of machines in that cell. In addition, the machine-component chart should be modified to reflect this change. This modification is necessary because after the bottleneck machine is duplicated in the related cell, the parts which were previously moving out of the cell to be processed on the bottleneck machine will remain in the cell. That means these parts will not create any inter-cellular moves any longer; therefore, this change should be incorporated into the modified machine-component chart. As soon as the bottleneck machine is duplicated in a particular cell, the related cell number will be recorded to avoid any further consideration of that cell.

Block XVIII in Figure 13 deals with the cost-benefit analysis of the duplication process when an additional machine should be bought. This block relates to routines NAVAL and IVALT.

Duplication of bottleneck machines is done to eliminate the inter-cellular moves created by them. In an ideal case, it is possible to assign all bottleneck machines to all the cells requiring them, and to eliminate all inter-cellular moves without the need for buying any additional machines. When and if this is achieved, then the objective of the machine-component grouping process is completely met. However, in a real world situation, some additional machines may be required. One approach to this problem is to assume that the additional machines can be bought without considering the consequences of such a decision in terms of costs incurred and benefits produced. Even though the desired set of mutually exclusive machine-component groups can be formed in this way, the user will not have the slightest idea about the economical consequences of the decisions made.

Another method of dealing with the duplication process is to consider as many factors as possible and use an economic analysis of costs and benefits involved in the process. NAVAL and IVALT routines are designed to perform such an analysis. These routines determine the extent by which the inter-cellular moves are reduced when a bottleneck machine is duplicated. The extent of this reduction directly relates to the amount of part transfer between two cells due to lack of an additional bottleneck machine and the number of each part required to meet the production schedule, NPRTS. For each cell the parts visiting another cell have been already determined by routine ASSGN. These are the parts for which some operations must be performed on the bottleneck machine in another cell. On the other hand, NPRTS's are input data. Therefore, the number of trips saved by addition of a bottleneck machine to a particular cell can be determined by summing up the number of trips associated with each part. Thus, for a particular cell and bottleneck machine the following expression can be written:

$$NTRIPS = \sum_{k=1}^{n} NPRTS_k * M_k \qquad (3.5)$$

Were,

NTRIPS = the number of inter-cellular trips due to the bottleneck machine

n = the number of parts having operations on the bottleneck machine

$NPRTS_k$ = the number of part K to be processed on the bottleneck machine

M_k = The number of times part k moves between the cell and bottleneck machine.

The reduction in the volume of trips between two cells as a result of duplication can be converted into a reduction in material handling cost to be used for the decision making process. This can be done in two ways:

(1) Using a facilities design algorithm (such as CRAFT) to determine the material handling costs before and after duplication of the bottleneck machine.

(2) Using an average cost per trip and calculating the total cost of all the trips involved.

In the first method, two from-to charts are developed for the inter-cellular trips of the machine cells. One of these charts is developed for the case when the bottleneck machine is not duplicated, and the inter-cellular trips to it still exist. The second one is constructed after the duplication of the bottleneck machine when such trips have been eliminated. The difference between the material handling costs determined by the algorithm in the two cases is the cost reduction due to duplication of the bottleneck machine.

Since using a facilities design algorithm for each xduplication case is computationally cumbersome, the second method is more practical. The second method needs some kind of an average cost per trip. Such an average cost can be estimated by using past data or information gained in different phases of the machine-component grouping process. Such information is obtained when the material handling costs of inter-cellular and intra-cellular trips are calculated and used as a basis for selection of the threshold value. Of course the information is not available for the first iteration of machine-component grouping and a rough estimate should be used. But, in the subsequent iterations, more information regarding the inter-cellular trips is obtained and can be used to determine an average cost per trip.

NAVAL and IVALT also calculate the annual cost incurred due to the purchase of an additional bottleneck machine. To determine such a cost the major cost factors are considered. These include: purchasing price, installation costs, useful life, salvage value, and the required rate of return on investment. A capital recovery formula is used to calculate the annual cost (taxes have not been considered):

$$CR = (P-S) * (A/p, i\%, N) + S * i \qquad (3.6)$$

Where,

CR = annual cost of capital recovery and return

P = total installed cost of the new machine

S = salvage value of the new machine

A/p = capital recovery factor

N = useful life

i = rate of return on investment

The annual cost, AC, is determined as follows:

$$AC = CR + OC$$

where,

OC = additional operating and maintenance cost due to the purchase of the new machine

The savings due to reduction in the inter-cellular trips and cost incurred due to the purchase of a bottleneck machine can be used as a basis for the decision regarding the duplication of the bottleneck machine.

Threshold Value Unit

Blocks XXII, XXIII, XXIV, XXV, and XXVI represent the major functions of the fourth unit. This unit is, mainly, concerned with the selection of a proper threshold value. As discussed before, the clustering algorithms based on the similarity coefficient method give one solution for each threshold value. The threshold value is a similarity value which indicates the similarity level at which two machines or groups of machines should be joined together. Therefore, it is a measure which shows how similar the group's members are. It does not indicate how good the machine cells are for production purposes. If the threshold value is very large, few machines with high pairwise similarity coefficients are clustered in each cell, and there will be a large number of small cells. In this case, not many part-families can be entirely processed in a single cell, and a large number of inter-cellular moves are created. If a small threshold value is selected, many machine cells are merged and few cells of large size are formed. In the latter case fewer inter-cellular moves are created, however, due to the large size of cells, the number of intra-cellular moves are increased. The material handling costs of all moves associated with each threshold value should be used as a basis for choosing a proper threshold value. The purpose of this unit is to calculate these costs.

Block XXII in figure 14 relates to calculation of inter-cellular material handling costs. The material handling cost for cells of a given threshold value is calculated as follows:

$$MCA = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} d_{ij} NTRIP_{ij} \qquad \text{for } i \neq j \qquad (3.7)$$

where,

MCA = inter-cellular material handling cost

N = number of cells

C_{ij} = cost of one unit distance of handling a unit load between cells i and j

d_{ij} = distance between cells i and j

$NTRIP_{ij}$ = number of trips taking place between cells i and j.

$NTRIP_{ij}$ depends upon the number of part types moving between cells i and j, number of times each part type moves (sequence of the operations of the part on machines in the cell), and the production volume for each part:

$$NTRIP_{ij} = \sum_{k=1}^{n} NP_k * M_k \qquad (3.8)$$

where,

n = number of part types having operations in both cells i and j

NP_k = production volume for part k

M_k = number of times part k moves between cells i and j

The number of part types moving between cells i and j, n is determined by calculating the number of exceptional parts having operations in cells i and j. This is done by the INRTC routine of the computer program. The number of times a part moves between cells i and j (before all its operations are complete) is determined by the SEQNC routine of the computer program, the production volume, NP is user's supplied data.

To illustrate the procedure for determination of the inter-cellular trips, consider the machine-component chart in Figure 16.

		Components					
		1	2	3	4	5	6
	A	1	1	1	1		
	B	1	1	1			
Machines	C		1	1	1	1	
	D				1	1	1
	E				1	1	1

Figure 16. A Machine-Component Chart with Five Machhines

In this machine-component chart, machines A, B and C belong to Cell 1, while machines D and E belong to cell 2. As can be seen, parts 4 and 5 are exceptional parts. Suppose one unit of each part (per week) is processed in the cells. Further more, assume that part 4 is processed on machines A, D, C, and E for its first, second, third and fourth operations, respectively. Part 5 is processed on machines C, D, and E for its first, second and third operations, respectively. According to the seqence of operations, the number of times part 4 is transferred between cells 1 and 2 is two ($M_4 = 2$). Part 5 is first processed in cell 1 (on machine C) and then is transferred to cell two for its final operations ($M_5 = 1$). The number of trips between cells 1 and 2 can be calculated as follows:

$$NTRIP_{1,2} = 2(1) + 1(1) = 3 \text{ trips/week}$$

Based on the number of inter-cellular trips, a from-to chart for the cells formed at each threshold value is developed. The from-to chart shows the flow volume between cells for a given threshold value. Also, the area requirement of each cell is determined (48) and used to develop an initial layout in which the cells are placed arbiterarily. The from-to chart, initial layout, and a move-cost chart are used by the CRAFT algorithm which determines the inter-cellular material handling cost. Since the final results of the CRAFT alogorithm depend upon the arrangement of cells in the initial layout, several initial layouts should be tried to improve the near optimal solution given by CRAFT. This process should be repeated for all selected threshold values.

Block XXII in Figure 14 relates to determination of the intra-cellular trips. The number of trips taking place between the machines depends upon the number of part types moving between them, the number of times each part type moves between the two machines (the sequence of operations on two machines), and the production volume of each part. The number of trips between two machines is determined in the same way in which the number of trips between two cells is determined (eq. 3.8). The intra-cellular material handling cost for a specific cell is determined as follows:

$$MCW = \sum_{i=1}^{N} \sum_{j=1}^{N} C'_{ij} d'_{ij} NTRIP'_{ij} \quad \text{for } i = j \qquad (3.9)$$

where,

MCW = intra-cellular material handling cost for a specific cell

N = number of machines in the cell

C'_{ij} =cost of one unit of distance of handling a unit load between machines i and j

d'_{ij} = travelling distance between machines i and j

$NTRIP'_{ij}$ = number of trips taking place between machines i and j

Based on the number of intra-cellular trips for each cell, a from-to chart is constructed which shows the flow volume between machines in that cell. The area requierements of machines in the cell are determined (48), and based on that an initial layout is developed in which each machine is treated as a department. The from-to chart, initial layout, and a move-cost chart is used by the CRAFT algorithm which eventually determines the intra-cellular material handling cost. Due to the heuristic nature of CRAFT, several initial layouts should be tried to improve the final results. The material handling costs between machines of all cells associated with each threshold value are calculated. The sum of these costs is the intra-cellular material handling cost of a given threshold value.

Block XXVI (Figure 14) relates to routine BUFER and OTPUT in the computer program. These two routines organize, tabulate, and print the results of the computer program. The following items are prepared and printed out:

(1) The initial machine-component chart and other initial values.
(2) A dendogram.
(3) A list of cells and machines in each cell for each threshold value.
(4) A from-to chart for each threshold value.
(5) The number of intra-cellular moves for each threshold value.
(6) The intra-cellular material handling cost of each cell.
(7) A machine-componet chart in which machines are grouped

into cells and parts are assigned to them.

(8) A list of bottleneck machines.

(9) A machine-component chart in which the bottleneck machines are duplicated. This chart should be very close to a block diagonal in form.

In the next chapter the solution procedures employed by the model are discussed. The analysis of results is given in Chapter V.

CHAPTER IV

SOLUTION METHODOLOGY

The cost based machine-component grouping model is designed to carry out a variety of functions discussed in the previous chapter. Since the model is complex and its functions are diverse, no single procedure or technique can be employed to do these functions. In fact, different procedures are utilized for performing different functions of the model. For finding the machine-component groups, cluster analysis techniques are appropriate. Dealing with bottleneck machines requires special data analysis techniques. Finally, the selection of the threshold value is based on material handling costs which may be calculated based on facilities design procedures.

Due to the huge volume of data involved in the machine-component grouping process, data analysis is a major problem and without employment of effective data storage, retrieval, and analysis techniques, it would be impossible to perform the many jobs planned to be done by the model. For this reason, a brief description of the data analysis techniques used by the model is covered in this chapter.

Data Storage and Analysis

The machine-component chart used for the clustering purpose is an M x N matrix with zero/one elements. M and N are the number of machines and parts, respectively. For practical purposes, hundreds of machines and thousands of parts could exist in a machine-component chart. Using one computer word to store a single data item will engage a large amount of computer storage. In addition, the data analysis would be very difficult, if this type of data storage is employed. However, the fact that the data in the machine-component chart are of binary type (a part either has an operation on a specific machine or does not) permits the usage of an alternative data storage technique, i.e., bit-level data storage which is discussed below.

Bit-Level Data Storage

Usually one computer word is used to store a single data item. If there are N parts in a machine-component chart, N computer words are necessary to store all the information related to the processing of parts on a single machine. By using bit-level storage for binary data, each bit in a computer word can be used to store one data item. A computer word in FORTRAN consists of 32 bits for most IBM machines. As a result, 32 data items can be stored in a single computer word; so the computer storage and computational effort necessary for data analysis can be reduced substantially. With bit-level storage, the number of computer words required for N parts in the machine-component chart will reduce to |N/32| (the smallest integer greater than or equal to N/32) and for CDC computers, which have a larger number of bits per word, the reduction would be even greater. Generally, the reduction depends upon the number of bits, NBITS, which varies in different computers.

When bit-level storage is used, it is necessary to read, store, and print the data as binary numbers. This cannot be done in FORTRAN directly. However, this problem can be overcome in two ways. First, the sequence of zeros and ones can be stored in a computer word by expressing them in exponents of two. For example, the five-digit sequence 10010 can be produced and stored in word NUMBER as

$$\text{NUMBER} = 2^4 + 2 = 18 \qquad (4.1)$$

In general, to set the nth bit of a computer word (the most right bit is bit number zero) equal to one, the word should be set equal to 2^n. If several bits are required to be one, the appropriate exponents of two are added together as in equation 4.1.

Secondly, hexadecimal numbers can be used to generate the desired binary sequence. Hexadecimal numbers are recognized by FORTRAN (VS FORTRAN Level 77) and can be easily converted to binary numbers. The correspondence between hexadecimal and binary numbers is shown in Table I. A binary number can be easily converted into a hexadecimal by coding each four digits of the binary sequence to one hexadecimal digit. For example, binary sequences 10001001 and 11110011 are 89 and F3 in hexadecimal, respectively. On the other hand, any hexadecimal number can be converted into a binary number by converting each of its digits into four binary digits.

TABLE I

CORRESPONDENCE BETWEEN BINARY AND HEXIDECIMAL NUMBERS

Binary	Hexadecimal	Binary	Hexidecimal
0000	0	1001	9
0001	1	1010	A
0010	2	1011	B
0011	3	1100	C
0100	4	1101	D
0101	5	1110	E
0110	6	1111	F
0111	7		
1000	8		

In VS FORTRAN (Level 77) hexadecimal numbers are preceded by letter Z to be distinguished from decimals. A word can be set to a hexadecimal number by a DATA statement as follows:

```
DATA |MASK (31), MASK (0)| Z 80000000, Z 00000001      (4.2)
```

This DATA statement will produce masks 31 and 0.

MASK (31) = 1000 0000 0000 0000 0000 0000 0000 0000

MASK (0) = 0000 0000 0000 0000 0000 0000 0000 0001

By using DATA statements, different binary sequences can be generated and stored in masks. For each mask, only the ith bit is stored as a one, while the other bits are zeros. To reproduce a binary sequence, a set of masks and the operator OR are used. The logical operator OR obtains the logical sum of two words. That means, the ith bit in the resulting word would be one if the ith bit in one of the two words is one. To illustrate the point, suppose it is desirable to produce sequence 1100100100 in word IWORD. Since bits in positions 9, 8, 5, and 2 score one, masks 9, 8, 5, and 2 are to be used. The procedure is as follows

MASK (9) = 1 0 0 0 0 0 0 0 0 0

MASK (8) = 0 1 0 0 0 0 0 0 0 0

MASK (5) = 0 0 0 0 1 0 0 0 0 0

MASK (2) = 0 0 0 0 0 0 0 1 0 0

The masking operation and the related value of IWORD at each iteration are presented in Table II.

To read and store a binary number by this procedure, each digit in the binary sequence is read by the computer. Then, the positions of "ones" in the sequence are determined and associated masks and the OR logical operator are used to compress each of the NBITS data items in a single computer word.

In cases of sparse data sets, where only a few digits score one in the binary sequences, it is usually more efficient to use the sequence number of "ones" as input. For example, in sequence 1 0 0 0 0 0 0 0 0 1 positions 9 and 0 contain a one; therefore, sequence numbers 9 and 0 can be used as inputs. These two numbers provide all the information required to generate the sequence. In fact, this binary sequence may be reproduced by ORing masks 9 and 0 which are related to the sequence numbers 9 and 0, respectively.

TABLE II

MASKING OPERATION ON IWORD

Iteration	Masking Operation	IWORD
1	IWORD = 00000 00000	0 0 0 0 0 0 0 0 0 0
2	IWORD = IWORD.OR.MASK (9)	1 0 0 0 0 0 0 0 0 0
3	IWORD = IWORD.OR.MASK (8)	1 1 0 0 0 0 0 0 0 0
4	IWORD = IWORD.OR.MASK (5)	1 1 0 0 1 0 0 0 0 0
5	IWORD = IWORD.OR.MASK (2)	1 1 0 0 1 0 0 1 0 0

Data Storage and Retrieval

A major portion of the data in the machine-component grouping process is in matrix form. In most cases, a large part of the matrix employed for data storage remains unused. For example, in the similarity matrix only one-half of the matrix is used (Figure 4). Although, for processing purposes, it is simpler to store data in matrix form, it is usually more economical to use arrays for this purpose. In this work, a major portion of the data in the machine-component process, including the machine-component chart, is stored in arrays. Since the binary data are first compressed in words and then stored in array form, a special procedure can be employed for accessing and using the data.

Suppose there are NMCHN machines and NPART parts in the machine-component chart, and the number of bits per each computer word is NBITS. The number of words, NWORD, required to store all information relating to the processing of parts on each machine (one row in the machine-component chart) is:

$$\mathrm{NWORD} = \lceil \mathrm{NPART/NBITS} \rceil \tag{4.3}$$

The ceiling function ($\lceil\ \rceil$) shows that NWORD should be rounded off to the smallest integer equal to or greater than NPART/NBITS. NWORD and NBITS are two important parameters in finding the location of each data item in the related array. If the data in the machine-component chart is stored in array IWORD, then the first NBITS of data items of machine i are stored at location L_i which can be determined as follows:

$$L_i = (i-1) * \mathrm{NWORD} + 1 \tag{4.4}$$

The associated data is stored in IWORD (L_i). All the data related to machine i are stored in IWORD (L_i) to IWORD $(L_i + \text{NWORD})$.

Sometimes part j is given and one needs to know whether it has an operation on machine i or not. In this case, the following steps are necessary:

(1) The number of words required to store the data related to the (i-1) previous machines is determined:

$$L_{i-1} = (i-1) * \text{NWORD}$$

(2) The number of words required to store the data items of row i prior to column j (the first j-1 entries in row i) is determined.

$$\mathcal{L}_{j-1} = \lceil j/\text{NBITS} \rceil \qquad (4.5)$$

(3) $L_j = L_{j-1} + \mathcal{L}_{j-1}$ is the location (in IWORD) of the word which contains the information regarding the operation of part j on machine i.

(4) The specific bit which carries the information is determined as,

$$K = j - \text{NBITS} * (L_j - 1) \qquad (4.6)$$

where,

K is the position of the bit related to part j in IWORD (L_j).

To determine whether this bit is zero or one, the logical operator SHIFT should be used. This operator and a few other operators which have been widely used in the data analysis are discussed next.

Data Analysis

Due to the huge amounts of data involved in the machine-component grouping process, data analysis takes a complex form. To simplify the analysis, special procedures and techniques must be employed. Bit-level data storage reduces the storage requirement and provides the opportunity for utilization of logical operators which facilitate the data analysis process. The logical OR operator which is employed to generate a binary sequence is one of them. Two other operators have been widely used in the data analysis of the machine-component grouping process: the logical AND and SHIFT operator.

The logical AND operator is used to obtain the logical product of two words. The ith bit in the resulting word is set to one if the ith bits in both words score one. For example, consider the following binary numbers:

IWORD (1) = 1 1 0 1 1 0 1 1 1 1
IWORD (2) = 0 0 1 1 1 1 0 0 0 1

the result of the AND operation is as follows,

IWORD = IWORK (1) .AND. IWORD (2) = 0 0 0 1 1 0 0 0 0 1 (4.7)

Another logical operator is the SHIFT operator. This operator shifts a specific bit of a word several places to right or left depending upon the given argument for the shift operation. In VS FORTRAN the shift function is defined as,

$$I2 = ISHFT\ (IWORD,M) \tag{4.8}$$

where,

I2 = the value of IWORD after shift operation

M = the shift argument, if $M>0$, IWORD is shifted to left by M places; if $M<0$, IWORD is shifted to right by M places; if $M=0$, IWORD remains unchanged.

For example, suppose IWORD = 1 1 0 1 1 1 1 1 0 1, then the results of shift operations are as follows:

I1 = ISHFT (IWORD,2) = 0 1 1 1 1 1 0 1 0 0

I2 = ISHFT (IWORD,3) = 1 1 1 1 1 0 1 0 0 0

The most significant bit of a computer word is used to determine the sign of the number stored in that word. If this bit is zero, the number is positive, otherwise, it is negative. In the above example I1 is positive while I2 is negative. Therefore, the last bit can be used to obtain some information about the other bits of a word. For this reason, in this work, the first 31th bits of a computer word have been used to store data, the last bit has been reserved for checking the value of the word after each shift operation. As a result, the number of bits, NBITS in the computer program is 31 rather than 32.

Cluster Analysis Techniques

Cluster analysis refers to a variety of procedures used to group elements with some common characteristics. Most clustering techniques have been developed in the field of numerical taxonomy and have been used in this field as well as in many other areas ranging from psychology to manufacturing. Sneath and Sokal, in their book entitled Principle of Numerical Taxonomy have discussed many aspects of the subject in detail (59).

To cluster a set of entities, most clustering techniques use a measure of similarity (similarity coefficients) defined for each pair of the entities. When the entities are of binary type, one way to define the similarity measure is "the percent of match" for the values of the two variables. A "match" between two variables occurs when they have the same values. To calculate such a similarity coefficient, a 2 x 2 table may be used to show the different alternatives (see Figure 17) (2).

This simple arrangement results in a series of similarity coefficients, depending on how a 'match' is interpreted. If simple matching is considered, the coefficient is calculated as:

$$S_{ij} = (a+d)/(a+b+c+d) \qquad (4.9)$$

where

S_{ij} = the similarity coefficient between objects i and j,

a = the number of one matches,

d = the number of zero matches,

b and c = the number of occurrences in which one of variables i and j is zero.

In this case both zero and one matches are included in the numerator and denominator. One major drawback of this coefficient, in most cases, is that the inclusion of zero matches makes it artificially large.

		Variable j		
		1	0	
Variable i	1	a	b	a + b
	0	c	d	c + d
		a + c	b + d	n

Figure 17. 2 x 2 Table for Two Variables

Russel and Bio (2) have suggested to exclude the zero matches from the numerator. The related similarity coefficient is calculated as:

$$S_{ij} = a/(a+b+c+d) \qquad (4.10)$$

In the above formula, the zero matches are included in the denominator. Jaccard (2) has defined another similarity coefficient in which the zero matches are excluded from the numerator and denominator. This similarity coefficient is defined as:

$$S_{ij} = a/(a+b+c) \tag{4.11}$$

In all cases mentioned so far, equal weights were given to the matched and unmatched pairs; however, there are a series of similarity coefficients in which different weights are given to each of them. One such coefficient, in which weight of two is given to the matched pairs, is defined by Dice (2) and can be written as:

$$S_{ij} = 2a/(2a+b+c) \tag{4.12}$$

There are several other similarity coefficients of this type where the way different pairs are weighted varies. A complete discussion of the similarity coefficients is given by Michael A. Anderberg (2).

Cluster analysis techniques, generally, fall into two categories: hierarchic and non-hierarchic. Hierarchic clustering methods are those in which each cluster is a member of a larger cluster. The basic procedure for these clustering methods is to start with a high similarity level and group together all the elements with a similarity coefficient greater than this level. Then the similarity level is lowered step by step, and the existing groups are merged to form fewer groups of larger sizes. The process continues until all groups are embedded in a single group.

The requirement for all clustering algorithms is a similarity matrix which shows the strength of all pairwise associations among the objects to be clustered. The entries of the matrix are the measures of similarities defined before. If S_{ij} is the similarity coefficient between objects i and j, the similarity matrix can be constructed as in Figure 18. If the similarity coefficient is symmetric, i.e. $S_{ij} = S_{ji}$, then the matrix can be reduced to its lower triangle as in Figure 19. Therefore, the total number of similarity coefficients for n entries is $[n(n-1)]/2$. This similarity matrix is accessed by the clustering algorithm to group the entries. The basic steps for a hierarchic clustering method can be summarized as follows:

1. It starts with the similarity coefficient of 100 percent at which most entities form separate groups.
2. The similarity is lowered by a pre-determined increment, and all pairs of clusters having similarity coefficient of greater than this new similarity level are merged to form clusters of larger size.
3. The similarity matrix is revised and the similarity coefficient for each pair of the existing clusters is calculated.
4. Steps two and three are repeated until all clusters merge to a single one encompassing all the entities. At each similarity level the associated clusters are recorded in order to have a complete record of the results.

Single linkage, complete linkage, and average linkage clustering algorithms are three simplest and most popular clustering algorithms. A single linkage clustering algorithm was discussed in Chapter II. A brief discussion of the other two follows.

$$\begin{array}{lccccccc}
 & \text{Object } j & & & & & & \\
\text{Object } i & 1 & 2 & 3 & \cdot\;\cdot & n-1 & \cdot & n \\
1 & - & s_{12} & s_{13} & \cdot\;\cdot & s_{1,n-1} & & s_{1n} \\
2 & s_{21} & - & s_{23} & \cdot\;\cdot & s_{2,n-1} & & s_{2n} \\
3 & s_{31} & s_{32} & - & \cdot\;\cdot & \cdot\;\cdot\;\cdot\;\cdot & & s_{3n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
n & s_{n1} & s_{n2} & s_{n3} & \cdot\;\cdot & s_{n,n-1} & & -
\end{array}$$

Figure 18. General Similarity Matrix

$$\begin{array}{lccccc}
 & \text{Object } j & & & & \\
\text{Object } i & 1 & 2 & \cdot\;\cdot\;\cdot & n-1 & \cdot\;\cdot\;\cdot\;\cdot\; n \\
1 & & & & & \\
2 & s_{21} & & & & \\
3 & s_{31} & s_{32} & & & \\
\vdots & \vdots & \vdots & & & \\
n & s_{n1} & s_{n2} & \cdot\;\cdot & s_{n,n-1} &
\end{array}$$

Figure 19. Similarity Matrix With Symmetric Similarity Coefficients

The Complete linkage clustering method was devised to overcome the chaining problem of SLCA. This method is also called farthest neighbor clustering by Lance and Williams (60). In the complete clustering algorithm, the similarity coefficient between two clusters is defined as the similarity coefficient between the two farthest members in each cluster. This implies that the criterion for admission to a cluster is tough and clusters join each other with great difficulty and at a very low similarity level. As a result, very tight and discrete clusters will be formed.

In SLCA, due to the chaining problem, a single linkage causes two clusters with low similarity to join together. On the other hand, in the complete linkage method, many elements are excluded from joining together due to lack of high similarity coefficients with all members of the cluster. To overcome these two problems, a series of other clustering algorithms have been developed. One of these methods is the average linkage clustering algorithm which has been developed by Sokal (60). The similarity coefficient between a candidate and a cluster or two clusters, in this method, is determined based on the average similarity of all pairs involved. Since there are different methods for calculating the average similarity, different average linkage clustering methods have been developed.

One of the most popular average clustering methods is the unweighted pair-group method using an arithmetic average (UPGMA). This method was first developed by Sokal and

Michener (60). The UPGMA algorithm computes the similarity coefficient between a candidate and each member of a cluster and obtains the simple average of these similarities as a similarity measure for clustering. This implies that equal weights are given to each entity regardless of its position within the cluster.

Since the UPGMA algorithm has been used to form the machine cells in the present model, an example is used to illustrate this procedure. Figure 20 shows a machine-component chart. The pairwise similarity coefficients between machines are given in the similarity matrix in Figure 21. At the first iteration B and C join together and group BC is formed. Then, the similarity matrix is revised and the similarity coefficients between the new cell and the remaining machines are calculated and entered in the matrix. Using the average linkage method, the similarity coefficient between two groups i and j, S_{ij} is determined as,

$$S_{ij} = s_{ij}/(N_i \cdot N_j) \tag{4.13}$$

where,

s_{ij} = the sum of pairwise similarity coefficients between all members of the two groups,

N_i and N_j = the number of entities (machines) in groups i and j, respectively.

Therefore, the similarity coefficients between group BC and the remaining machines: A, D, and E are calculated as,

$$S_{BC,A} = \frac{S_{A,B} + S_{A,C}}{N_{BC} \cdot N_A} = \frac{0 + 0}{2(1)} = 0$$

$$S_{BC,D} = \frac{S_{D,B} + S_{D,C}}{N_{BC} \cdot N_D} = \frac{.50 + .33}{2(1)} \doteq .41$$

$$S_{BC,E} = \frac{S_{E,B} + S_{E,C}}{N_{BC} \cdot N_E} = \frac{0 + 0}{2(1)} = 0$$

The revised similarity matrix is shown in Figure 22. At the second iteration, the two most similar groups are A and E. These two join together and group AE is formed. The new similarity matrix is given in Figure 23. At the third iteration, groups D and BC join together and group BCD is formed. The revised similarity matrix is depicted in Figure 24. Finally, groups AE and BCD merge and group AEBCD is formed.

		Components									
		1	2	3	4	5	6	7	8	9	10
Machine	A	1	1				1	1	1		
	B			1	1	1				1	1
	C			1	1	1					1
	D			1	1		1			1	
	E	1	1					1	1		

Figure 20. Machine-Component Chart

		Machines				
		A	B	C	D	E
	A	-				
	B	0	-			
Machines	C	0	.80	-		
	D	.12	.50	.33	-	
	E	.80	0	0	0	-

Figure 21. Similarity Matrix for Data in Figure 16

		Groups			
		A	BC	D	E
	A	-			
	BC	0	-		
Groups	D	.12	.41	-	
	E	.80	0	0	-

Figure 22. Revised Similarity Matrix After First Iteration

		Groups		
		AE	BC	D
	AE	-		
Groups	BC	0	-	
	D	.06	.41	-

Figure 23. Revised Similarity Matrix After Second Iteration

		Groups	
		AE	BCD
Groups	AE	-	
	BCD	.03	-

Figure 24. Revised Similarity Matrix After Third Iteration

As can be seen in this example, the similarity coefficient between two groups is based on the similarity coefficients between all members of the two groups. For this reason, the chaining problem of SLCA does not exist when ALC is used.

Procedures for Machine-Component Grouping

The procedures for machine-component grouping can be categorized under three general classes:

(1) Procedures for forming machine cells

(2) Procedures for dealing with bottleneck machines

(3) Procedures for selection of the threshold value

The detailed discussion of all the procedures here is lengthy and not necessary because these procedures are dealt with in the program listings of Appendix C. However, the most important procedures and techniques employed in each class will be discussed below.

Procedures for Forming Machine Cells

The clustering techniques used to form machine cells have already been discussed. However, techniques used to calculate the similarity coefficients are further elaborated upon. Since we frequently need to refer to the information related to each machine in the machine-component chart, let us define a new term for this purpose: machine vector. Machine vector can be defined as an array containing the information related to the processing of parts on a specific machine. A machine vector, in fact, is a row in the machine-component chart.

Each machine vector contains NPART data items where NPART is the number of parts in the machine-component chart. The similarity coefficient between two machines i and j is determined to be

$$S_{ij} = \frac{\text{Number of components visiting both machines}}{\text{Number of components visiting either of the two machines}} \qquad (4.14)$$

To determine parts visiting one of the two machines, we have

$$MV = MV_i.OR.MV_j \qquad (4.15)$$

where,

MV = a machine vector containing the information regarding the operations of parts on machines i and j

MV_i and MV_j = machine vectors i and j.

Since the data items in each vector are compressed in NWORD computer words, the associated words should be OR'ed. Doing so, one needs to determine the location of the first word of each machine vector in array IWORD which has been used to store the data. If these locations are represented by L_i and L_j, they can be determined as follows:

$$L_i = (i-1) * \text{NWORD}$$
$$L_j = (j-1) * \text{NWORD} \tag{4.16}$$

Then equation 4.15 can be written as,

$$\text{MV} = \sum_{k=1}^{\text{NWORD}} \text{IWORD } (1_i+K).\text{OR}.\text{IWORD } (L_j+K) \tag{4.17}$$

It should be noted that the data in machine vector MV are compressed in NWORD words of temporary storage, ITEMP. Therefore, the last equation can be rewritten as

$$\sum_{k=1}^{\text{NWORD}} \text{ITEMP}(K) = \sum_{k=1}^{\text{NWORD}} \text{IWORD}(L_i+K).\text{OR}.\text{IWORD}(L_j+K) \tag{4.18}$$

As can be seen, any part which has an operation on either machine i or j, will have a score of one in the machine vector MV which is stored in ITEMP. To determine the number of parts visiting one of these two machines, it is enough to count the number of bits scoring one in machine vector MV. For this purpose, the number of these bits for each word of ITEMP is determined, the sum of these numbers indicates how many parts visit one of the two machines. The number of "ones" in each computer word is determined as follows:

(1) Set M = NBITS - J for the jth bit.

(2) Shift jth bit M locations to left.

(3) If the number in the computer word is negative, increment the number of "ones".

(4) Repeat steps 1, 2, and 3 for all bits in the computer word.

Suppose it is required to determine the number of "ones", CONTD in the machine vector MV. The data in this machine vector are stored in NWORD computer words each having NBITS of data items. This function can be programmed in VS FORTRAN as,

```
          L=0                                          (4.19)
          CONTD = 0
       DO 200 I=1, NWORD
          I1 = ITEMP(L+I)
       DO 100 J=1, NBITS
          I2 = ISHFT(I1,J)
          If (I2.LT.0) CONTD = CONTD + 1
   100 CONTINUE
   200 CONTINUE
```

The number of "ones", CONTD in the machine vector MV, indicates how many parts have operations on either of the two machines. The next step in calculation of the similarity coefficient is the determination of the number of parts visiting both machines. The procedure is exactly the same as the one used to determine CONTD, except, the logical operator OR is replaced by AND. If the number of parts visiting both machines is designated by CONTA, the similarity coefficient between machines i and j can be calculated as

$$S_{ij} = \text{CONTA/CONTD} \tag{4.20}$$

Based on the procedure discussed here, all pairwise similarity coefficients are calculated and stored in the similarity matrix.

The similarity matrix is the main input to the average linkage clustering algorithm which has been used to group similar machines into cells. The results of the clustering routine are used by routine TREE to form the dendogram as mentioned before. The clustering routine and routine TREE are based on the existing algorithms discussed by Anderberg (2). The dendogram is used by routine CELLS to form the machine cells.

Cell formation is the process of identifying the machine cells to be formed at each similarity level and developing the list of machines in these cells. The following factors are important in this process:

- For each cell, the list of machines should be easily available.
- For each machine, the parent cell (the cell to which the machine is assigned) should be identified.
- It should be easy to combine the cells and to form cells with larger sizes as the threshold value is lowered.

In machine-component formation, very often it is necessary to determine the machines belonging to a specific cell. Also, frequently it is required to determine the parent cell of a given machine. For these reasons, two arrays MCHIN and ICELL have been allocated to keep the records of machines and cells. The list of machines in

MCHIN is ordered such that the similar machines or groups of machines are close together. Another array, IFINH has been used to store the pointers to the last machine in each cell. The same pointers can be used to determine the first machine in each cell. This is true because the last machine in each cell (in MCHIN) immediately precedes the first machine in the next cell. To illustrate this, let us use the machines in the dendogram in Figure 5. For this dendogram with a threshold value of 1.00 the arrangement of the three arrays is shown in Figure 25.

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th></th></tr>
<tr><td>MCHIN</td><td>3</td><td>6</td><td>4</td><td>8</td><td>1</td><td>10</td><td>2</td><td>5</td><td>7</td><td>9</td><td>12</td><td>11</td><td>(a)</td></tr>
<tr><td>ICELL</td><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>4</td><td>5</td><td>6</td><td>7</td><td>7</td><td>8</td><td>8</td><td>(b)</td></tr>
<tr><td>IFINH</td><td colspan="2">2</td><td>3</td><td>4</td><td colspan="2">6</td><td>7</td><td>8</td><td colspan="2">10</td><td colspan="2">12</td><td>(c)</td></tr>
</table>

a = list of machines

b = list of cells

c = pointers to the last machine in each cell

Figure 25. Machine Cells at a Threshold Value of 1.00

At 1.00 similarity level, there are eight cells. The list of machines is stored in MCHIN; and ICELL keeps the record of the parent cell for each machine. Figure 25 shows that

$$\text{ICELL }(3) = 1$$
$$\text{ICELL }(6) = 1$$
$$\text{ICELL }(2) = 5$$
$$\text{ICELL }(11) = 8$$

IFINH keeps the pointer to the last machine in each cell. The pointer to the first machine in a cell is the pointer to the last machine of its previous cell plus one. Suppose IST and IFN designate the sequences of the first and the last machines in the seventh cell, respectively. These two variables can be determined as follows:

$$\text{IST} = \text{IFINH}(7-1) + 1 = \text{IFINH}(6) + 1 = 8 + 1 = 9$$
$$\text{IFN} = \text{IFINH}(7) = 10$$

The number of machines in this cell is

$$\text{NMCHN}_7 = 10 - 8 = 2$$

Of course, the pointer to the first machine in the first cell is always one (IFINH(0) = 1).

To show how the machine cells merge as the similarity level is lowered, suppose the threshold value is reduced to 0.50. At this level the arrangement of machines, cells, and pointers are as in Figure 26.

As can be seen, the merging of machine cells at each similarity level can easily take place by updating ICELL and IFINH. To prepare the list of machines in a specific cell, say, cell 3, it can be proceeded as follows:

```
IST = IFINH(3-1) + 1 = IFINH(2) + 1 = 7 + 1 = 8
                 IFN = IFINH(3) = 10
                 Do 100  I = IST,IFN
             100 MACIN = MACHIN(I)
```

where MACIN keeps the list of machines.

	1	2	3	4	5	6	7	8	9	10	11	12	
MCHIN	3	6	4	8	1	10	2	5	7	9	12	11	(a)
ICELL	1	1	1	1	2	2	2	3	3	3	4	4	(b)
IFINH	4				7			10			12		(c)

a = list of machines

b = list of cells

c = pointers to the last machine in each cell

Figure 26. Machine Cells at a Threshold Value of 0.50

The cell formation process is performed by routines ICELL, NEXTC, and IDCLL. The computer listings and detailed descriptions of the procedures for the cell formation process are given in Appendix C.

Procedures for Dealing With Bottleneck Machines

The identification of the bottleneck machines requires the following functions to be done:

(1) Machine cells must be formed and the list of machines in each cell be prepared.

(2) Operations of each part in different cells should be determined.

(3) Parts should be assigned to the associated machine cells.

(4) Exceptional parts should be identified.

(5) Number of exceptional parts visiting each machine must be determined.

The first function has already been discussed and is done by routine CELLS. To determine the operation done on a part in a specific machine cell, all machines in that cell must be checked to find out whether they perform any operations on that part or not. This is a tedious and time-consuming task. However, development and use of a new concept simplifies the process. A machine vector contains the information related to the processing of all parts on a specific machine. The same concept can be extended and applied to a cell, too. That means a cell vector can be defined as an array containing the information related to the processing of all parts in a specific cell.

To develop a cell vector, all the information in machine vectors belonging to the cell must be integrated into a single cell vector. This can be easily done by OR'ing all machine vectors of the cell. To illustrate the procedure, consider the machine-component chart in Figure

27. Suppose machines A, B, and C belong to cell I and machines D and E belong to cell II. Let the letters A to E be used to designate the associated machine vectors. The cell vectors for cells I and II (CVI, CVII) are developed as follows,

for cell I:

CVI = A.OR.B = 1 1 1 1 1 0 0 0 (4.21)

CVI = CVI.OR.C = 1 1 1 1 1 1 0 0

for cell II:

CVII = D.OR.E = 0 0 0 1 1 1 1 1 (4.22)

		Components							
		1	2	3	4	5	6	7	8
Machines	A	1	0	1	1	0	0	0	0
	B	0	1	1	1	1	0	0	0
	C	0	0	1	1	0	1	0	0
	D	0	0	0	1	1	1	1	1
	E	0	0	0	0	0	1	1	1

Figure 27. Machine-Component Chart

As can be seen, cell vectors CVI and CVII indicate which parts should be processed within cells I and II, respectively. Instead of dealing with five machine vectors, A to E, we can use these two vectors to obtain the information associated with the processing of any part in these cells. Since the data in the cell vectors are stored at bit-level, bit manipulation functions available in VS FORTRAN can be used to determine whether a part has any operation in a specific cell. These bit manipulation functions are based upon the logical operators OR, AND, and ISHFT which have been discussed before.

To determine whether a part has an operation in a specific cell, one can proceed as follows:

(1) The set of machine vectors belonging to the cell and the logical operator OR are used to construct the cell vector (equations 4.15, 4.21).

(2) The bit in the cell vector which contains the information related to the operation of the part in the cell is checked to find out whether it is zero or one (equation 4.8).

By using this procedure the cells visited by each part are determined and the exceptional parts are identified. In addition, the number of exceptional parts visiting each machine, NBTLK, and the number of machines in each cell visited by a part, NOP, are calculated.

It was mentioned in Chapter III that NBTLK's were used to identify the bottleneck machines and NOP's were employed to assign parts to appropriate cells. The detailed description of procedures used for these purposes are included in the computer listings in Appendix C. The procedure for the assignment of parts to machine cells is briefly described here.

The assignment of parts to different cells and bookkeeping for it are very similar to those used in dealing for machine cells (Figures 25 and 26). Here three arrays, NPRTC, ICELP, and JFINH have been allocated to keep the list of parts, the record of cells, and the pointers to the last part of each cell, respectively. The relationships among these arrays and the way they are updated are quite similar to MCHIN, ICELL, and IFINH. The identification of bottleneck machines and assignment of parts to cells are performed by routines BOTLK, ASSGN, and NOPRN. The computer listings and descriptions of these routines are given in Appendix C.

After the machine cells are formed and parts are assigned to them, the critical machines and cells are identified, duplication takes place, and the machine-component chart is modified. For each bottleneck machine which is duplicated in several cells, a list containing these cells is necessary. Keeping such a list is helpful in two ways:

(1) It identifies the machines being duplicated.

(2) It gives a list of relevant cells for each duplicated machine.

Array MCHND has been allocated to keep the records of cells for each bottleneck machine. In addition, a simple data packing technique has been employed to simplify the data storage and retrieval for this purpose.

In the machine-component grouping process there are usually a limited number of cells (below 100). A few of these cells may require a specific bottleneck machine. For this reason, a few two-digit numbers are required to designate the cells requiring a specific bottleneck machine. Since each computer word can store a large integer number, it is not necessary to use one computer word to store a single cell number. In fact, several cell numbers can be packed into a single word. To illustrate this, suppose there are less than 100 cells; and one of the bottleneck machine, say machine 5, should be duplicated in cells 9, 25, 30, 46, and 19. Since less than 100 cells are involved, two-digit numbers may be used for cell identification. If the largest integer that can be stored in the computer word is a 10-digit number, then five cell numbers can be stored per each computer word. The procedure is as follows:

The first cell number is stored in MCHND

$$\mathrm{MCHND}(5) = 9$$

For the second cell number we have

$$\mathrm{MCHND}(5) = \mathrm{MCHND}(5) + 25 * 10^2 = 2509$$

For the third, fourth, and fifth cells we can write

$$\mathrm{MCHND}(5) = \mathrm{MCHND}(5) + 30 * 10^4 = 302509$$

$$\mathrm{MCHND}(5) = \mathrm{MCHND}(5) + 46 * 10^6 = 46302509$$

$$\mathrm{MCHND}(5) = \mathrm{MCHND}(5) + 19 * 10^8 = 1946302509$$

As can be seen, a single word has been used to keep all the records. This job is done by routine IPACK of the computer program.

To retrieve the data for a bottleneck machine, the related word in MCHND should be divided by 100 in subsequent steps. For the first record, it can be proceeded as,

$$N = MCHND\,(5)/100 = 19463025$$

$$MCHIN = MCHND(5) - N * 100 = 1946302509 - 1946302500 = 9$$

For the second record

$$N1 = N/100 = 194630$$

$$MCHIN = N - N1 * 100 = 19463025 - 19463000 = 25$$

In the same way all other records can be retrieved. Routine INPAK of the computer program is used for this purpose.

Eight routines DUPLT, NAVAL, IVALT, DATA, MODIF, UPDAT, IPACK, and INPAK have been developed to carry out the machine duplication process. The computer listings and descriptions of these routines are given in Appendix C.

Procedures for Selection of the Threshold Value

After the cells are formed and parts are assigned to them, it is possible to determine which part requires intercellular moves in order to have all its operations completed. Since in the real world situations there are a large number of parts and machines, without a systematic approach it would be very difficult to carry out this job. To overcome this problem, the concept of a machine/cell vector and the application of logical operators are very crucial.

Since cell vectors contain the information related to the processing of all parts in a specific cell, they can be used as a basis for determining the inter-cellular moves. To clarify this, suppose it is desired to determine the parts which have operations in both cells i and j. To identify such parts, the logical product of cell vectors i and j must be used.

$$CV = CV_i \text{ .AND. } CV_j$$

where,

CV = cell vector containing the information related to parts having operation in both cells i and j

CV_i and CV_j = cell vectors of cells i and j.

The bits, which score one in CV, relate to parts creating inter-cellular moves. These bits can be identified (Equations 4.8, 4.19) and the related parts can be determined. To illustrate the procedure consider the machine-component chart in Figure 28.

Suppose machines A and D are assigned to cell I; machines B, C, and E are assigned to cell II. To determine the inter-cellular moves between cells I and II, it is proceeded as follows:

$$CVI = MVA.OR.MVD = 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

where,

CVI = cell vector for cell I

MVA, MVD = machine vectors for machines A and D.

For the second cell,

CVII = MVB.OR.MVC = 0 1 0 0 1 1 0 1

CVII = CVII.OR.MVE = 1 1 0 0 1 1 0 1

where,

CVII = cell vector for cell II

MVB, MVC, MVE = machine vectors for machines in cell II

Finally,

CV = CVI.AND.CVII = 1 0 0 0 1 1 0 0

		Components							
		1	2	3	4	5	6	7	8
Machines	A	1		1	1	1		1	
	D				1	1	1	1	
	B		1			1	1		
	C		1				1		1
	E	1	1						1

Figure 28. Machine-Component Chart

As can be seen, parts 1, 5, and 6 have operations in both cells and create inter-cellular moves. If the production requirement (the number of each part required) and the sequence of operations of parts are known, the number of trips, NTRIP, between the two cells can be determined (eq. 3.8). NTRIP'S show how many parts are transferred between two cells over a specific period of time. Based on NTRIP'S a from-to chart can be developed which shows all inter-cellular trips for the entire cellular manufacturing system. This from-to chart in conjunction with a move-cost chart and an initial layout can be used by the CRAFT algorithm to determine the material handling cost associated with a specific cells arrangement.

With a similar approach the intra-cellular trips and the associated material handling costs are determined. Machine vectors related to the machines in each cell are used to determine the intra-cellular moves for each cell. Then, these intra-cellular moves are used to determine the intra-cellular trips which are employed to develop from-to charts for different cells. By using the CRAFT algorithm the material handling cost of all the trips is calculated.

For each threshold value, the inter-cellular and intra-cellular material handling costs are determined. The sum of these two costs is the total material handling cost associated with each threshold value. The threshold value which generates the minimum material handling cost should be considered as a proper threshold value.

Routines INTRC, INTRA, INDS, and SEQNC are used to determine the materials handling costs associated with the different arrangements of machine cells. The computer listings and descriptions of these routines are given in Appendix C.

CHAPTER V

ANALYSIS OF RESULTS

The objective of this chapter is to further explain and evaluate the results derived from the cost based machine-component grouping model that was developed in this research. Results obtained from the model are presented, analyzed, and compared with those of a previous model. Since the model deals with two distinct problems: the selection of a threshold value; and the identification and duplication of bottleneck machines, the results are presented and discussed in two parts. In the first part, the results related to the selection of the threshold value are evaluated; in the second part the problem of bottleneck machines is presented and discussed.

To validate the model and illustrate using the model, a problem which has been already solved by Professor Burbidge (9, p. 172), using a manual solution procedure was chosen as the test problem. This is a machine-component grouping problem where 16 machines and 43 parts are involved. The initial machine-component chart of the test problem is depicted in Figure 29. The dendogram for this problem is prepared by the TREE routine of the computer program and is presented in Figure 30. The dendogram shows the machine cells and their associated machines for 10 different

Components

Machines	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1																																					1					1	
2		1								1																		1				1					1	1		1		1	
3							1										1																	1	1	1							
4					1				1					1					1		1		1						1														
5					1			1	1					1	1	1			1		1		1						1				1								1		1
6	1	1				1	1	1				1	1	1			1		1				1									1	1	1			1		1	1		1	1
7	1												1												1																		
8	1	1	1					1	1		1	1			1				1	1	1		1	1			1	1			1						1	1			1		1
9		1		1						1								1										1				1					1	1		1		1	
10	1											1	1												1	1					1								1				
11			1						1											1				1			1			1													
12											1											1		1			1			1													
13			1																					1																			
14		1				1											1																		1								
15					1									1					1		1												1								1		1
16		1					1			1								1														1					1	1				1	

Figure 29. The Original Machine-Component Chart

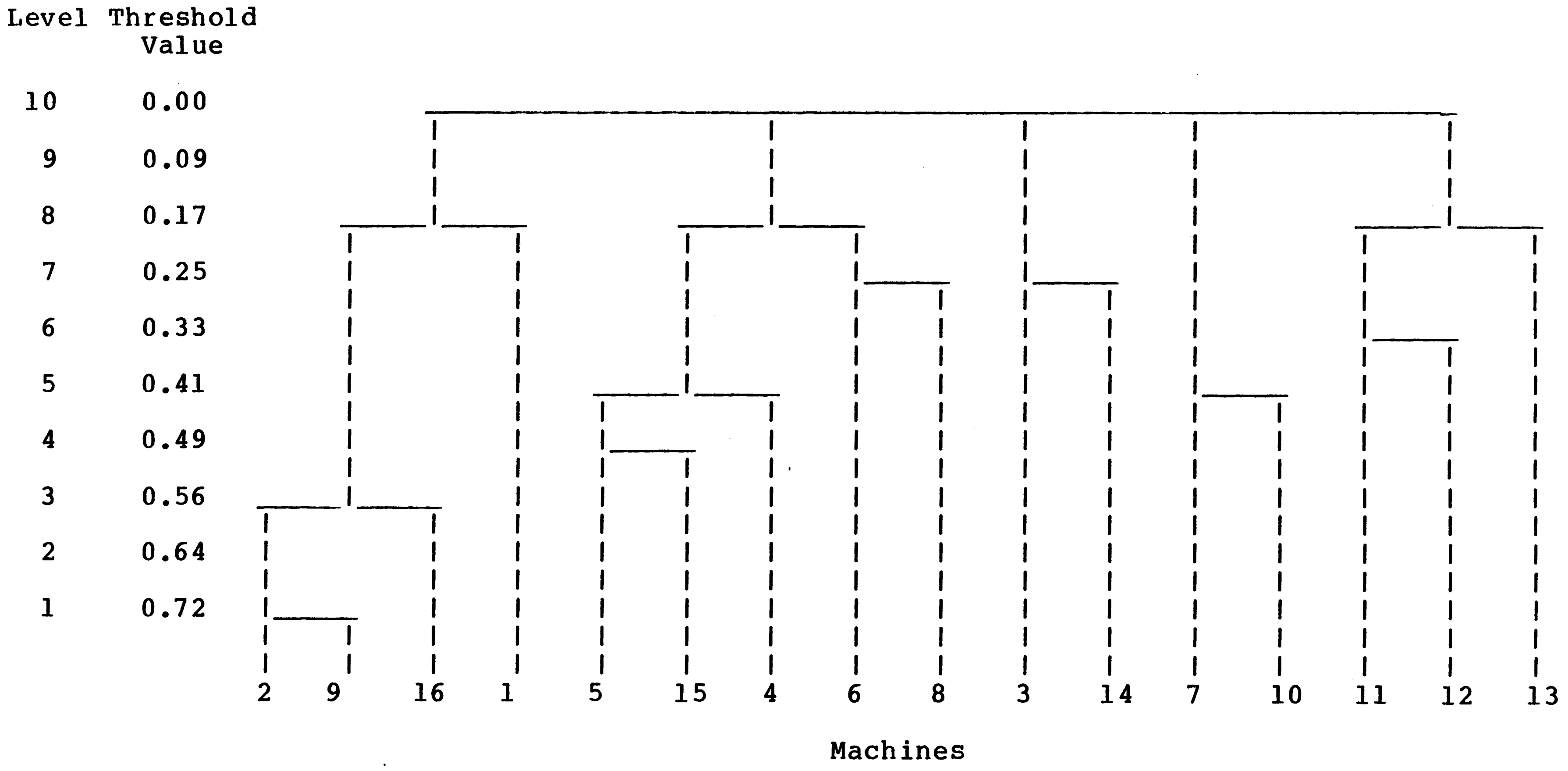


Figure 30. Dendogram For Test Problem

threshold values.

Selection of the Threshold Value

The machine-component grouping algorithms based on the similarity coefficient method give a set of solutions rather than a unique solution. Each solution relates to the threshold value used to form the groups. For this reason, the selection of a proper threshold value, which gives the best alternative solution based on some criteria, is a major problem. In this work, the material handling costs have been used as a basis for determination of a proper threshold value.

Lists of cells and the associated machines for 10 different threshold values are prepared by the CELLS routine. The results for a selected set of threshold values are given in Tables III through VIII.

The results show that for high threshold values very few machines join together and there are a large number of groups. For a threshold value of 0.72, only machines 2 and 9 are grouped together and there are 15 cells (Table III). Even when the threshold value decreases to 0.56, only three machines are grouped into the first cell, and the rest of them remain single (Table IV). For threshold values of 0.41, 0.33, 0.25, and 0.17, the number of cells is reduced to 11, 10, 8, and 5 respectively (Tables V through VIII).

As expected, at high threshold values, a large number of inter-cellular moves are generated. The number of inter-cellular moves is a function of the number of groups formed. At low threshold values, a smaller number of machine cells are formed, and less inter-cellular moves are generated. However, with a smaller number of cells, there will be a larger number of machines in each cell, so the transportation volume within each cell increases.

TABLE III

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.72

Cell Number	Machines in Each Cell
1	2, 9
2	16
3	1
4	5
5	15
6	4
7	6
8	8
9	3
10	14
11	7
12	10
13	11
14	12
15	13

TABLE IV

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.56

Cell Number	Machines in Each Cell
1	2, 9, 16
2	1
3	5
4	15
5	4
6	6
7	8
8	3
9	14
10	7
11	10
12	11
13	12
14	13

TABLE V

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.41

Cell Number	Machines in Each Cell
1	2, 9, 16
2	1
3	5, 15, 4
4	6
5	8
6	3
7	14
8	7, 10
9	11
10	12
11	13

TABLE VI

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.33

Cell Number	Machines in Each Cell
1	2, 9, 16
2	1
3	5, 15, 4
4	6
5	8
6	3
7	14
8	7, 10
9	11,12
10	13

TABLE VII

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.25

Cell Number	Machines in Each Cell
1	2, 9, 16
2	1
3	5, 15, 4
4	6, 8
5	3, 14
6	7, 10
7	11, 12
8	13

TABLE VIII

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.17

Cell Number	Machines in Each Cell
1	2, 9, 16, 1
2	5, 15, 4, 6, 8
3	3, 14
4	7, 10
5	11, 12, 13

Based on the inter-cellular moves between different cells, a from-to chart is developed. To do so, the production data for each part are used to calculate the number of inter-cellular trips created by each part (3.8). These trips are the entries of a from-to chart which is used to determine the material handling cost associated with each threshold value. The from-to chart showing the inter-cellular trips between machine cells for a specific threshold value is prepared by the INTRC routine. The from-to charts for four different threshold values are given in Figures 31, 32, 33, and 34.

The results in Figures 31, 32, 33 and 34 show that from-to charts associated with higher threshold values contain more inter-cellular trips. However, there are fewer intra-cellular trips associated with high threshold values. For the four threshold values given here, the associated intra-cellular trips are 42, 45, 55, and 93, respectively. It is not possible to determine which threshold value is better, except by converting these inter-cellular and intra-cellular moves into material handling costs.

The CRAFT algorithm is one of the facilities design algorithms which can be used to determine the inter-cellular material handling cost. The inputs to this algorithm are: (1) a control card, (2) a from-to chart, (3) a move-cost chart, (4) an initial layout (62).

The first item in the input list is a control card which describes the parameters of the problem and chooses the different options available in CRAFT. The from-to chart is prepared by the INTRC routine in the computer program. The move-cost chart contains costs per unit distance of transporting a unit load among cells and is input to the CRAFT algorithm. Finally, the initial layout is developed based upon the arrangement of the machine cells for a given threshold value.

		Machine Cells										
		1	2	3	4	5	6	7	8	9	10	11
Machine Cells	1	-										
	2	2	-									
	3	0	0	-								
	4	6	2	6	-							
	5	4	1	8	8	-						
	6	1	0	0	3	0	-					
	7	1	0	0	3	1	2	-				
	8	0	0	0	4	3	0	0	-			
	9	0	0	1	0	5	0	0	0	-		
	10	0	0	0	0	3	0	0	0	3	-	
	11	0	0	0	0	2	0	0	0	2	1	-

Figure 31. From-to Chart for Threshold Value = 0.41

		Machine Cells									
		1	2	3	4	5	6	7	8	9	10
	1	-									
	2	2	-								
	3	0	0	-							
	4	6	2	6	-						
Machine Cells	5	4	1	8	8	-					
	6	1	0	0	3	0	-				
	7	1	0	0	3	1	2	-			
	8	0	0	0	4	3	0	0	-		
	9	0	0	1	0	6	0	0	0	-	
	10	0	0	0	0	2	0	0	0	2	-

Figure 32. From-to Chart for Threshold Value = 0.33

		Machine Cells							
		1	2	3	4	5	6	7	8
	1	-							
	2	2	-						
	3	0	0	-					
Machine Cells	4	8	2	10	-				
	5	2	0	0	5	-			
	6	0	0	0	5	0	-		
	7	0	0	1	6	0	0	-	
	8	0	0	0	2	0	0	2	-

Figure 33. From-to Chart for Threshold Value = 0.25

		Machine Cells				
		1	2	3	4	5
	1	-				
	2	8	-			
Machine Cells	3	2	5	-		
	4	0	5	0	-	
	5	0	6	0	0	-

Figure 34. From-to Chart for Threshold Value = 0.17

The intra-cellular material handling cost, for each threshold value, can also be calculated by using the CRAFT algorithm. The summation of the inter-cellular and intra-cellular costs is the total material handling cost associated with a specific threshold value. Such a total cost should be calculated for all threshold values, and the threshold value associated with the minimum total cost should be considered more favorably.

In practice, it is not necessary to determine the total material handling costs of all the threshold values involved. Many alternative solutions are infeasible for practical purposes, so they can easily be discarded. The machine cells associated with the extreme threshold values are examples of these types of solutions. For example, in the test problem, at the threshold value of 0.72, only two machines join together and the related group layout is practically the same as a functional layout. The same is true for a very low threshold value which causes all machines be clustered into a single cell.

For the purpose of illustrating the selection procedure, the four threshold values of 0.41, 0.33, 0.25, and 0.17 have been chosen. Then, from-to charts related to these threshold values (Figures 31, 32, 33, 34) are used to determine the material handling costs associated with the machine cells formed at each similarity level. For simplicity, one monetary unit has been used as an entry in the move-cost chart used by the CRAFT algorithm. In the initial layout, one unit square is allocated to each machine. The inter-cellular material handling costs calculated by CRAFT are presented in Table IX.

As mentioned before, the CRAFT algorithm can also be used to determine the intra-cellular material handling cost. Doing so, a from-to chart based on the intra-cellular trips for each cell is constructed and an initial layout, and a move-cost chart are used by CRAFT to determine the intra-cellular material handling cost for each cell. In this illustrative case, it is assumed that moving a unit load within a cell is 40% cheaper than moving it among the cells. The intra-cellular material handling costs for the threshold values of 0.41,0.33,0.25, and 0.17 are given in Table 1X.

TABLE IX

INTER-CELLULAR AND INTRA-CELLULAR MATERIAL HANDLING COSTS

No.	Threshold Value	Inter-cellular Material Handling Cost	Intra-cellular Material Handling Cost	Total Cost
1	0.41	128.0	31.2	159.2
2	0.33	119.0	32.0	151.0
3	0.25	89.5	39.0	128.5
4	0.17	49.0	75.0	124.0

The results in Table IX indicate that for the given inputs, the material handling costs are at their minimum with the threshold value of 0.17. Therefore, all other factors equal the machine cells associated with this threshold value should be selected and used to establish a cellular manufacturing system.

Identification and Duplication of Bottleneck Machines

The selection of the threshold value helps us to choose a specific arrangement of machine cells among several alternative solutions derived from the dendogram. At the threshold value of 0.17, five cells of different sizes are formed. The cell numbers and lists of machines associated with these cells are given in Table X.

TABLE X

CELLS AND ASSOCIATED MACHINES FOR THRESHOLD VALUE = 0.17

Cell Number	Machines in Each Cell
1	2, 9, 16, 1
2	5, 15, 4, 6, 8
3	3, 14
4	7, 10
5	11, 12, 13

After the machine cells associated with the selected threshold value are determined, the parts are assigned to the related cells. Then, the machines and parts in the machine-component chart are rearranged such that the machine-component groups can be realized through visual inspection of the chart. These tasks are carried out by the ASSGN routine of the computer program; the results are presented in Figure 35. The dendogram merely shows the machines in each cell for a given threshold value. But the machine-component chart prepared by the ASSGN routine provides the user with the opportunity to visualize the machines and parts of each cell in the same chart (Figure 35).

The machines and parts in this machine-component chart have not been divided into mutually exclusive machine-

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	1	5	6	8	9	11	12	14	15	16	19	20	21	23	29	31	33	34	39	41	43	17	35	36	13	25	26	32	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
5													1		1	1			1	1	1	1		1	1	1		1			1	1											
15													1						1			1		1				1			1	1											
4													1			1			1			1		1	1	1																	
6	1		1				1	1		1	1	1		1	1			1	1			1			1			1	1	1		1	1			1							
8	1					1		1	1			1			1	1	1	1		1		1	1	1	1		1				1	1							1		1	1	
3			1																										1				1	1	1								
14	1													1																			1	1									
7												1																								1	1						
10												1						1									1			1						1	1	1					
11																1							1																1		1	1	1
12																	1																							1	1	1	1
13																																							1		1		

Figure 35. Machine-Component Chart Before Considering Bottleneck Machines, Parts are Assigned to Cells, "One" Entries Outside the Blocks Represent Intercellular Moves

component groups (Figure 35). This is due to a large number of inter-cellular moves created by the bottleneck machines. To obtain a block diagonal form, which shows the mutually exclusive machine cells, the bottleneck machines should be identified, so that they can be duplicated in the related cells. This is done by the BOTLK routine. This routine prepares the list of bottleneck machines and their associated inter-cellular moves. The list for the test problem is given in Table XI.

TABLE XI

BOTTLENECK MACHINES AND RELATED INTER-CELLULAR MOVES

No.	Bottleneck Machines	Number of Inter-Cellular Moves
1	6	8
2	8	7
3	10	4
4	11	2
5	14	2
6	3	2
7	7	1
8	12	1

In the next step, the DUPLT routine carries out the duplication process. Two cases are considered. First, it is assumed that one can duplicate as many machines as required to eliminate all the inter-cellular moves and to form a set of mutually exclusive machine-component groups. No economic factors are considered here. Secondly, the duplication process can be based upon an economic analysis of the major cost factors involved.

Duplication Without Considering Cost Factors

When the duplication process is carried out without considering the cost factors, all inter-cellular moves can be eliminated and a complete block diagonal form can be obtained. This is true because it is possible to duplicate any machine creating inter-cellular moves. However, it is obvious that such a practice will not be allowed in real world situations. Therefore, it is more logical to impose some restrictions on the machines which are candidates for duplication. For this purpose, the model imposes a lower limit, LIMIT, on the number of inter-cellular moves created by a bottleneck machine; only those machines creating a larger number of inter-cellular moves than LIMIT are duplicated by the DUPLT routine. The results of the duplication process for three different values of LIMIT are presented in Figures 36, 37, and 38. These results can be compared with the Burbidge's manual solution which is given

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	8	9	14	15	16	19	21	23	29	33	41	43	6	17	34	35	36	1	12	13	25	26	31	39	3	11	20	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
5	1		1				1	1		1	1																																
8	1					1		1	1																																		
3			1																																								
14	1																																										
5												1	1	1	1	1	1	1	1	1	1	1	1	1																			
15												1			1			1	1			1	1	1																			
4												1		1	1			1	1	1	1																						
6													1		1			1		1		1		1																			
8													1	1		1		1	1	1			1	1																			
11														1																													
3																										1	1	1	1														
14																									1	1		1															
6																									1	1	1																
7																														1		1	1										
10																														1	1	1	1	1	1	1							
6																														1	1	1				1							
8																														1	1				1								
11																																					1		1		1	1	1
12																																						1		1	1	1	1
13																																					1				1		
8																																					1	1	1		1	1	1

Figure 36. Final Machine-Component Chart (LIMIT = 0)

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	8	9	14	15	16	19	21	23	29	33	41	43	6	17	34	35	36	1	12	13	25	26	31	39	3	11	20	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1			1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
6	1		1				1	1		1	1																																
8	1					1		1	1																																		
5												1	1	1	1	1	1	1	1	1	1	1	1	1																			
15												1			1			1	1			1	1	1																			
4												1		1	1			1	1	1	1																						
6													1		1			1		1		1		1																			
8													1	1		1		1	1	1			1	1																			
3			1																							1	1	1	1														
14	1																								1	1		1															
6																									1	1	1																
7																														1		1	1										
10																														1	1	1	1	1	1	1							
6																														1	1	1				1							
8																														1	1				1								
11														1																							1		1		1	1	1
12																																						1		1	1	1	1
13																																					1				1		
8																																					1	1	1		1	1	

Figure 37. Final Machine-Component Chart (LIMIT = 2)

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	1	5	6	8	9	11	12	14	15	16	19	20	21	23	29	31	33	34	39	41	43	17	35	36	13	25	26	32	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
5													1		1	1			1	1	1	1		1	1	1		1			1	1											
15													1						1			1		1				1			1	1											
4													1			1			1			1		1	1	1																	
6	1		1				1	1		1	1	1		1	1			1	1			1			1			1	1	1		1	1			1							
8	1					1		1	1			1			1	1	1	1		1		1	1	1	1		1				1	1							1		1	1	
3			1																										1				1	1	1								
14	1													1																			1	1									
7												1																								1	1						
10												1						1									1			1						1	1	1					
11																1							1																1		1	1	1
12																	1																							1	1	1	1
13																																							1		1		

Figure 38. Final Machine-Component Chart (LIMIT = 10)

in Figure 39.

The duplication process begins with the machine creating the largest number of inter-cellular moves. In the subsequent steps, machines creating a smaller number of inter-cellular moves are considered. The process continues until one of the two following conditions are met:

(1) None of the remaining bottleneck machines create a larger number of inter-cellular moves than LIMIT.

(2) All inter-cellular moves are eliminated.

As mentioned before, it is not desirable to duplicate all the bottleneck machines. Therefore, the duplication process is terminated as soon as condition one is met. Sometimes, before condition one is satisfied, all the inter-cellular moves are eliminated and the duplication process is ended. This happens because the duplication of a bottleneck machine causes some parts be reassigned. The reassignment of parts, in turn, eliminates some of the inter-cellular moves. For example, the duplication of machines 6 and 8 in cell 4 (Figure 37) causes part 12 be removed from cell 2 and be reassigned to cell 4. This part previously had two operations in cell 2 and one operation in cell 4. After the duplication, this part can be process in cell 4 completely. The reassignment of this part eliminates the need for transporting it between cells 2 and 4.

The results in Figures 36, 37, and 38 indicate that the value of LIMIT has a great impact on the final form of the machine-component chart. When this value is zero, all

Components

Machines	1	13	39	25	12	31	26	42	37	2	32	36	10	43	28	18	4	27	24	3	20	30	11	22	17	7	35	6	34	36	19	23	14	43	5	9	21	41	10	29	8	33	16
10	1	1	1	1	1	1	1																																				
6	1	1	1		1																																						
7	1	1		1																																							
8	1				1	1																																					
9								1	1	1	1	1	1	1	1	1	1																										
2								1	1	1	1	1	1	1	1																												
16								1	1	1	1	1	1			1										1																	
6								1	1	1	1			1																													
8									1	1		1			1																												
1								1	1																																		
11																		1	1	1	1	1														1							
12																		1	1			1	1	1																			
8																		1	1	1	1		1																				
13																		1		1																							
3																									1	1	1		1	1													
6																									1	1		1	1														
14										1															1		1	1															
5																															1	1	1	1	1	1	1	1	1	1	1	1	1
4																															1	1	1		1	1	1			1			
6																															1	1	1	1							1	1	
15																															1		1	1	1		1	1				1	
8																															1	1		1		1	1	1	1		1		

Figure 39. Burbidge's Manual Solution

required bottleneck machines are duplicated, so a complete block diagonal form is obtained (Figure 36). If the lower limit is two (LIMIT=2), all the machines creating a larger number of inter-cellular moves than two are candidates for duplication. However, only machines 6 and 8 are duplicated. The reason is that these two machines are creating inter-cellular moves more than any other candidates (Table XI), so they are duplicated first. As soon as these machines are duplicated, some parts are reassigned. The reassignment of these parts results in the elimination of some inter-cellular moves. As a result, none of the remaining candidates create more than two inter-cellular moves. There are three inter-cellular moves in the final machine-component chart associated with LIMIT=2 (Figure 37). This machine-component chart is very close to the one constructed by Burbidge through manual solution (Figure 39).

When the lower limit is as high as 10, no machine is a candidate for the duplication, so the final machine-component chart is the same as in Figure 35. This is due to the fact that no bottleneck machine produces this many inter-cellular moves. The maximum number of inter-cellular moves is created by machine 6 and is equal to eight (Table XI).

Even though the selection of a lower limit such as LIMIT builds more flexibility into the model and provides the opportunity for evaluation of alternative solutions, it is still far from being a realistic approach. For this reason, an economic analysis of the major cost factors affecting the duplication process has been used as a basis for the decision regarding the duplication of bottleneck machines.

Duplication Based on Cost Factors

Two major cost factors have been considered in the economic analysis of the duplication process: (a) the material handling cost due to inter-cellular moves caused by a bottleneck machine, and (b) the costs incurred by the purchase of a new machine. For calculation of the material handling cost the following data are necessary:

(1) The exceptional parts having operations on the bottleneck machine.

(2) The number of each exceptional part to be processed on the bottleneck machine.

(3) The average cost per trip.

(4) The sequence in which each part is processed.

The exceptional parts having operation on the bottleneck machines are identified by the ASSGN routine ofthe computer program. The other two items are user's supplied data. To illustrate the procedure, the process of machines 6 and 8 in the related cells are explained. A list of exceptional parts which should be processed in these machines is given in Table XII. The parent cells, processing times, and production requirements of these parts are given in the same table.

TABLE XII

EXCEPTIONAL PARTS ON MACHINES 6 AND 8

No.	Part Number	Parent Cell	Bottleneck Machine(s)	Processing Time (hours)	Production Requirements (per week)
1	2	1	6, 8	0.10, 0.10	50
2	7	1	6	0.10	60
3	28	1	8	0.20	40
4	32	1	6	0.20	50
5	37	1	6, 8	0.10, 0.30	40
6	38	1	8	0.40	70
7	40	1	6	0.20	60
8	42	1	6	0.10	70
9	17	3	6	0.30	20
10	13	4	6	0.20	120
11	1	4	8	0.20	60
12	3	5	8	0.30	70
13	24	5	8	0.10	80
14	27	5	8	0.20	90

The data in columns 2, 3, and 4 of Table XII are prepared by routines BOTLK and ASSGN. The processing times and production requirements are inputs to the model. The additional necessary data for machines 6 and 8 are given in Table XIII. All the data items in this table are the user's supplied data. The cost data are based upon one monetary unit.

in addition to the data in Table XIII, the sequence of operations of parts, the average cost of transporting a unit load one unit of distance, ACOST, and the production hours per day, H, should be known. The ACOST for this illustrative example is assumed to be one monetary unit and the production hours per day equals to eight hours. The sequence of operations is such that all parts are processed within one cell and then are transferred to the next cell. The NAVAL routine uses the data in Table XII, use factors, and production hours per day to determine whether any additional machines of types 6 and 8 are required; if so, the IVALT routine uses the data in Table XIII and the ACOST to evaluate the economical feasibility of acquiring an additional machine. The evaluation is based upon two major factors: (a) the material handling cost of inter-cellular moves due to machines 6 and 8 and (b) the additional cost incurred by purchasing a new machine.

TABLE XIII

COST DATA FOR MACHINES 6 AND 8

No.	Item	Machine 6	Machine 8
1	Price	14,000	12,000
2	Installation Cost	300	200
3	Salvage Value	2,000	2,000
4	Useful Life	5 yrs	6 yrs
5	Required Rate of Return	10%	10%
6	Additional Operating Cost	300	250
7	Use Factor	0.90	0.90

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	6	8	9	14	15	16	19	21	23	29	33	34	41	43	17	35	36	1	12	13	25	26	31	39	3	11	20	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1			1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
8	1					1		1	1																																		
5												1		1	1	1	1	1	1	1	1	1	1		1	1																	
15												1				1			1	1			1		1	1																	
4												1			1	1			1	1	1	1																					
6	1		1				1	1		1	1		1	1		1			1		1		1	1		1	1																
8														1	1		1		1	1	1				1	1																	
3			1																					1			1	1	1														
14	1												1														1	1															
7																														1		1	1										
10																														1	1	1	1	1	1	1							
6																														1	1	1				1							
8																														1	1				1								
11															1																						1		1		1	1	1
12																																						1		1	1	1	1
13																																					1				1		
8																																					1	1	1		1	1	

Figure 40. Final Machine-Component Chart Considering Economic Factors

Machines 6 and 8 have already been assigned to cell 2. The cells from which the exceptional parts should be transported to cell 2 are identified by the DUPLT routine. The exceptional parts having operations on machines 6 and 8 and their associated cells (parent cells) are presented in Table XIV.

TABLE XIV

EXCEPTIONAL PARTS AND THEIR PARENT CELLS

No.	Bottleneck Machine	Exceptional Parts	Parent Cell
1	6	2, 7, 32, 37, 40, 42	1
2	6	17	3
3	6	13	4
4	8	2, 28, 37, 38	1
5	8	1	4
6	8	3, 24, 27	5

The NAVAL routine determines how many bottleneck machines should be duplicated in different cells in order to eliminate the associated inter-cellular moves (Equation 3.4). The results for the test problem are summarized in Table XV.

The exceptional parts 2, 7, 32, 37, 40, and 42 must be transferred from cell 1 to cell 2 to be processed on machine 6 (Table XIV). With the given production requirements and processing times for these parts (Table XII), the number of machines type 6 required to process all these parts within cell 2 is 1.22 (Table XV). Since it is assumed that the machine requirements of the production schedule have already been met, it is possible to assign one of the existing machines of type 6 to cell 1. However, for the fraction part (0.22), the case is different. This fraction of machine is not required to meet the requirements of the production schedule, but it is required to make cell 1 independent of cell 2. For this reason, it should be decided whether it is more economical to buy a new machine for cell 1, or to transport the related parts to cell 2 for processing. The IVALT routine is used to do this economic analysis.

TABLE XV

BOTTLENECK MACHINE REQUIRED BY DIFFERENT CELLS

No.	Machine	Cell	Number of Machines Required
1	6	1	1.22
2	6	3	0.17
3	6	4	0.67
4	8	1	1.78
5	8	4	0.33
6	8	5	0.30

If the fraction is large enough to justify the assignment of one machine to the related cell, the duplication is done without any economic analysis. In the illustrative case presented here, for any fraction greater than 0.50 one machine is assigned to the relevant cell. The data in Table XV indicate that the fraction of machine 6 required by cell 4 is 0.67, and the fraction of machine 8 required by cell 1 is 0.78. In each of these two cases one additional machine is assigned to the related cell. For the remaining cells, the fraction is less than 0.50 and the duplication should be based on the economic analysis done by IVALT. The results of such an analysis are summarized in Table XVI.

TABLE XVI

DUPLICATION BASED ON COST FACTORS

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	60.00	72.01	No
2	6	3	20.00	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	-----	-----	Not Required
5	8	4	60.00	53.69	Yes
6	8	5	57.36	53.69	Yes

The resulting machine-component chart after all feasible duplications have been done is depicted in Figure 40. For the given data, there are more inter-cellular moves associated with this solution than with Burbidge's solution (Figure 39). This is true because in this case it is not economically feasible to duplicate as many machines as have been duplicated in Burbidge's solution. To compare the two solutions, the costs due to the purchase of new machines and inter-cellular material handling costs due to these machines for the two cases are determined. The sum of these costs for a given solution can be calculated as follows:

$$TC = MC + AC/NW$$

where,

TC = total cost per week

MC = weekly inter-cellular material handling costs due to bottleneck machines

AC = annual cost due to purchase of new machines

NW = number of working weeks per year

The inter-cellular material handling cost (for N cells) can be determined as follows:

$$MC = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} d_{ij} NTRIP_{ij} \quad i=j$$

where,

MC = inter-cellular material handling cost per week

C_{ij} = cost of one unit distance of handling a unit load between cells i and j

d_{ij} = travelling distance between cells i and j

$NTRIP_{ij}$ = number of trips between cells i and j

A CRAFT algorithm may be used to determine the material handling costs. However, material handling costs of the two solutions have been calculated by multiplying the number of trips times the average cost per trip (see pages 76,77). The procedure for calculation of the annual cost (AC) was discussed on page 78 (eq. 3.6).

Results of the duplication process for the Burbidge's solution are summarized in Table XVll.

TABLE XVll

DUPLICATION RESULTS FOR BURBIDGE'S SOLUTION

NO.	Duplicated Machines	Cells
1	6	1
2	6	3
3	6	4
4	8	1
5	8	4
6	8	5

A comparison of the duplication results in Table XVl and XVll reveals that in the Burbidge's solution two additional machines of type 6 have been bought and assigned to cells 1 and 3. All other duplications in the two cases are the same. Costs due to the purchase of new machines and intracellular material handling costs due to these machines for the two cases are given in Table XV111 (data are from Table XVI).

TABLE XV111

DUPLICATION RESULTS FOR BURBIDGE'S SOLUTION AND THE NEW SOLUTION

			BULBIDGE'S SOLUTION		NEW SOLUTION	
NO.	BOTTLE-NECK MACHINES	PARENT CELLS	COST OF NEW MACHINES	MATERIAL HANDLING COSTS	COST OF NEW MACHINES	MATERIAL HANDLING COSTS
1	6	1	72.01	0	0	60
2	6	3	72.01	0	0	20
3	8	4	53.69	0	53.69	0
4	8	5	53.69	0	53.69	0
COSTS			252.40 +	0	107.38 +	80
TOTAL COST			252.40		187.38	

The results in Table XVIII show that with the given data (TABLES X11 and XIII), the total cost of the new solution is 187.38, while the total cost of the Burbidge's solution is 251.40. By chosing the new solution the saving will be 64.02 monetary units per week.

Sensitivity Analysis

Due to uncertainty about some of the model's parameters, it is necessary to evaluate the sensitivity of results to changes in these parameters. Three major sensitivity analyses are included:

(1) Sensitivity to similarity measure
(2) Sensitivity to production volume
(3) Sensitivity to cost coefficients.

Sensitivity to Similarity Measure

It was mentioned before that the results of a clustering algorithm depend upon the type of similarity coefficients used for the clustering purpose. Several types of similarity coefficients were discussed in Chapter IV (Equations 4.9-4.12). Some of these similarity coefficients are not appropriate for the machine-component grouping process.

In this work, two machines are said to be similar, if among the parts having operations on the two machines, some visit both machines. If a part has operations on both machines i and j, its related bits in machine vectors i and j score one. That means a match has occurred between the bits scoring one (a one match). On the other hand, if a part does not have any operations on either of the two machines, its bits in machine vectors i and j are zero. In the latter case, a match between bits scoring zero has occurred (a zero match). As can be seen, on the contrary to

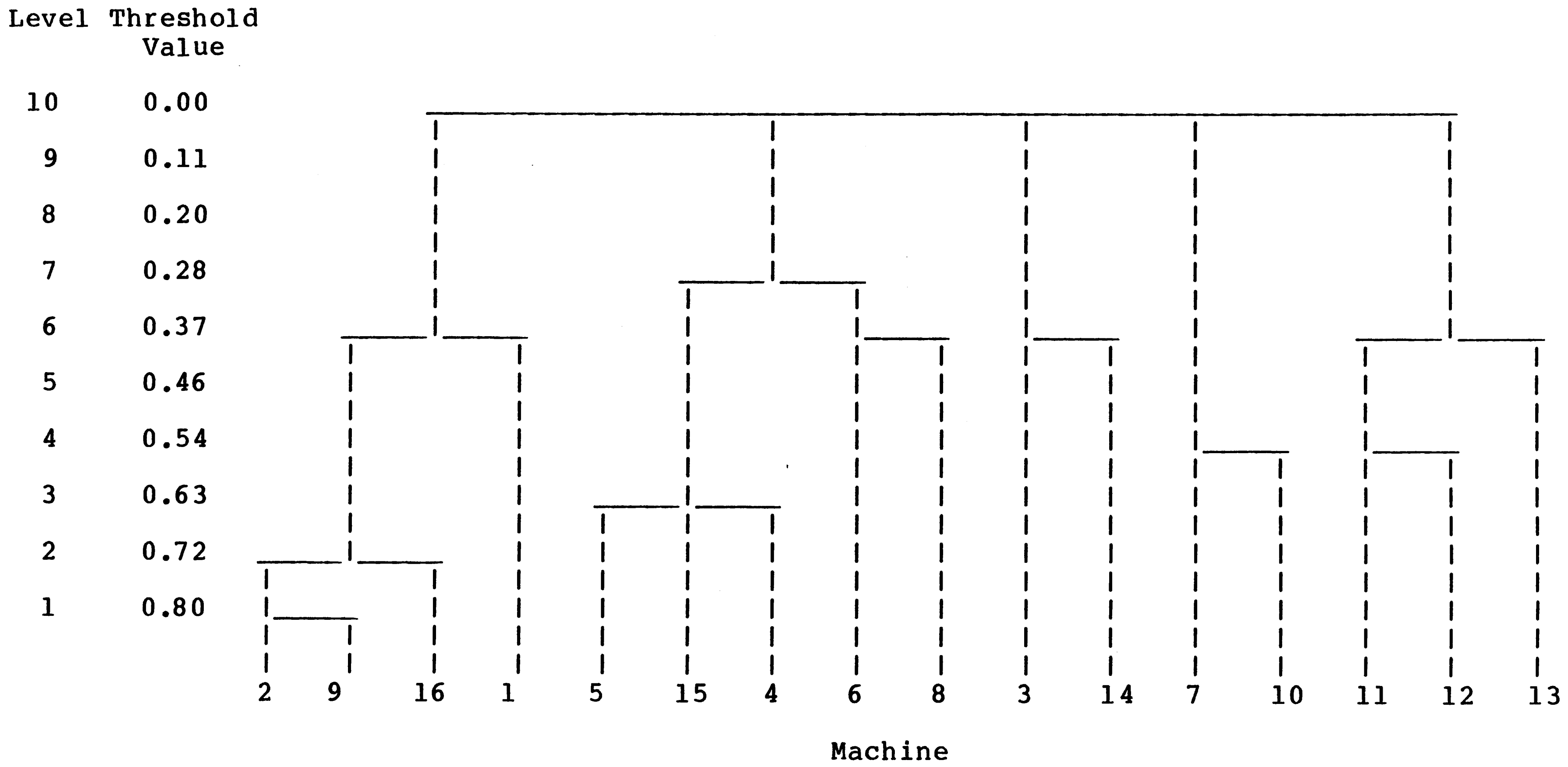


Figure 41. Dendogram For Dice Similarity Coefficient

one matches, zero matches do not create any similarity between two machines. Therefore, only those similarity coefficients in which the zero matches have been excluded should be used for the machine-component grouping process.

Among the similarity coefficients in which zero matches have been excluded, the following three coefficients have been chosen for the sensitivity analysis:

(1) Jaccard's similarity coefficient

(2) Dice's similarity coefficient

(3) A similarity coefficient in which the unmatched pairs are weighted by two.

Each of these similarity coefficients are used to determine the similarity between machines i and j.

Jaccard's similarity coefficient is written as (Equation 4-11),

$$S_{ij} = a/(a+b+c)$$

where,

S_{ij} = the similarity coefficient between machines i and j

a = the number of bits scoring one in both machine vectors (one matches)

b,c = the number of bits scoring one in at least one of the two machine vectors

In this similarity coefficient, the matched (a) and unmatched (b,c) pairs have been considered equally important and have been given the same weights. This similarity coefficient has been used to solve the test problem.

The Dice's similarity coefficient can be written as

(Equation 4-12),

$$S_{ij} = 2a/(2a+b+C)$$

In this similarity coefficient the matched pairs are considered to be more important than the unmatched pairs. The matched pairs are weighted by two, while the weight of the unmatched pairs is one.

Finally, in the third similarity coefficient, the unmatched pairs are weighted by two, while the matched pairs have a weight of one. This similarity coefficient can be written as,

$$S_{ij} = a/[a + 2(b+C)]$$

The results for Jaccard's similarity coefficient have been already presented and discussed (Figures 29, 30). The dendograms for the second and third similarity coefficients are depicted in Figures 41 and 42. The comparison of the dendograms in Figures 30 and 41 reveals some changes in the machine cells in the two cases. First, in the dendogram in Figure 41, the machine cells join together at higher similarity coefficients. Secondly, the cell formation processes in the two cases are different. For example, in Figure 30, machines 5 and 15 join together at the threshold value of 0.49, and then machine 4 joins them at the threshold value of 0.41. In Figure 41, however, machines 4, 5, and 15 join together in one step at the threshold value of 0.63.

When the third similarity coefficient is used, the dendogram (Figure 42) is similar to the dendogram in Figure 30, except the cells are joined together at lower threshold values. For a threshold value as low as 0.10, the three similarity coefficients give the same results.

Sensitivity to Production Volume

To determine the sensitivity of the results (for the duplication process) to the production volume, the production requirements of the exceptional parts (Table XII) have been changed by -20, -10, 10, and 20%. The results indicate that the changes in the production volume affect the duplication process in two ways. First, the changes in the production volume may bring about some changes in the machine requirements of the individual cells. In some

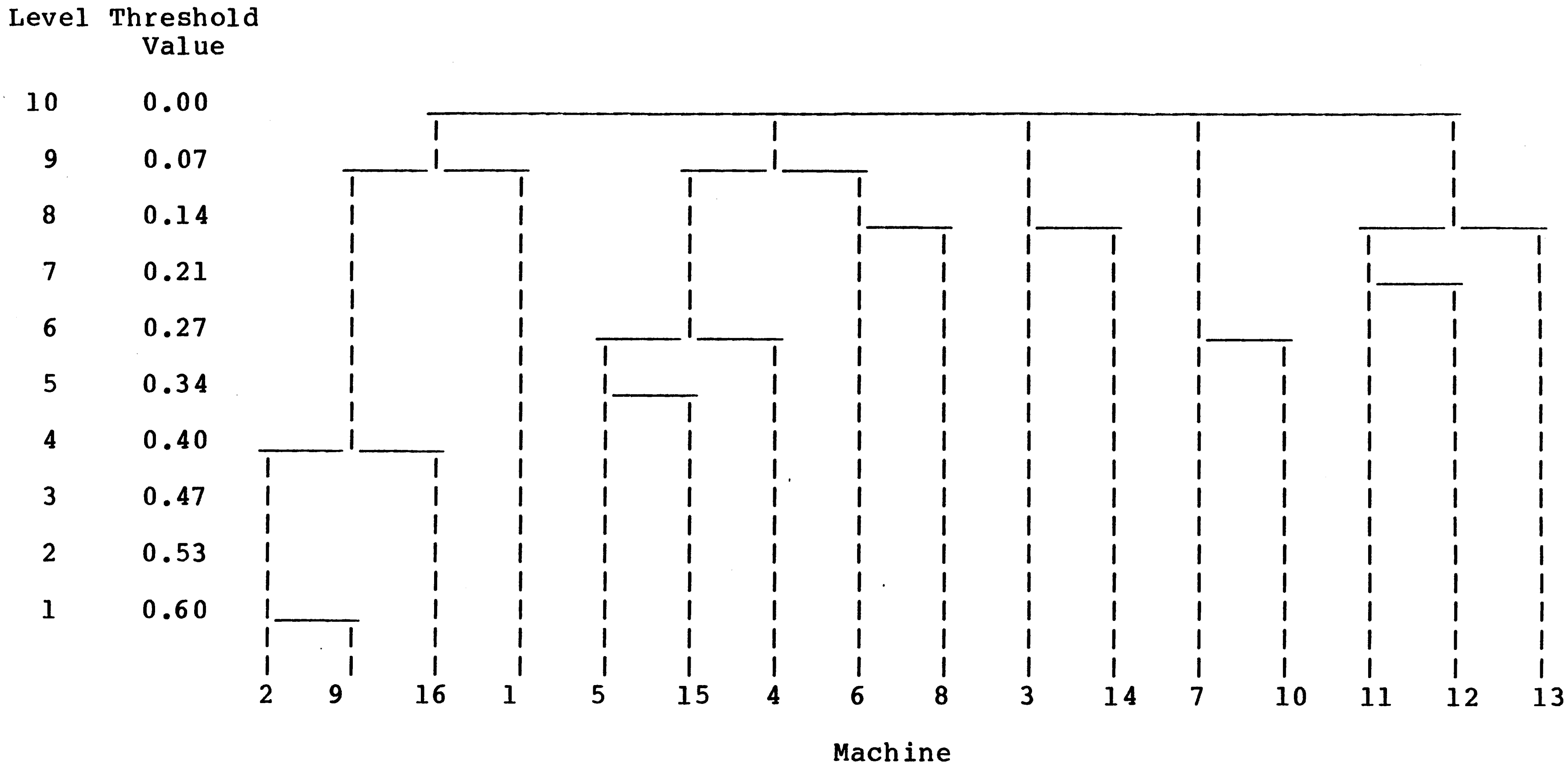


Figure 42. Dendogram For the Third Similarity Coefficient

cases, the changes in the machine requirements of a cell eliminate the need for the purchase of a new machine, in other cases it creates the need for acquiring a new machine.

Secondly, the production volume determines the amounts of inter-cellular trips, which, in turn, determine the inter-cellular material handling cost. This material handling cost is one of the major factors affecting the decision regarding the purchase of a new machine. Therefore, any change in the production volume will have some effects on the duplication process. The results of the sensitivity analysis are given in Tables XIX through XXII.

TABLE XIX

DUPLICATION PROCESS FOR 20% DECREASE IN PRODUCTION VOLUME

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	-----	-----	Not Required
2	6	3	16	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	33.63	53.69	No
5	8	4	48.00	53.69	No
6	8	5	-----	-----	Not Required

TABLE XX

DUPLICATION PROCESS FOR 10% DECREASE IN PRODUCTION VOLUME

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	27.00	72.01	No
2	6	3	18.00	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	53.68	53.69	No
5	8	4	54.00	53.69	Yes
6	8	5	32.17	53.69	No

TABLE XXI

DUPLICATION PROCESS FOR 10% INCREASE IN PRODUCTION VOLUME

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	93.00	72.01	Yes
2	6	3	22.00	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	-----	-----	Not Required
5	8	4	66.00	53.69	Yes
6	8	5	80.17	53.69	Yes

TABLE XXII

DUPLICATION PROCESS FOR 20% INCREASE IN PRODUCTION VOLUME

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	126.00	72.01	Yes
2	6	3	24.00	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	-----	-----	Not Required
5	8	4	72.00	53.69	Yes
6	8	5	-----	-----	Not Required

The final machine-component chart (after duplication) for two extreme cases where the production volume is changed by ±20% are presented in Figures 43 and 44.

Sensitivity to Cost Coefficients

As mentioned before, the economic analysis of the duplication process is done based on two major cost factors: the cost incurred due to the purchase of a bottleneck machine; and the inter-cellular material handling cost due to the lack of that machine.

The material handling cost depends upon the average transportation cost of a unit load in a unit travelling distance (ACOST). In most real world situations, it is not

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	6	8	9	12	14	15	16	19	21	23	29	31	33	34	41	43	17	35	36	1	13	25	26	39	3	11	20	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
6	1		1				1	1		1	1																																
5												1		1	1		1	1	1	1	1	1	1		1		1	1															
15												1					1			1	1				1		1	1															
4												1			1		1			1	1	1	1																				
6													1	1		1	1			1		1			1	1		1	1														
8	1					1		1	1					1	1	1		1		1	1	1		1			1	1				1											
3			1																							1			1	1	1												
14	1												1																1	1													
7																																1	1	1									
10																1								1								1	1	1	1	1							
6																																1	1			1							
11															1																						1		1		1	1	1
12																																						1		1	1	1	1
13																																					1				1		
8																																					1	1	1		1	1	

Figure 43. Final Machine-Component Chart for 20% Decrease in Production Volume

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	6	8	9	14	15	16	19	21	23	29	33	34	41	43	17	35	36	1	12	13	25	26	31	39	3	11	20	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
6	1		1				1	1		1	1																																
8	1					1		1	1																																		
5												1		1	1	1	1	1	1	1	1	1	1		1	1																	
15												1				1			1	1			1		1	1																	
4												1			1	1			1	1	1	1																					
6													1	1		1			1		1		1	1		1	1																
8														1	1		1		1	1	1				1	1																	
3			1																					1			1	1	1														
14	1												1														1	1															
7																														1		1	1										
10																														1	1	1	1	1	1	1							
6																														1	1	1			1								
8																														1	1				1								
11															1																						1		1		1	1	1
12																																						1		1	1	1	1
13																																					1				1		
8																																					1	1	1		1	1	

Figure 44. Final Machine-Component Chart for 20% Increase in Production Volume

possible to determine such an average cost without uncertainty. For this reason, the results of the duplication for different possible values of this average cost should be examined. In the illustrative case presented here, the results when the average cost takes the values of 0.5, 1.0, 1.5, 2.0, and 2.5 have been determined. The results of the duplication process for ACOST = 0.5 are presented in Table XXII and Figure 45. The results for values of 1.0, 1.5, 2.0, and 2.5 are similar to the results of the duplication process presented in Table XVI.

TABLE XXIII

DUPLICATION PROCESS FOR ACOST = 0.5

No.	Machine	Cell	Material Handling Cost (weekly)	Cost Due to Purchase of a New Machine (weekly)	Buying New Machine
1	6	1	30.00	72.01	No
2	6	3	10.00	72.01	No
3	6	4	-----	-----	Not Required
4	8	1	-----	-----	Not Required
5	8	4	30.00	53.69	No
6	8	5	28.09	53.69	No

Components

Machines	2	4	7	10	18	28	32	37	38	40	42	5	6	8	9	11	12	14	15	16	19	20	21	23	29	31	33	34	41	43	17	35	36	1	13	25	26	39	3	22	24	27	30
2	1			1		1	1	1	1	1	1																																
9	1	1		1	1	1	1	1	1	1	1																																
16	1		1	1	1		1	1	1		1																																
1								1			1																																
8	1					1		1	1																																		
5												1		1	1			1	1	1	1		1	1	1		1		1	1													
15												1						1			1		1				1		1	1													
4												1			1			1			1		1	1	1																		
6	1		1				1	1		1	1		1	1			1	1			1			1			1	1		1	1												
8														1	1	1	1		1		1	1	1	1		1			1	1				1					1		1	1	
3			1																									1			1	1	1										
14	1												1																		1	1											
7																																		1	1	1							
10																	1									1								1	1	1	1	1					
6																																		1	1			1					
11															1							1																	1		1	1	1
12																1																								1	1	1	1
13																																							1		1		

Figure 45. Final Machine-Component Chart (ACAST = 0.5)

CHAPTER VI

SUMMARY AND CONCLUSIONS

The survival of any organization in the highly competitive manufacturing environment of today calls for the continuous improvement of existing manufacturing techniques as well as the introduction of new scientific principles in improving production systems. One such principle that has been very effective in solving many manufacturing problems, especially those of batch-type manufacturing, is Group Technology. The application of Group Technology to the manufacturing process begins with finding part-families and forming machine cells. Therefore, the machine-component grouping process is an integral part of the application of Group Technology to the manufacturing process.

The objective of this research has been to develop a cost based model which can effectively deal with the machine-component grouping problem. The work done through this research has made two major contributions to the area of machine-component grouping. First, it has broadened the application of the similarity coefficient method. Second, it has introduced economic analysis to the machine-component grouping process.

The similarity coefficient method is an analytical procedure which is used to cluster similar objects together. The machine-component grouping model based on this method and with the new capabilities developed through this research effectively deals with the machine-component grouping problem. The cost based machine-component grouping model developed through this work has the following capabilities.

(1) It eliminates one of the major drawbacks of the existing models (models based on the similarity coefficient method) by dealing with the problem of bottleneck machines. The new model identifies the bottleneck machines, determines the number of inter-cellular moves for each of them, and duplicates these machines if necessary. The development of this new capability provides a practical ground for the application of the similarity coefficient method to the machine-component grouping process.

(2) The present model overcomes the problem of selecting a proper threshold value; it uses the inter-cellular and intra-cellular material handling costs as a basis for the selection of a proper threshold value.

(3) The final results of the machine-component grouping process are presented in a matrix form. In fact, the initial machine-component chart is modified to reflect the changes introduced by the machine-component grouping process. The modified chart is used to show the machine-component groups formed by the model. This provides the user with the opportunity to visualize the machine cells and

their related parts on the same chart. With the final results in a matrix form, it is easier to realize the way in which the duplication process affects the structure of the machine cells.

The development of the cost based machine-component grouping model also introduces economic analysis to the machine-component grouping process. In the new model the decision about the duplication of a bottleneck machine is based upon the analysis of the costs incurred and benefits produced by the duplication process.

Finally, the procedures and techniques employed in this work, to a great extent, simplify the machine-component grouping process. The data storage and analysis techniques used in this work reduced the computer storage and computational effort required by the machine-component grouping process.

Recommendations for Further Studies

Group Technology, in general, and machine-component grouping in particular, are relatively new. As a result, the prospect for research in the area of machine-component grouping is great. As an extension to this research the following may be considered:

(1) Imposing an upper and lower limit on the size of machine cells.

(2) Assigning some of the bottleneck machines to a special cell which can be accessed by the cells requiring those

bottleneck machines.

(3) Considering some other cost factors in the duplication process. One such factor is the cost of setup times for parts having operations on the bottleneck machines. In addition, there are many other related areas in which work remains to be initiated or extended.

BIBLIOGRAPHY

1. Abou-zeid, M. R., "Group Technology," _Industrial Engng_, Vol. 7, 1975, pp. 32-39.

2. Anderberg, M. R., _Cluster Analysis For Applications_, Academic Press, New York, 1973.

3. Beeby, W. D. and Thompson, A. R., "A Broader View of Group Technology," _Computer and Industrial Engng_, Vol. 3, 1979, pp. 298-312.

4. Biegel, J. E., _Production Control: A Quantitative Approach_, 2nd ed., Prentice Hall Inc., Englewood Cliffs, N. J., 1971.

5. Bobrowicz, V. F., "The Code System," _SME Group Technology Seminar_, February 1976.

6. Brankamp, K., "Objectives, Layout, and Possibilities of a Workpiece Classification System," _Intern Seminar on Group Technology_, Turin, Italy, September 1969.

7. Burbidge, John L., "Production Flow Analysis," _Seminar on the First Steps to Group Technology_, Birneihill Institute, East Kilbride, Glasgow, 1970.

8. Burbidge, John L., "Production Flow Analysis on the Computer," _Third Annual Conference of the Institution of Production Engineers_, 1973.

9. Burbidge, John L., _The Introduction of Group Technology_, Wiley, New York, 1975.

10. Burbidge, John L., "A Manual Method of Production Flow Analysis," _Production Engineer_, 1977, pp. 34-38.

11. Burbidge, John L. "The Simplification of Material Flow Systems," _Int. Jour. Prod. Res._, Vol. 20, 1982, pp. 339-347.

12. Carrie, A. S., "Numerical Taxonomy Applied to Group Technology and Plant Layout," _Int. Jour. Prod. Res._, April 1973, pp. 399-416.

13. Chisholm, A. W. J., "Design for Economic Manufacture," _Ann. of CIRP_, Vol. 22, 1973, pp. 243-247.

14. Congaware, Terry A. and Inyong, Ham., "Cluster Analysis Appications for Group Technology Manufacturing Systems," Dep. of Industrial Engng and Management Systems Engng., The Pennsylvania State Univ., Univ. Park, PA, 1981.

15. Cunningham, K. M. and Ogilvie, J. C., "Evaluation of Hierarchic Grouping Techniques: A Preliminary Study," The Computer Journal, Vol. 15, No. 3, 1972, pp. 209-213.

16. De Beer, C., and De Witte, J., "Production Flow Synthesis," Ann. CIRP, Vol. 25, 1978.

17. Defays, D., "An Efficient Algorithm for a Complete Link Method," The Computer Journal, Vol. 20, No. 4, 1977, pp. 364-366.

18. De Witte, J., "The Use of Similarity Coefficients in Production Flow Analysis," Int. Jour. Prod. Res., Vol. 18, No. 4, 1980, pp. 503-514.

19. Durie, F. R. E., "A Survey of Group Technology and Its Potential User's Application in U. K., " The Prod. Engineer, Vol. 49, Feb. 1970, pp. 51-61.

20. Durie, F. R. E., "A Survey of Group Technology and Its Potential User's Application in U. K.," Report and Discussion, The Prod. Engineer, March 1970, pp. 108-115.

21. Eckert, R., "Codes and Classification Systems," American Machinist, Vol. 119, Dec. 1975, pp.88-92.

22. Edwards, G. A. B., Readings in Group Technology, The Machinery Press Co., London, 1971.

23. Edwards, G. A. B., "The Family Grouping Philosophy," Int. Jour. Prod. Res., Vol. 9, 1971, pp. 337-352.

24. Elmaghraby, S. E., The Design of Production Systems, Van Nostrand Reinhold Co., New York, 1966.

25. Everitt, B. S., Cluster Analysis, Heinemann, London, 1974.

26. Everitt, B. S., "Unresolved Problems in Cluster Analysis," Biometrics, Vol. 35, March 1971, pp. 169-181.

27. Fisher, L. and Van Noss, J. W., "Admissible Clustering Procedures," Biometrika, Vol. 58, 1971, pp. 91-104.

28. Francis, Richard L. and White, John A., Facilities Layout and Location, Prentice-Hall Inc., Englewood Cliffs, N. J., 1974.

29. Gallagher, C. C., Banerjee, S. K., and Southern, G., "Group Technology in the Ship Building Industry," Int. Jour. Prod. Res., Vol. 12, Jan. 1974, pp. 45-54.

30. Gettleman, K. M., "Organize Production for Parts Not Processes," Modern Machine Shop, Vol. 44, Nov. 1971, pp. 50-60.

31. Gupta, M. Rajiv and Tompkins, James A., "An Examination of the Dynamic Behavior of Part-Families in Group Technology," Int. Jour. Prod. Res., Vol. 20, 1982, pp. 73-86.

32. Hall, P. D., "GT Gets Production into Top Gear," Metalworking Production, Vol. 119, Feb. 1975, pp. 49-51.

33. Haveli, G., "Group Technology Not Classification and Coding," American Machinist, Vol. 122, Aug. 1978, pp. 158-162.

34. Hyer, Lea Nancy, "A Framework for Production Planning and Control of Cellular Manufacturing," Decision Sciences, Vol. 13, 1982, pp. 681-695.

35. Ivanov, E. K., Group Production Organization and Technology, Business Publication Ltd., London, 1968.

36. Jackson, David, Cell System of Production, Business Books Limited, London, 1978.

37. Jardine, N. and Sibson, R., "The Construction of Hierarchic and Non-Hierarchic Classification," Computer Journal, Vol. 11, 1968, pp. 177-184.

38. Kennedy, James N., "A Review of Some Cluster Analysis Methods," AIIE Transactions, Vol. 6, Sept. 1974, pp. 216-226.

39. King, J. R., "Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm," Int. Jour. Prod. Res., Vol. 18, March 1980, pp. 213-237.

40. King, J. R. and Nakarnchai, V., "Machine-Component Group Formation in Group Technology: Review and Extension," Int. Jour. Prod. Res., Vol. 20, 1982, pp. 117-133.

41. Kuiper, F. and Kent, Fisher Lloyd, "A Monte Carlo Comparison of Size Clustering Procedures," Biometrics, Vol. 31, Sept. 1975, pp. 777-783.

42. Marklew, J. J., "A System of Tool Pre-setting and Group Technology for Small Turned Parts," Machinery, Vol. 112, Apr. 1968, pp. 636-642.

43. Marklew, J. J., "The Cell System - A Variety of Applications in the Same Factory," Machinery, Vol. 118, 1971, pp. 442-447.

44. McAuley, John, "Machine Grouping for Efficient Production," The Production Engineer, Vol. 52, Feb. 1972, pp. 53-57.

45. McCormick, W. T., Schweitzer, P. J., and White, T. W., "Problem Decomposition and Data Reorganization by a Clustering Technique," Opns. Res., Vol. 20, 1972, pp. 993-1009.

46. Mitrofanov, S. P., Scientific Principles of Group Technology, parts 1-3, English Translation, National Lending Library for Science and Technology, 1966.

47. Mojena, R., "Heirarchic Grouping Methods and Stopping Rules! An Evaluation," The Computer Journal, Vol. 20, No. 4, pp. 359-362.

48. Muther, Richard, Systematic Layout Planning, CBI Publishing Co. Inc., Boston, Massachusetts, 1973.

49. Opitz, H., A classification System to Describe Work Pieces, Pergamon Press, Oxford, 1970.

50. Opitz, H. and Weindahl, H. P., "Group Technology and Manufacturing Systems for Small and Medium Quantity Production," Int. Jour. Prod. Res., Vol. 9, 1971, pp 181-203.

51. Phillips, H. Rohan and ElGomayel, Joseph, "Group Technology Applied to Product Design," The Manufacturing Productivity Committee, Purdue University, 1977.

52. Pullen, R. D., "A Survey of Cellular Manufacturing Cell," The Production Engineer, Vol. 55, 1976, pp. 451-454.

53. Purcheck, G. F. K., "A Mathematical Classification as a Basis for the Design of Group Technology Production Cells," Prod. Engineer, Vol. 54, Jan. 1975, pp. 35-48.

54. Rajagoplan, R. and Batra, J. L., "Design of Cellular Production Systems" A Graph Theoretic Approach," *Int. Jour. Prod. Res.*, Vol. 13, 1975, pp. 567-579.

55. Ranson, G., *Group Technology*, McGraw-Hill, London, 1972.

56. Ruddell, Reed Jr., *Plant Layout*, Richard D. Irwin, Inc., Homewood, Illinois, 1961.

57. Sibson, R., "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method," *The Computer Journal*, Vol. 16, No. 1, pp. 30-33.

58. Sidders, P. A., "Flow Production of Parts in Small Batches," *Machinery*, Vol. 100, Apr. 1962, pp. 920-929.

59. Sneath, P. H. and Sokal, R. R., *Numerical Taxonomy*, W. H. Freeman & Co., San Francisco, 1973.

60. Sokal, R. R. and Sneath, P. H. A., *Principle of Numerical Taxonomy*, Freeman, 1968.

61. Standish, Thomas A., *Data Structure Techniques*, Addison-Welsley Co., 1980.

62. Tompkins, James A. and Moore, M. James, "Computer Aided Layout: A User's Guide," Publication No. 1 in the Monograph Series, Facilities Planning and Design Division, AIIE, Inc., 1980.

63. Tuttle, H. C., "Parts Grouping Pay Handsomely," *Production*, Vol. 73, May 1974, pp. 99-105.

APPENDIX A

OUTLINE OF THE CRAFT ALGORITHM

OUTLINE OF THE CRAFT ALGORITHM

The Computerized Relative Allocation of Facilities Technique (CRAFT) is the first computerized improvement technique for facilities design. The objective of CRAFT is to develop a layout in which the transportation costs are close to minimum. CRAFT assumes the material handling cost of a trip between the departments is a linear function of the travelling distance. It uses the rectilinear distance between the department centriods to calculate the material handling costs. The CRAFT algorithm takes the initial layout and interchanges the adjacent departments or departments of equal area to achieve a near optimal layout.

The basic inputs to CRAFT inclue a from-to chart, a move-cost chart, and an initial layout. A from-to chart is a square matrix whose entries represent the flow volume among different departments in a facility. One way to measure the flow volume between two departments is to calculate the number of trips (movement of parts) taking place between these departments over a specific period of time. These trips are the entries of a from-to chart which shows the volume of materials flow between departments. Figure 46 shows a from-to chart for a facility with five departments. The entries in the from-to chart are the number of trips per week.

		Departments				
		A	B	C	D	E
	A	-	1	2	1	2
	B	5	-	3	2	2
Departments	C	4	2	-	5	1
	D	2	1	4	-	3
	E	3	1	5	2	-

Figure 46. A From-To Chart for a Facility With Five Departments

A move-cost chart simply shows the costs per unit distance of handling a unit load among different departments, in a facility. CRAFT assumes such unit costs are known.

Finally, the initial layout indicates the area requirements of different departments of the facility to be designed. This initial layout is improved (by interchanging departments) in an iterative process until a near optimal layout is achieved.

A more detailed discussion of CRAFT is given by Francis et al. (28) and Tompkins et al. (62).

APPENDIX B

DESCRIPTION OF VARIABLES USED IN THE COMPUTER PROGRAM FOR THE COST BASED MACHINE-COMPONENT GROUPING MODEL

```
C              LIST OF VARIABLES USED IN THE COMPUTER PROGRAM OF THE
C                   COST BASED MACHINE-COMPONENT GROUPING MODEL
C
C  ACOST             THE AVERAGE COST OF MOVING A UNIT LOAD ONE
C                    UNIT DISTANCE
C
C  ADIST             THE AVERAGE TRAVELLING DISTANCE OF THE
C
C                    INTER-CELLULAR MOVES
C
C  BLANK             BLANK CHARACTER( HOLDS FOUR BLANK CHARACTERS
C                    FOR DRAWING THE DENDOGRAM)
C
C   COST             COST OF INTER-CELLULAR TRIPS FOR THE BOTTLENECK
C                    MACHINES
C
C  COSTI             INSTALLATION COST OF A BOTTLENECK MACHINE
C
C   DASH             DASH CHARACTER( HOLDS FOUR "-" CHARACTERS FOR
C                    DENDOGRAM)
C
C  DASHI             A VARIABLE HOLDING THREE "-" AND LETTER I
C                    ( FOR DENDOGRAM)
C
C    DEV             A FRACTION OF MACHINE REQUIRED BY A CELL
C
C      H             PRODUCTION HOURS PER DAY
C
C     IB             A COUNTER FOR NUMBER OF DUPLICATED MACHINES
C
C  IBTLK(NBPN)       PART NUMBER FOR NBPN-TH EXCEPTIONAL PART
C
C   IBMS             THE NUMBER OF DUPLICATED MACHINES
C
C  ICELL(MCHN)       THE CELL NUMBER OF THE CELL TO WHICH MACHINE
C                    MCHN BELONGS
C
C  ICONT             FOR TWO MACHINES, THE NUMBER OF PARTS HAVING
C                    OPERATIONS ON AT LEAST ONE OF THE TWO MACHINES
C
C  IDATA(J)          A LOCALIZED NAME FOR IWORD(J)
C
C     ID             AN IDENTIFICATION NUMBER( USED FOR CELL NUMBER)
C
C  IDCEL(NC)         THE CELL NUMBER FOR CELL NC
C
C  IDCLN             ID NUMBER OF THE COLUMN IN THE SIMILARITY MATRIX
C                    RELATING TO THE LARGEST SIMILARITY COEFFICIENT
C
C  IDCLP(NP)         ID NUMBER FOR THE PARENT CELL OF PART NP
C
C  IDMAX(I)          ID NUMBER OF THE COLUMN( IN THE SIMILARITY MATRIX)
C                    RELATING TO THE LARAGEST SIMILARITY COEFFICIENT
C                    IN ROW I
```

```
C
C     IDR             THE LOCATION OF AN ENTRY IN THE SIMILARITY
C                     MATRIX STORED IN SMTRX
C
C   IDROW             ID NUMBER FOR THE ROW( IN THE SIMILARITY MATRIX)
C                     RELATING TO THE LARGEST SIMILARITY COEFFICIENT
C
C   IFINH(N)          THE POINTER TO THE LAST MACHINE IN CELL N
C
C   ILAST             THE LAST STAGE IN THE CLUSTERING PROCESS
C
C      IM             THE NUMBER OF DIGITS IN A CELL NUMBER
C
C   IMCHN             THE STARTING LOCATION OF MACHINE VECTOR I
C                     IN ARRAY IWORD
C
C   IMERG(K)          ONE OF THE TWO CLUSTERS MERGING AT STAGE K
C                     ( THE ONE WITH A SMALLER ID)
C
C      IN             INPUT UNIT NUMBER
C
C   INMBR(I)          THE NUMBER OF MACHINES IN CELL I
C
C      IO             OUTPUT UNIT NUMBER
C
C   IPART(L)          THE PART NUMBER OF THE L-TH EXCEPTIONAL PART
C
C   IPMRG(K)          THE LAST STAGE( BEFORE STAGE K) AT WHICH CLUSTER
C                     I WAS AT MERGE
C
C   IPRTC(N)          CELL VECTOR N
C
C   IPRTS(NP)         THE NUMBER OF PART NP TO BE PRODUCED( EXCEPTIONAL
C                     PARTS)
C
C   ITEMP(J)          A TEMPORARY LOCATION FOR THE J-TH DATA ITEM IN
C                     A MACHINE VECTOR
C
C      IW             A TEMPORARY STORAGE FOR WORD I IN ARRAY IWORD
C
C   IWORD(I)          A STORAGE FOR I-TH NBITS DATA ITEMS IN THE
C                     MACHINE-COMPONENT CHART
C
C   JCONT             FOR TWO MACHINES, THE NUMBER OF PARTS VISITING
C                     BOTH MACHINES
C
C   JMCHN             THE STARTING LOCATION OF MACHINE VECTOR J IN
C                     ARRAY IWORD
C
C   JMERG(K)          ONE OF THE TWO CLUSTERS MERGING AT STAGE K
C                     (THE ONE WITH A LARGER ID)
C
C   JNMBR(J)          THE NUMBER OF MACHINES IN CELL J
C
```

```
C  JPART(I)          PART NUMBER OF THE I-TH EXCEPTIONAL PART
C
C  JPMRG(K)          THE LAST STAGE( BEFORE STAGE K) AT WHICH THE
C                    CLUSTER I WAS AT MERGE
C
C  JTEMP(I)          A TEMPORARY LACATION FOR DATA ITEM I
C
C      K             A COUNTER FOR MERGING LEVELS
C
C  KLAST             A LEVEL IN THE DENDOGRAM PRECEDING LEVEL K
C
C  LASTN             THE LAST MACHINE OF THE PREVIOUS CELL
C
C  LEMIT             THE MAXIMUM ALLOCATED STORAGE(STORG)
C
C  LEVEL(K)          THE LEVEL IN THE DENDOGRAM TO WHICH THE K-TH
C                    CLUSTER BELONGS
C
C  LIMIT             A LIMIT ON THE NUMBER OF INTER-CELLULAR MOVES.
C                    IF A MACHINE IS CREATING A LARGER NUMBER OF
C                    INTER-CELLULAR MOVES THAN LIMIT, IT IS CONSIDERED
C                    FOR DUPLICATION
C
C  LINE(I)           THE LINE NUMBER OF MACHINE I IN THE DENDOGRAM
C
C  LIST(I)           THE MACHINE NUMBER OF THE I-TH MACHINE IN THE
C                    SIMILARITY MATRIX
C
C  MASK(I)           THE I-TH MASK
C
C    MAX             THE MAXIMUM STORAGE REQUIRED FOR DIFFERENT ARRAYS
C
C   MAXB             THE MAXIMUM NUMBER OF INTER-CELLULAR MOVES CREATED
C                    BY A BOTTLENECK MACHINE
C
C     MB             A BOTTLENECK MACHINE
C
C  MBTLK             THE BOTTLENECK MACHINE CREATING THE LARGEST NUMBER
C                    OF INTER-CELLULAR MOVES
C
C   MCHN             A MACHINE NUMBER
C
C  MCHNB(I)          CELL NUMBERS IN WHICH MACHINE I IS DUPLICATED
C
C  MCHND(I)          THE MACHINE NUMBER OF THE I-TH DUPLICATED MACHINE
C
C  MCHNS(I)          THE MACHINE VECTOR FOR THE DUPLICATED MACHINE I
C
C     MK(I)          THE SEQUENCE OF OPERATION FOR PART NP
C
C   MNOP             MAXIMUM NUMBER OF MACHINES VISITED BY A PART
C                    (IN DIFFERENT CELLS)
C
C  N1 TO N27         LENGTHS OF DIFFERENT ARRAYS USED IN THE COMPUTER
```

```
C                   PROGRAM( SEE SUBROUTINE ALOCT)
C
C   NAVL            THE NUMBER OF MACHINES REQUIRED BY A CELL( THE
C                   BOTTLENECK MACHINES)
C
C  NBITS            NUMBER OF BITS PER COMPUTER WORD( COMPUTER
C                   SPECIFICATION)
C
C  NBTLK(MCHN)      THE NUMBER OF INTER-CELLULAR MOVES CREATED
C                   BY MACHINE MCHN
C
C   NBTS            A LOCALIZED NAME FOR NBITS
C
C     NC            A COUNTER FOR NUMBER OF CELLS
C
C  NCELS            THE NUMBER OF CELLS FOR A GIVEN THRESHOLD VALUE
C
C  NEXT(I)          THE NEXT LEVEL AT WHICH CLUSTER I WILL BE AT MERGE
C
C  NLINE(I)         THE MACHINE NUMBER ASSOCIATED WITH LINE I
C                   ( IN THE DENDOGRAM)
C
C  NCLSR            A COUNTER FOR THE NUMBER OF CLUSTERS
C
C  NMCHN            THE NUMBER OF MACHINE TYPES IN THE MACINE-COMPONENT
C                   CHART
C
C  NMOVE            NUMBER OF TIMES PART I MOVES BETWEEN TWO CELLS
C                   OR TWO MACHINES
C
C    NOP            THE NUMBER OF OPERATIONS OF A SPECIFIC PART
C                   IN A CELL
C
C  NPART            THE NUMBER OF PART TYPES IN THE MACHINE-COMPONENT
C                   CHART
C
C  NPRTC(J)         THE PART NUMBER OF THE J-TH PART IN THE
C                   MACHINE-COMPONENT CHART
C
C  NPRTS(J)         THE NUMBER OF PART J TO BE PRODUCED
C
C  NTRIP(MCHN)      THE NUMBER OF TRIPS BETWEEN TWO CELLS
C                   CREATED BY MACHINE MCHN
C
C  NTRPW            THE TOTAL NUMBER OF INTER-CELLULAR TRIPS
C
C  NWORD            THE NUMBER OF COMPUTER WORDS REQUIRED TO
C                   STORE A MACHINE VECTOR( THE DATA IN ONE
C                   ROW OF THE MACHINE-COMPONENT CHART)
C
C  NXP              NUMBER OF EXCEPTIONAL PART
C
C  OCOST            THE OPERATING COST OF A BOTTLENECK MACHINE
C
```

```
C  PRCNT            THE PERCENTAGE OF OCCUPANCY WHICH CALLS FOR
C                   AN ADDITIONAL MACHINE
C
C  PRICE            THE PRICE OF A BOTTLENECK MACHINE
C
C      R            THE REQUIRED RATE OF RETURN ON INVESTMENT
C
C  SMALL            A SMALL NUMBER FOR INITIALIZING SSMAX
C
C  SMERG(K)         THE SIMILARITY COEFFICIENT OF THE K-TH MERGE
C
C  SMTRX(I)         THE SIMILARITY COEFICIENT OF THE I-TH PAIR
C
C  SSMAX            THE LARGEST SIMILARITY COEFFICIENT IN THE
C                   CURRENT SIMILARITY MATRIX
C
C   STEP            THE STEP SIZE FOR CALCULATING DIFFERNT THRESHOLD
C                   VALUES
C
C  STEPS(K)         THE THRESHOLD VALUE OF THE K-TH LEVEL
C
C  STORG            A MASTER ARRAY ALLOCATING THE TOTAL STORAGE
C                   REQUIRED FOR DIFFERENT ARRAYS
C
C  SVALU            SALVAGE VALUE OF A BOTTLENECK MACHINE
C
C   TIME            THE SUM OF THE PROCESSING TIMES OF THE EXCEPTIONAL
C                   PARTS ON A BOTTLENECK MACHINE
C
C  TIMES(I)         THE PROCESSING TIME OF I-TH EXCEPTIONAL PART ON
C                   THE CRITICAL BOTTLENECK MACHINE
C
C  UFCTR            USE FACTOR FOR A BOTTLENECK MACHINE
C
C  ULIFE            USEFUL LIFE OF A BOTTLENECK MACHINE
C
C  TRI(I,J)         A SEGMENT OF THE DENDOGRAM ON LINE I AND LEVEL J
C
C  WCOST            WEEKLY COST
C
C  WHOUR            WEEKLY PRODUCTION HOURS
C
C  WWEEK            THE NUMBER OF WORKING WEEKS PER YEAR
C
C     X1            A TEMPORARY STORAGE FOR ICONT
C
C     X2            A TEMPORARY ATORAGE FOR JCONT
C
C
C
C
C
C
C
```

APPENDIX C

FORTRAN CODES FOR MACHINE-COMPONENT GROUPING MODEL

```
C
C
C                    ********************************************
C                    **                                        **
C                    **   FORTRAN CODES FOR COMPUTER PROGRAM   **
C                    **    OF THE COST BASED MACHINE-COMPO-    **
C                    **          NENT GROUPING MODEL           **
C                    **                                        **
C                    **                   BY                   **
C                    **            HAMID SEIFODDINI            **
C                    **              SUMMER 1984               **
C                    **                                        **
C                    ********************************************
C
C
C  THIS COMPUTER PROGRAM IS COMPOSED OF 30 ROUTINES AND PERFORMS A
C  VARIETY OF FUNCTIONS PLANNED FOR THE MACHINE-COMPONENT GROUPING
C  MODEL
C
C
C
C                    ********************************************
C                    **                                        **
C                    **               MAIN ROUTINE             **
C                    **                                        **
C                    ********************************************
C
C  THIS IS AN ADMINISTRATIVE ROUTINE WHICH ESTABLISHES LINES OF COMUNI-
C  CATION AMONG DIFFERENT SUBROUTINES OF THE CLUSTERING UNIT. IT CALLS
C  SUBROUTINES ALOCT,INPUT,SMLTY,CLSTR,AND TREE
C  INPUT FORMAT:2I2,I3,I5
C
      DIMENSION STORG(700)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST2/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
     .N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27
C
C  SET INPUT AND OUTPUT UNITS NUMBER
C
      IN=5
      IO=6
C
C  READ CONTROL DATA
C
      READ(UNIT=IN,FMT=1200) NBITS,NMCHN,NPART,LEMIT
C
C  DETERMINE THE NUMBER OF COMPUTER WORDS -NWORD
C
      NWORD=NPART/NBITS
      IF(NWORD*NBITS.LT.NPART) NWORD=NWORD+1
C
C  ALLOCATE THE STORGE -STORG
C
      CALL ALOCT(LEMIT,NMCHN,NWORD,NPART)
```

```
C
C  GET INPUTS
C
      CALL INPUT(STORG(N1),STORG(N2),NWORD)
C
C  DETERMINE THE SIMILARITY COEFFICIENTS BETWEEN MCHINE I AND THE
C  REMAINING MACHINES
C
      IMCHN=1
      L=0
      DO 1040 I=2,NMCHN
      II=I-1
      IMCHN=IMCHN+NWORD
      JMCHN=1
      DO 1030 J=1,II
      L=L+1
C
C  CALCULATE THE SIMILARITY COEFFICIENT
C
      CALL SMLTY(STORG(IMCHN),STORG(JMCHN),STORG(N10),L,
     .IMCHN,JMCHN,NWORD)
      JMCHN=JMCHN+NWORD
1030  CONTINUE
1040  CONTINUE
C
C  CLUSTER SIMILAR MACHINES TOGETHER
C
      CALL CLSTR(STORG(N3),STORG(N4),STORG(N5),STORG(N6),STORG(N7),
     .STORG(N8),STORG(N9),STORG(N10),STORG(N11),STORG(N12),STORG(N13),
     .STORG(N14),STORG(N15))
C
C  CONSTRUCT THE DENDOGRAM
C
      CALL TREE(STORG(N3),STORG(N4),STORG(N5),STORG(N6),STORG(N7),STORG(
     .N9),STORG(N10),STORG(N17),STORG(N18),STORG(N19),STORG(N20),STORG(N
     .21),STORG(N22),STORG(N23),STORG(N24),STORG(N25),STORG(N26),
     .STORG(N1),NWORD,STORG(N2))
      STOP
1200  FORMAT(1X,2I2,I3,I5)
      END
      SUBROUTINE ALOCT(LEMIT,NMCHN,NWORD,NPART)
      COMMON /LIST2/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
     .N15,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27
C
C                 ********************************************
C                 **                                        **
C                 **           SUBROUTINE ALOCT             **
C                 **                                        **
C                 ********************************************
C
C  THIS SUBROUTINE DETERMINES THE DIMENSIONS OF STORG(1) TO STORG(26)
C  IT ALSO CHECKS THE MAXIMUM STORAGE REQUIRED AGAINST THE ALLOCATED
C  STORAGE
C
```

```
C
      N1=1
C
C  STORAGE FOR IWORD
C
      N2=N1+NWORD*NMCHN
C
C  STORAGE FOR ITEMP
      N3=N2+NPART
C
C  STORAGE FOR ARRAYS IMERG,JMERG,SMERG,IPMRG,JPMRG,LAST,NEXT,SMTRX,
C  IDMAX,SMAX,LIST,INMBR,JNMBR
C
      N4=N3+NMCHN
      N5=N4+NMCHN
      N6=N5+NMCHN
      N7=N6+NMCHN
      N8=N7+NMCHN
      N9=N8+NMCHN
      N10=N9+NMCHN
      N11=N10+NMCHN*(NMCHN-1)/2
      N12=N11+NMCHN
      N13=N12+NMCHN
      N14=N13+NMCHN
      N15=N14+NMCHN
C
C  STORAGE FOR ARRAYS LINE,TRI,LABLE,NLINE,LEVEL,LAST,ICELL,MCHIN,
C  IDCEL,ISTRT,IFINH
C
      N16=N15+NMCHN-1
      N17=N10+12*NMCHN
      N18=N17+3*NMCHN
      N19=N18+NMCHN
      N20=N19+NMCHN
      N21=N20+NMCHN
      N22=N21+NMCHN-1
      N23=N22+NMCHN
      N24=N23+2*NMCHN
      N25=N24+NMCHN
      N26=N25+NMCHN
      N27=N26+NMCHN
      MAX=N16
      IF(MAX.LT.N27) MAX=N27
C
C  IF THE REQUIRED STORAGE IS GREATER THAN THE ALLOCATED STORAGE GIVE
C  AN ERROR MESSAGE
C
      IF(MAX.GT.LEMIT) CALL ERROR(1)
      RETURN
      END
C
C                   *************************************************
C                   **                                             **
C                   **            SUBROUTINE INPUT                 **
```

```
C                 **                                              **
C                 ************************************************
C
C  THIS SUBROUTINE READS THE INPUTS(RELATING TO PARTS) AND INITIALIZES
C  MASKS
C  INPUT FORMAT:72I1,I2,36I2,36I2
C
      SUBROUTINE INPUT(IDATA,ITEMP,NWORD)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST3/ NPRTS,NSTEP
      DIMENSION IDATA(1),ITEMP(1),MASK(31),NPRTS(44)
C
C  INITIALIZE MASKS
C
      DATA (MASK(I),I=1,30)/Z40000000,Z20000000,Z10000000,
     .Z08000000,Z04000000,Z02000000,Z01000000,
     .Z00800000,Z00400000,Z00200000,Z00100000,
     .Z00080000,Z00040000,Z00020000,Z00010000,
     .Z00008000,Z00004000,Z00002000,Z00001000,
     .Z00000800,Z00000400,Z00000200,Z00000100,
     .Z00000080,Z00000040,Z00000020,Z00000010,
     .Z00000008,Z00000004,Z00000002/
      DATA MASK(31)/Z00000001/
      L=0
C
C  DETERMINE THE MIDEL ROW IN THE MACHINE-COMPONENT CHART
C
      MDL=NMCHN/2
C
C  TITLE FOR THE ORIGINAL MACHINE-COMPONENT CHART
C
      WRITE(IO,1000) (I,I=1,NPART)
1000  FORMAT(1X,50X,'PARTS'//11X,43I2/)
C
C  FOR EACH MACHINE READ ONE ROW IN THE MACHINE-COMPONENT CHART
C
      DO 1040 I=1,NMCHN
      READ(UNIT=IN,FMT=1200) (ITEMP(K),K=1,NPART)
      WRITE(IO,1005) (ITEMP(K),K=1,NPART)
1005  FORMAT(1X,I10,43(1X,I1)/)
      IF(I.EQ.MDL) WRITE(IO,1006)
1006  FORMAT(1X,'MACHINES')
      M=0
      DO 1020 J=1,NWORD
      L=L+1
      IDATA(L)=0
      DO 1010 JJ=1,NBITS
      M=M+1
      IF(ITEMP(M).NE.1) GO TO 1010
      IDL=IDATA(L)
      MJJ=MASK(JJ)
      IDATA(L)=IOR(IDL,MJJ)
1010  CONTINUE
1020  CONTINUE
```

```
1040  CONTINUE
C
C  READ THE NUMBER OF LEVELS IN THE DENDOGRAM AND THE NUMBER OF EACH
C  PART TO BE PRODUCED
C
      READ(UNIT=IN,FMT=1300)  NSTEP
      READ(UNIT=IN,FMT=1400)  (NPRTS(L),L=1,36)
      READ(UNIT=IN,FMT=1500)  (NPRTS(L),L=37,44)
      RETURN
1200  FORMAT(43I1)
1300  FORMAT(I2)
1400  FORMAT(36I2)
1500  FORMAT(8I2)
      END
C
C                  ********************************************
C                  **                                        **
C                  **            SUBROUTINE SMLTY            **
C                  **                                        **
C                  ********************************************
C
C  THIS  SUBROUTINE CALCULATES ALL PAIRWISE SIMILARITY COEFFICIENTS
C  AND CONSTRUCTS THE INITIAL SIMILARITY MATRIX. THE SIMILARIT MAT-
C  RIX IS STORED IN ARRAY SMTRX. THE  LOCATIN OF A SIMILARITY COEF-
C  FICIENT IN SMTRX IS FOUND BY FUNCTION INDXS.
C  THE FOLLOWING LOCAL VARIABLES HAVE BEEN USED IN SMLTY:
C  IT1   THE RESULT OF IWORD.OR.JWORD
C  IT2   THE RESULT OF IWORD.AND.JWORD
C  IT11  A TEMPORARY LOCATION FOR IT1
C  IT22  A TEMPORARY LOCATION FOR IT2
C   I1   THE RESULT OF SHIFT OPERATION ON IT1
C   I2   THE RESULT OF SHIFT OPERATION ON IT2
C
      SUBROUTINE SMLTY(IWORD,JWORD,SMTRX,L,IMCHN,JMCHN,NWORD)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST2/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
     .N15,N16,N17,N18,N19,N20,N21,N22
      DIMENSION IWORD(1),JWORD(1),SMTRX(1)
C
C  INITIALIZE COUNTERS
C
      ICONT=0
      JCONT=0
      DO 1010 N=1, NWORD
      IW=IWORD(N)
      JW=JWORD(N)
C
C  FIND THE RESULTS OF .OR. AND .AND. OPERATIONS ON IT1 AND IT2
C
      IT1=IOR(IW,JW)
      IT2=IAND(IW,JW)
      IT11=IT1
      IT22=IT2
C
```

```
C  COUNT THE NUMBER OF BITS SCORING ONE IN IT11 AND IT22
C
      DO 1005 M=1,NBITS+1
      M1=M-NBITS-1
      M2=-M1
      I1=ISHFT(IT11,M1)
      ICONT=ICONT+I1
      IT11=IBCLR(IT11,M2)
      I2=ISHFT(IT22,M1)
      JCONT=JCONT+I2
      IT22=IBCLR(IT22,M2)
1005  CONTINUE
1010  CONTINUE
C
C  CHECK FOR ZERO IN DENOMINATOR
C
      IF(ICONT.EQ.0) GO TO 1020
C
C  CALCULATE THE SIMILARITY COEFFICIENTS AND STORE THEM IN SMTRX
C
      X1=ICONT
      X2=JCONT
      SMTRX(L)=X2/X1
1020  RETURN
      END
C
C
C       ****************************************************************
C       **                                                            **
C       **  THE THREE SUBROUTINES CLSTR,REVIS,AND TREE ARE BASED      **
C       **  ON THE CLUSTERIG ALGORITHMS DISCUSSED BY M. R.            **
C       **  ANDERBERG( REFERENCE 2 IN THE BIBLIOGRAPHY).              **
C       **                                                            **
C       ****************************************************************
C
C
C                   ******************************************
C                   **                                      **
C                   **           SUBROUTINE CLSTR           **
C                   **                                      **
C                   ******************************************
C
C  THIS SUBROUTINE CLUSTERS TOGETHER THE TWO MOST SIMILAR MACHINES AT
C  EACH ITERATION. THEN, THE REVIS SUBROUTINE IS CALLED AND THE SIMI-
C  LARITY MATRIX IS REVISED.
C  THE FOLLOWING LOCAL VARIABLES ARE USED:
C  L   THE NEXT LOCATION IN SMTRX
C  II  THE NUMBER OF ENTITIES IN ROW I OF THE SIMILARITY MATRIX
C
      SUBROUTINE CLSTR(IMERG,JMERG,SMERG,IPMRG,JPMRG,LAST,NEXT,SMTRX,
     .IDMAX,SMAX,LIST,INMBR,JNMBR)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST2/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
     .N15,N16,N17,N18,N19,N20,N21,N22
```

```
      DIMENSION IMERG(1),JMERG(1),SMERG(1),IPMRG(1),JPMRG(1),LAST(1),
     .NEXT(1),SMTRX(1),IDMAX(1),SMAX(1),LIST(1),INMBR(1),JNMBR(1)
C
C  INITIALIZE VARIABLES AND ARRAYS
C
      SMALL=-1
      K=1
      NCLSR=NMCHN
      DO 1010 I=1,NCLSR
      LAST(I)=0
      NEXT(I)=0
      INMBR(I)=1
      LIST(I)=I
      SMAX(I)=SMALL
1010  CONTINUE
C
C  FIND THE LARGEST SIMILARITY COEFFICIENT IN EACH ROW OF SMTRX
C
      L=0
      DO 1020 I=2,NCLSR
      II=I-1
      DO 1015 J=1,II
      L=L+1
      IF(SMAX(I).GT.SMTRX(L)) GO TO 1015
      SMAX(I)=SMTRX(L)
      IDMAX(I)=J
1015  CONTINUE
1020  CONTINUE
C
C  FIND THE LARGEST SIMILARITY COEFFICIENT IN SMTRX
C
1025  SSMAX=SMALL
      DO 1030 J=2,NCLSR
      I=LIST(J)
      IF(SMAX(I).LT.SSMAX) GO TO 1030
      SSMAX=SMAX(I)
      IDROW=I
      IROW=J
1030  CONTINUE
C
C  IDENTIFY THE ROW AND COLUMN OF SMTRX ASSOCIATED WITH THE EXTREME
C  VALUE
C
      IDCLN=IDMAX(IDROW)
C
C  RECORD MERGE DATA(FOR SUBROUTINE TREE)
C
      IMERG(K)=IDCLN
      JMERG(K)=IDROW
      SMERG(K)=SSMAX
      IPMRG(K)=LAST(IDCLN)
      JPMRG(K)=LAST(IDROW)
      LAST(IDCLN)=K
      IF(IPMRG(K).EQ.0) GO TO 1040
```

```
      IPK=IPMRG(K)
      NEXT(IPK)=K
1040  IF(JPMRG(K).EQ.0) GO TO 1050
      JPK=JPMRG(K)
      NEXT(JPK)=K
1050  K=K+1
C
C  TERMINATE IF ALL MACHINES ARE MERGED
C
      IF(K.GE.NMCHN) GO TO 1100
C
C  UPDATE FOR NEXT LEVEL
C
      NCLSR=NCLSR-1
      IF(IROW.GT.NCLSR) GO TO 1070
C
C  REMOVE THE MERGED ENTRY(IDROW) FROM THE LIST AND UPDATE LIST
C
      DO 1060 J=IROW,NCLSR
      LIST(J)=LIST(J+1)
1060  CONTINUE
C
C  REVISE THE SIMILARITY MATRIX
C
1070  CALL REVIS(SMTRX,IDMAX,SMAX,SSMAX,INMBR,JNMBR,LIST,IDROW,IDCLN,
     .NCLSR,NMCHN)
      GO TO 1025
1100  RETURN
      END
C
C                      ********************************************
C                      **                                        **
C                      **           SUBROUTINE REVIS             **
C                      **                                        **
C                      ********************************************
C
C  THIS SUBROUTINE REVISES THE SIMILARITY MATRIX ANY TIME A NEW CELL IS
C  FORMED. THE CALCULATION OF THE SIMILARITY COEFFICIENTS IN THIS
C  SUBROUTINE IS BASED ON THE AVERAGE LINKAGE CLUSTERING METHOD.
C
      SUBROUTINE REVIS(SMTRX,IDMAX,SMAX,SSMAX,INMBR,JNMBR,LIST,IDROW,IDC
     .LN,NCLSR,NMCHN)
      DIMENSION SMTRX(1),IDMAX(1),SMAX(1),INMBR(1),JNMBR(1),LIST(1)
C
C  DETERMINE THE NUMBER OF MACHINES IN THE NEWLY FORMED CELL
C
      INMBR(IDCLN)=INMBR(IDCLN)+INMBR(IDROW)
      DO 1010 J=1,NCLSR
C
C  FIND THE LOCATIN OF ENTRY I AND IDROW OF THE SIMILARITY MATRIX
C  IN SMTRX.
C
      I=LIST(J)
      IF(I.EQ.IDCLN) GO TO 1010
```

```
      IDR=INDXS(I,IDROW)
      IDC=INDXS(I,IDCLN)
C
C  FIND THE SUM OF SIMILARIT COEFFICEINTS BETWEEN THE NEWLY FORMED
C  CELL AND THE EXISTING CELLS
C
      SMTRX(IDC)=SMTRX(IDC)+SMTRX(IDR)
1010  CONTINUE
      DO 1020 J=1,NCLSR
      I=LIST(J)
C
C  SKIP THE ROWS PRIOR TO IDROW
C
      IF(I.EQ.IDCLN) GO TO 1030
1020  CONTINUE
1030  IF(J.EQ.1) GO TO 1050
1035  JJ=J-1
      SMAX(I)=SMALL
      DO 1040 L=1,JJ
      J1=LIST(L)
      IDR=INDXS(I,J1)
C
C  FIND THE NEW EXTREME VALUES OF EACH ROW
C
      SSMAX=SMTRX(IDR)/(INMBR(I)*INMBR(J1))
      IF(SSMAX.LT.SMAX(I)) GO TO 1040
      SMAX(I)=SSMAX
      IDMAX(I)=J1
1040  CONTINUE
1050  J=J+1
      IF(J.GT.NCLSR) RETURN
      I=LIST(J)
      IF(IDMAX(I).EQ.IDROW.OR.IDMAX(I).EQ.IDCLN) GO TO 1035
C  K   THE SIMILARITY LEVELS
C  L   A LOCATION IN SMTRX
C
      GO TO 1050
      END
C
C                   ********************************************
C                   **                                        **
C                   **              FUNCTION INDXS            **
C                   **                                        **
C                   ********************************************
C
C  THIS FUNCTION FINS THE LOCATION OF THE SIMILARITY COEFFICIENTS IN
C  ARRAY SMTRX
C
      FUNCTION INDXS(I,J)
      IF(I.GT.J) GO TO 1020
      INDXS=(J-2)*(J-1)/2+I
      RETURN
1020  INDXS=(I-2)*(I-1)/2+J
      RETURN
```

```
      END
C
C                   ********************************************
C                   **                                        **
C                   **            SUBROUTINE ERROR            **
C                   **                                        **
C                   ********************************************
C
C  THIS SUBROUTINE GIVES ERROR MESSAGES FOR UNEXPECTED INPUT VALUES
C
      SUBROUTINE ERROR(IND)
      GO TO (1010,1030,1050) IND
1010  WRITE(IO,1020)
1020  FORMAT(1X,'THE REQUIRED STORAGE EXCEEDS THE STORAGE LIMIT')
      RETURN
1030  WRITE(IO,1040)
1040  FORMAT(1X,'UNEXPECTED INPUT')
1050  RETURN
      END
C
C                   ********************************************
C                   **                                        **
C                   **            SUBROUTINE TREE             **
C                   **                                        **
C                   ********************************************
C
C  THIS SUBROUTINE CONSTRUCTS AND DRAWS THE DENDOGRAM OF THE MACHINE-COM
C  PONENT GROUPING PROBLEM. IT CALLS SUBROUTINE CELLS.
C
      SUBROUTINE TREE(IMERG,JMERG,SMERG,IPMRG,JPMRG,NEXT,TRI,LABL,LINE
     .,NLINE,LEVEL,LAST,ICELL,MCHIN,IDCEL,ISTRT,IFINH,IWORD,NWORD,
     .ITEMP)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST3/ NPRTS,NSTEP
      DIMENSION IMERG(1),JMERG(1),SMERG(1),IPMRG(1),JPMRG(1),NEXT(1),
     .LAST(1),LINE(1),NLINE(1),LEVEL(1),LABL(1),IWORD(1),IDCEL(1),
     .ISTRT(1),IFINH(1),ICELL(1),MCHIN(2,16),ITEMP(1),NPRTC(70),IPRTS(44
     .),IRDER(44),NPRTS(44)
      DIMENSION TRI(10,16)
      DIMENSION STEPS(11)
      DATA DASH,DASHI,BLANK,BLNKI/4H----,4H---I,4H    ,4H   I/
      DATA DASH1/4H---1/
C
C  INITIALIZE
C
      IFRST=1
      ILAST=NMCHN-1
      DO 1020 I=1,NMCHN
      LINE(I)=0
      NLINE(I)=0
      LAST(I)=0
      DO 1010 J=1,NSTEP
      TRI(J,I)=BLANK
1010  CONTINUE
```

```
1020  CONTINUE
C
C  DETERMINE THE STEP SIZE
C
      STEPN=NSTEP
      STEP=(SMERG(ILAST)-SMERG(IFRST))/STEPN
      STEPS(1)=SMERG(IFRST)+STEP
C
C  DETERMINE THE SIMILARITY COEFFICIENTS OF DIFFERENT SIMILARITY LEVELS
C
      DO 1030 I=2,NSTEP-1
      STEPS(I)=STEPS(I-1)+STEP
1030  CONTINUE
      STEPS(NSTEP)=SMERG(ILAST)
C
C  ASSIGN DIFFERENT CLUSTERS TO DIFFERENT SIMILARITY LEVELS ACCORDING TO
C  THE SIMILARITY COEFFICIENTS AT WHICH THEY MERGED.
C
      K=1
      DO 1050 J=1,NMCHN-1
1035  IF(SMERG(J).GE.STEPS(K)) GO TO 1040
      K=K+1
      IF(K.GE.NSTEP) GO TO 1060
      GO TO 1035
1040  LEVEL(J)=K
1050  CONTINUE
1060  DO 1070 L=J,NMCHN-1
      LEVEL(L)=NSTEP
1070  CONTINUE
C
C  BEGIN THE DENDOGRAM WITH THE MOST SIMILAR PAIRS
C
      K=IFRST
      LIN=0
      ILINP=0
C
C  MERGE MACHINE CELLS IMERG(K) AND JMERG(K)
C
1075  IMRGK=IMERG(K)
      JMRGK=JMERG(K)
      IF(IPMRG(K).NE.0) GO TO 1080
C
C  INCREMENT LINE NUMBER FOR DENDOGRAM
C
      LIN=LIN+1
      LINE(IMRGK)=LIN
      NLINE(LIN)=IMRGK
1080  IF(JPMRG(K).NE.0) GO TO 1090
      LIN=LIN+1
      LINE(JMRGK)=LIN
      NLINE(LIN)=JMRGK
C
C  FILL IN THE PRINT LINES WITH APPROPRIATE CHARACTERS
C
```

```
1090  LVLK=LEVEL(K)
      MARK=0
      ILINE=LINE(IMRGK)
1095  IF(LVLK-LAST(ILINE)-1) 1140,1100,1120
C
C  THE LAST SEGMENT OF A LINE BEFORE MERGE
C
1100  TRI(LVLK,ILINE)=DASHI
      LVLKP=LVLK
      IF(ILINP.LT.ILINE) ILINP=ILINE
      LAST(ILINE)=LVLK
      GO TO 1140
C
C  FILL THE INTERMEDIATE SEGMENTS
C
1120  IB=LAST(ILINE)+1
      IE=LVLK-1
      DO 1130 L=IB,IE
      TRI(L,ILINE)=DASH
1130  CONTINUE
      GO TO 1100
C
C  REPEAT THE PROCESS FOR JMERG(K)
C
1140  MARK=MARK+1
      IF(MARK.NE.1) GO TO 1145
      ILINE=LINE(JMRGK)
      GO TO 1095
C
C  DETERMINE THE LINES BETWEEN IMERG(K) AND JMERG(K)
C
1145  ILNK=LINE(IMRGK)
      JLNK=LINE(JMRGK)
      IF(ILNK.GT.JLNK) GO TO 1150
      IFRST=JLNK
      ILAST=ILNK
      GO TO 1160
1150  IFRST=ILNK
      ILAST=JLNK
1160  IF(IFRST.EQ.(ILAST+1)) GO TO 1175
C
C  FILL IN THE VERTICAL LINES
C
      IB=ILAST+1
      IE=IFRST-1
      DO 1170 L=IB,IE
      IF(TRI(LVLK,L).EQ.DASHI.OR.TRI(LVLK,L).EQ.DASH1) GO TO 1170
      TRI(LVLK,L)=BLNKI
      LAST(L)=LEVEL(K)
1170  CONTINUE
C
C  SET THE NEXT LINE NUMBER
C
1175  LINE(IMRGK)=(LINE(IMRGK)+LINE(JMRGK))/2
```

```
C
C  GO TO THE NEXT LEVEL
C
      KLAST=K
      MARK1=0
      K=NEXT(K)
C
C  MARK THE END OF A CLUSTER
C
      IF(LEVEL(K).LE.LVLK) GO TO 1176
      TRI(LVLKP,ILINP)=DASH1
      ILINP=0
C
C  CHECK FOR THE END OF DENDOGRAM
C
1176  IF(K.GE.NMCHN.OR.K.LT.1) GO TO 1500
C
C  AT THIS POINT THE MACHINE CELLS WITH MORE THAN ONE MEMBER JOIN TOGE-
C  THER
C
      IF(IPMRG(K).GT.0) GO TO 1180
C
C  SET ONE OF THE BRANCHES NEGATIVE (TO AVOID FURTHER CONSIDERATION)
C  AND GO DOWN THE OTHER BRANCH
C
      IPMRG(K)=-IPMRG(K)
      GO TO 1075
1180  IF(JPMRG(K).GT.0) GO TO 1190
      JPMRG(K)=-JPMRG(K)
      GO TO 1075
1190  IF(IPMRG(K).EQ.KLAST) GO TO 1200
      JPMRG(K)=-JPMRG(K)
      K=IPMRG(K)
      GO TO 1210
1200  IPMRG(K)=-IPMRG(K)
      K=JPMRG(K)
C
C  IF ALL BRANCHES ARE CONSTRUCTED PRINT THE DENDOGRAM
C
1210  IF(K.GT.NMCHN.OR.K.LT.1) GO TO 1500
C
C  CHECK FOR THE END OF DENDOGRAM
C
      IF(IPMRG(K)-JPMRG(K)) 1220,1075,1240
1220  IF(IPMRG(K).EQ.0) GO TO 1250
1230  K=IPMRG(K)
      GO TO 1210
1240  IF(JPMRG(K).EQ.0) GO TO 1230
1250  K=JPMRG(K)
      GO TO 1210
C
C  PRINT THE DENDOGRAM
C
1500  DO 1260 M=1,NMCHN
```

```
      WRITE(IO,1600) NLINE(M),(TRI(L,M),L=1,NSTEP)
1260  CONTINUE
      WRITE(IO,1265) (J,J=1,NSTEP),(STEPS(J),J=1,NSTEP)
1265  FORMAT(1X/11X,10I4/11X,10F4.1)
C
C  IDENTIFY MACHINE CELLS AND THEIR ASSOCIATED MACHINES
C
      CALL CELLS(ICELL,MCHIN,TRI,NLINE,ISTRT,IFINH,IDCEL,STEPS,
     .IWORD,NWORD,NPRTC,IMERG,JMERG,IPMRG,ITEMP)
      RETURN
1600  FORMAT(1X,I5,5X,10A4)
      END
C
C                    *********************************************
C                    **                                         **
C                    **            SUBROUTINE CELLS             **
C                    **                                         **
C                    *********************************************
C
C  THIS SUBROUTINE IDENTIFIES THE MACHINE CELLS FORMED AT DIFFERENT
C  THRESHOLD VALUES AND DETERMINES THE MACHINES IN EACH CELL. IT CALLS
C  SUBROUTINES DATA,NEXTC,INTRA,ACOST,INTRC,INIT,OTPUT,BOTLK,DUPLT,AND
C  UPDAT. IT ALSO CALLS FUNCTION IDCLL.
C
      SUBROUTINE CELLS(ICELL,MCHIN,TRI,NLINE,ISTRT,IFINH,IDCEL,STEPS,
     .IWORD,NWORD,NPRTC,IBTLK,JBTLK,ITRIP,JTEMP)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /LIST3/ NPRTS,NSTEP
      DIMENSION TRI(10,16),ICELL(15),MCHIN(2,16),NLINE(16),ISTRT(15),
     .IFINH(1),IDCEL(15),STEPS(10),IWORD(1),JTEMP(1),IBTLK(1),
     .JBTLK(1)
      DIMENSION NPRTS(44),IPRTC(32),NPRTC(44),ICELP(44),JFINH(-1:16),
     .NBTLK(16),MCHNB(15),MCHND(15),MCHNS(15),NTRIP(20),IPART(44)
      DATA DASH,DASHI,DASH1,BLANK,BLNKI/4H----,4H---I,4H---1,4H    ,4H
     . I/
C
C  GET INPUTS
C
      CALL DATA(IPART,NPRTJ,TIMES,K,L,MCHN,1)
C
C  INITIALIZE
C
      NBTS=NBITS
      NC=1
      LVL=1
      LASTN=1
      IFLAG=1
      ITEMP=0
C
C  ASSIGN MACHINES TO CELLS
C
      DO 1010 N=1,NMCHN
      MCHNB(N)=0
```

```
C
C  PREPARE LISTS OF MACHINES IN CELLS
C
      MCHIN(IFLAG,N)=NLINE(N)
      ICELL(NLINE(N))=NC
C
C  FIND CELL NUMBERS
C
      IF(ITEMP.EQ.0) ITEMP=IDCLL(TRI,N,LVL)
C
C  CHECK THE LAST MACHINE IN THE CURRENT CELL
C
      IF(TRI(LVL,N).EQ.DASH.OR.TRI(LVL,N).EQ.DASH1) CALL NEXTC(ISTRT,IFI
     .NH,IDCEL,ITEMP,LASTN,NSTEP,NC,N)
1010  CONTINUE
C
C  CHECK THE LAST MACHINE IN THE LAST CELL
C
      IF(TRI(LVL,NMCHN).EQ.DASHI) CALL NEXTC(ISTRT,IFINH,IDCEL,ITEMP,
     .LASTN,NSTEP,NC,N)
C
C  PRINT LABLES
C
1020  WRITE(IO,1022) LVL
1022  FORMAT(1X//////45X,'********** LEVEL',I3,2X,'**********'//)
      WRITE(IO,1240) LVL,STEPS(LVL)
1240  FORMAT(1X/////25X,'SIMILARITY LEVEL',I10/25X,'SIMILARITY COEFFICIE
     .NT',F8.4//25X,'CELLS',18X,'MACHINES')
C
C  NUMBER OF CELLS FOR THE NEXT SIMILARITY LEVEL
C
      NCELS=NC-1
C
C  INITIALIZE POINTERS
C
      IFINH(0)=0
      JFINH(0)=0
C
C  PRINT LISTS OF MACHINES IN CELLS
C
      DO 1025 N=1,NCELS
      IST=IFINH(N-1)+1
      IFN=IFINH(N)
      WRITE(IO,1242) N,(MCHIN(IFLAG,M),M=IST,IFN)
1242  FORMAT(1X,25X,I3,10X,16I4)
1025  CONTINUE
C
C  REPEAT THE PROCESS FOR LEVELS OTHER THAN ONE
C
      LVL=LVL+1
      NTRPW=0
C
C  IF ALL LEVELS DONE ,RETURN
C
```

```
      IF(LVL.GE.NSTEP) RETURN
C
C  INITIALIZE
C
      ITEMP=0
      L=0
      IPFLG=IFLAG
      IFLAG=3-IFLAG
      LASTN=1
      NCELS=NC-1
      NC=1
      L=0
      ITEMP=0
      MARK=0
      M=0
      TEMPS=0
      AC=0
C
C  FIND CELLS OF THE NEXT LEVEL
C
      DO 1040 N=1,NCELS
C
C  FIND ID NUNBER OF THE CELL
C
      ID=IDCEL(N)
C
C  FIND POINTERS TO THE FIRST AND LAST MACHINES IN EACH CELL
C
      IST=ISTRT(N)
      IFN=IFINH(N)
C
C  FIND THE LOCATION OF MACHINE VECTOR IST IN ARRAY IWORD
C
      IW=(MCHIN(IPFLG,IST)-1)*NWORD
      DO 1028 I=1,NWORD
      M=M+1
C
C  COPY MACHINE VECTOR IST IN IPRTC
C
      IPRTC(M)=IWORD(IW+I)
1028  CONTINUE
C     IK=0
      DO 1030 J=IST,IFN
C
C  GET A COPY OF CURRENT MACHINE LISTS
C
      L=L+1
C     IK=IK+1
      MEM=MCHIN(IPFLG,J)
      MCHIN(IFLAG,L)=MEM
C
C  RECORD CELL NUMBER OF MACHINES
C
      ICELL(MEM)=NC
```

```
C
C  DETERMINE INTRA-CELLULAR TRIPS
C
      IF(J.GT.IST) CALL INTRA(MCHIN,IFLAG,L,J,IST,NTRPW,NPRTS,IWORD,IPRT
     .C,NWORD,NBTS,M,IFN,IK)
1030  CONTINUE
C
C  FIND THE NEW CELL NUMBERS
C
      IF(ITEMP.EQ.0) ITEMP=IDCLL(TRI,ID,LVL)
C
C  CHECK THE LAST MACHINE OF THE CURRENT CELL
C
      IF(TRI(LVL,ID).EQ.DASH.OR.TRI(LVL,ID).EQ.DASH1) CALL NEXTC(
     .ISTRT,IFINH,IDCEL,ITEMP,LASTN,NSTEP,NC,L)
C
C  DETERMINE INTER-CELLULAR TRIPS
C
      IF(N.GT.1) CALL INTRC(IPRTC,N,NPRTS,NWORD,NBTS,NCELS,NTRPW)
1040  CONTINUE
C
C  CHECK THE LAST MACHINE IN THE LAST CELL
C
      IF(TRI(LVL,ID).EQ.DASHI) CALL NEXTC(ISTRT,IFINH,IDCEL,ITEMP,
     .LASTN,NSTEP,NC,L)
      NCELS=NC-1
C
C  DISCARD THE MACHINE CELLS ASSOCIATED WITH EXTREMEE THRESHOLD VALUES
C
      IF(NCELS.GT.5) GO TO 1020
C
C  INITIALIZE VARIABLES USED IN THE BOTTLENECK UNIT
C
      CALL INIT(MCHNB,MCHNS,MCHND,NCELS,NWORD,MCHN,NMCHN,LB,1,IWORD,
     .NPART,NBP,IM,MBTLK,MARK,NPRTC)
      WRITE(IO,1045)
1045  FORMAT(///22X,'THE ORIGINAL MCHINE-COMPONIENT CHART'///31X,'
     .  PARTS')
      CALL OTPUT(IWORD,NPRTC,MCHNB,MCHNS,MCHND,NLINE,IFINH,JFINH,JTEMP,
     .NCELS,NWORD,IM)
C
C  TAKE CARE OF BOTTLENECK MACHINES
C
      CALL BOTLK(IWORD,NLINE,IPRTC,NPRTC,IFINH,JFINH,MCHNB,MCHND,MCHNS,
     .IBTLK,JBTLK,ICELL,IPART,NMCHN,NPART,NWORD,NBITS,NCELS,NBP,MBTLK,IM
     .)
C
C  PRINT INTERMEDIATE RESULTS
C
      WRITE(IO,1046)
1046  FORMAT(///20X,'THE MCHINE-COMPONENET CHART BEFORE'/20X,'CONSIDE
     .RING BOTTLENECK MACHINES'///,35X,'PARTS')
      CALL OTPUT(IWORD,NPRTC,MCHNB,MCHNS,MCHND,NLINE,IFINH,JFINH,JTEMP,
     .NCELS,NWORD,IM)
```

```
1050  IF(MAXB.GT.LIMIT) THEN
C
C  DUPLICATION PROCESS
C
      CALL DUPLT(IWORD,IPRTC,IPART,NPRTC,NPRTS,NTRIP,IFINH,JFINH,ICELL,
     .NLINE,MCHNB,MCHND,MCHNS,IBTLK,JBTLK,NCELS,NMCHN,NPART,NWORD,NBITS,
     .NBP,LB,MBTLK,IM)
      GO TO 1050
      END IF
C
C  MARK THE BOTTLENECK MACHINES
C
      MARK=1010101010
      DO 1060 I=1,NMCHN
      K=ICELL(I)
      IF(MCHND(I).GT.0) JBTLK(K)=MARK
1060  CONTINUE
C
C  UPDATE CELL VECTORS
C
      DO 1070 I=1,NCELS
      CALL UPDAT(IWORD,IPRTC,NLINE,MCHNS,MCHND,IFINH,I,NWORD)
1070  CONTINUE
C
C  PRINT THE FINAL MACHINE-COMPONENT CHART
C
      WRITE(IO,1075)
1075  FORMAT(///22X,'THE FINAL MCHINE-COMPONENT CHART'///31X,'
     .PARTS')
      CALL OTPUT(IWORD,NPRTC,MCHNB,MCHNS,MCHND,NLINE,IFINH,JFINH,JTEMP,
     .NCELS,NWORD,IM)
      GO TO 1020
      END
C
C                   ********************************************
C                   **                                        **
C                   **           SUBROUTINE NEXTC             **
C                   **                                        **
C                   ********************************************
C
C  THIS SUBROUTINE KEEPS THE RECORD OF POINTERS TO THE FIRST AND LAST
C  MACHINES IN EACH CELL AND UPDATES THEM
C
      SUBROUTINE NEXTC(ISTRT,IFINH,IDCEL,ITEMP,LASTN,NSTEP,NC,N
     .)
      DIMENSION ISTRT(15),IFINH(15),IDCEL(15)
C
C  RECORD THE POINTER TO THE FIRST AND LAST MACHINES IN CELL N
C
      ISTRT(NC)=LASTN
      IFINH(NC)=N
C
C  RECORD THE CELL NUMBER OF CELL N
C
```

```
      IDCEL(NC)=ITEMP
C
C  UPDATE THE POINTERS
C
      ITEMP=0
      LASTN=N+1
      NC=NC+1
      RETURN
      END
C
C                   *********************************************
C                   **                                         **
C                   **            FUNCTION IDCLL               **
C                   **                                         **
C                   *********************************************
C
C  THIS FUNCTION FINDS CELL NUMBERS
C
      FUNCTION IDCLL(TRI,N,LVL)
      DIMENSION TRI(10,16)
      DATA BLANK,BLNKI/4H    ,4H   I/
      N1=N-1
C
C  NEXT LEVEL IN DENDOGRAM
C
      LVL1=LVL+1
C
C  NEXT LINE IN DENDOGRAM
C
1010  N1=N1+1
C
C  SKIP THE BLANK LINES
C
      IF(TRI(LVL1,N1).EQ.BLANK.OR.TRI(LVL1,N1).EQ.BLNKI) GO TO 1010
C
C  RECORD CELL NUMBER
C
      IDCLL=N1
      RETURN
      END
C
C                   *******************************************
C                   **                                       **
C                   **           SUBROUTINE INTRA            **
C                   **                                       **
C                   *******************************************
C
C  THIS SUBROUTINE DETERMINES THE INTRA-CELLULAR TRIPS
C
      SUBROUTINE INTRA(MCHIN,IFLAG,L,J,IST,NTRPW,NPRTS,IWORD,IPRTC,NWORD
     .,NBTS,M,IFN,IK)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      DIMENSION MCHIN(2,16),IWORD(1),IPRTC(1),NPRTS(1),ITEMP(8)
      IE=IFN-IST+1
```

```
      IF(IK.EQ.2) WRITE(IO,1000) (I,I=1,IE)
1000  FORMAT(1X//20X,'MACHINES'/10X,10I4)
C
C  FIND THE STARTING LOCATION OF MACHINE VECTOR L IN ARRAY IWORD
C
      NWI=(MCHIN(IFLAG,L)-1)*NWORD
      DO 1005 I=1,NWORD
C
C  ADD THE MACHINE VECTOR L TO CELL VECTOR N
C
      IW=IPRTC(M-NWORD+I)
      JW=IWORD(NWI+I)
      IPRTC(M-NWORD+I)=IOR(IW,JW)
1005  CONTINUE
C
C  FIND INTRA-CELLULAR TRIPS BETWEEN MACHINE L AND THE OTHER MACHINES
C
      LI=0
      J1=J-1
      DO 1010 I=IST,J1
      LI=LI+1
      ITEMP(IL)=0
      NWJ=(MCHIN(IFLAG,I)-1)*NWORD
C
C  FIND THE RESULT OF.AND. OPERATION ON MACHINE VECTORS
C
      NTRP=INDS(NWI,NWJ,NPRTS,IWORD,NWORD,NBTS,NPART)
      CALL SQUNC(MK,JPART,ICELL,NXP,NP,IM,NMOVE )
      ITEMP(IL)=NTRP*NMOVE
      NTRPW=NTRPW+NTRP
1010  CONTINUE
      WRITE(IO,1020) IK,(ITEMP(K),K=1,IK-1)
1020  FORMAT(8X,10I4)
      RETURN
      END
C
C                    ************************************************
C                    **                                            **
C                    **            SUBROUTINE INTRC                **
C                    **                                            **
C                    ************************************************
C
C  THIS SUBROUTINE DETERMINES THE INTER-CELLULAR TRIPS
C
      SUBROUTINE INTRC(IPRTC,N,NPRTS,NWORD,NBTS,NCELS,NTRPW)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      DIMENSION IPRTC(1),NPRTS(1),NTRPA(120)
C
C  FIND THE INTER-CELLULAR TRIPS BETWEEN CELL I AND THE OTHER CELLS
C
      M=(N-2)*(N-1)/2
      DO 1006 I=1,N
C
C  INITIALIZE
```

```
C
1006  NTRPA(I+M)=0
      NWJ=0
      II=N-1
      NWI=NWORD*II
      DO 1010 J=1,II
C
C  FIND THE RESULT OF .AND. OPERATION ON CELL VECTORS
C
      CALL SQUNC(MK,JPART,ICELL,NXP,NP,IM,NMOVE )
      NTRPA(J+M)=INDS(NWI,NWJ,NPRTS,IPRTC,NWORD,NBTS,NPART)*
     .NMOVE
      NWJ=NWJ+NWORD
1010  CONTINUE
C
C  PRINT FROM-TO CHART AND NUMBER OF INTRA-CELLULAR TRIPS
C
      IF(N.EQ.NCELS) THEN
      WRITE(IO,1020) NTRPW
1020  FORMAT(1X///25X,'THE TOTAL NUMBER OF TRIPS BETWEEN MACHINES WITHIN
     . CELLS=',I4)
      WRITE(IO,1030) (I,I=1,NCELS)
1030  FORMAT(1X///40X,'MACHINE CELLS'/22X,16I4/)
      M=1
      DO 1050 K=2,NCELS
      M1=M+K-2
      WRITE(IO,1040) K,(NTRPA(L),I=M,M1)
1040  FORMAT(1X/18X,16I4)
      M=M1+1
1050  CONTINUE
      END IF
      RETURN
      END
C
C                   ********************************************
C                   **                                        **
C                   **            FUNCTION INDS              **
C                   **                                        **
C                   ********************************************
C
C  THIS FUNCTION FIND THE RESULTS OF .AND. OPERATION ON TWO DATA VECTORS
C  (MCHINE OR CELL VECTORS)
C
      FUNCTION INDS(NWI,NWJ,NPRTS,IWRD,NWORD,NBTS,NPART)
      DIMENSION NPRTS(1),IWRD(32)
      INDS=0
C
C  FIND RESULTS OF IWORD.AND.JWORD
C
      DO 1020 I=1,NWORD
      IW=IWRD(NWI+I)
      JW=IWRD(NWJ+I)
      I1=IAND(IW,JW)
C
```

```
C  COUNT THE NUMBER OF BITS SCORING ONE IN THE RESULTIG DATA VECTOR
C  (AFTER .AND. OPERATION)
C
      DO 1010 N=1,NBTS+1
      M1=N-1-NBTS
      M2=-M1
      I2=ISHFT(I1,M1)
C
C  FIND PART NUMBER RELATING TO BIT M1 IN I1
C
      NPRT=(I-1)*NBTS+N
C
C  CALCULATE NUMBER OF TRIPS
C
      INDS=INDS+I2*NPRTS(NPRT)
C
C  LOOK FOR THE LAST PART
C
      IF(NPRT.GT.NPART) GO TO 1020
      I1=IBCLR(I1,M2)
1010  CONTINUE
1020  CONTINUE
      RETURN
      END
C
C                  **********************************************
C                  **                                          **
C                  **            SUBROUTINE BOTLK              **
C                  **                                          **
C                  **********************************************
C
C  THIS SUBROUTINE DEALS WITH BOTTLENECK MACHINES. IT CALLS SUROUTINES
C  ASSGN AND MODIF
C
      SUBROUTINE BOTLK(IWORD,NLINE,IPRTC,NPRTC,IFINH,JFINH,MCHNB,MCHND,
     .MCHNS,IBTLK,JBTLK,ICELL,IPART,NMCHN,NPART,NWORD,NBITS,NCELS,NBP,
     .MBTLK,IM)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION IWORD(1),NLINE(1),IPRTC(1),NPRTC(1),IFINH(1),
     .JFINH(-1:16),MCHNB(1),MCHND(1),MCHNS(1),IBTLK(1),JBTLK(1),NBTLK(16
     .),ICELP(44),ICELL(1),IPART(1)
C
C  INITIALIZE
C
      NBPN=0
      DO 1010 I=1,NMCHN
      IF(I.LE.(NCELS+2)) JFINH(I-2)=0
1010  NBTLK(I)=0
      DO 1020 I=1,NPART
      NP=I
      MNOP=0
C
C  FIND THE LOCATION OF THE STORAGE RELATING TO PART NP(IN ARRAY IWORD)
C
```

```
      NW=NP/NBITS
      IF(NW*NBITS.LT.NP) NW=NW+1
      M=NP-(NW-1)*NBITS
C
C  ASSIGN PARTS TO CELLS
C
      CALL ASIGN(IPRTC,IWORD,NLINE,IFINH,JFINH,IBTLK,NP,NW,M,MNOP,NWORD,
     .NBITS,MCHNS,MCHNB,MCHND,NCELS,IDCLP,NBPN,IM,ICELL,IPART)
1020  CONTINUE
      NBP=NBPN
C
C  DETERMINE THE BOTTLENECK MACHINE CREATING THE LARGEST NUMBER OF
C  INTER-CELLULAR MOVES
C
      MAXB=0
      DO 1040 I=1,NMCHN
      IF(MCHND(I).NE.0) GO TO 1040
      IF(NBTLK(I).GT.MAXB) THEN
      MAXB=NBTLK(I)
      MBTLK=I
      END IF
1040  CONTINUE
C
C  MARK MACHINES NOT DUPLICATED
C
      MCHND(MBTLK)=-1
C
C  SET THE POINTERS TO LAST PARTS IN DIFFERENT CELLS
C
      DO 1050 N=1,NCELS
      JFINH(N)=JFINH(N-1)+JFINH(N)
1050  CONTINUE
C
C  ASSIGN PART NP TO THE RELATED CELL
C
      DO 1060 I=1,NPART
      ID=ICELP(I)
      JFINH(ID-1)=JFINH(ID-1)+1
      NPRTC(JFINH(ID-1))=I
1060  CONTINUE
      RETURN
      END
C
C                   *********************************************
C                   **                                         **
C                   **            SUBROUTINE ASSGN             **
C                   **                                         **
C                   *********************************************
C
C  THIS SUBROUTINE DETERMINES NBTLK FOR EACH MACHINE AND NOP FOR EACH
C  PART. IT ALSO DETERMINES THE PARENT CELL OF EACH PART. IT CALLS
C  FUNCTION NOPRN AND SUBROUTINE MODIF.
C
      SUBROUTINE ASIGN(IPRTC,IWORD,NLINE,IFINH,JFINH,IBTLK,NP,NW,M,MNOP,
```

```
     .NWORD,NBITS,MCHNS,MCHNB,MCHND,NCELS,IDCLP,NBPN,IM,ICELL,IPART)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION IPRTC(1),IWORD(1),NLINE(1),IFINH(1),NBTLK(16),ICELP(44),
     .JFINH(-1:16),MCHNB(1),MCHNS(1),MCHND(1),IBTLK(1),ICELL(1),
     .IPART(1)
      NC=0
C
C  IDENTIFY PARTS VISITING EACH CELL
C
      DO 1020 N=1,NCELS
      NOP=0
C
C  FIND THE STARTING LOCATION OF CELL VECTOR N
C
      NWI=(N-1)*NWORD+NW
      IW=IPRTC(NWI)
C
C  CHECK THE VALUE OF BIT M IN CELL VECTOR N
C
      I2=ISHFT(IW,M)
      IF(I2.GE.0) GO TO 1020
C
C  IF BIT M SCORES ONE, INCREMENT THE NUMBER OF CELLS VISITED BY PART NP
C
      NC=NC+1
C
C  DETERMINE THE POINTERS TO THE FIRST AND LAST MACHINES IN CELL N
C
      IST=IFINH(N-1)+1
      IFN=IFINH(N)
      DO 1010 J=IST,IFN
      MCHN=NLINE(J)
      IF(MCHND(MCHN).LE.0) THEN
C
C  COUNT THE NUMBER OF MACHINES VISITED BY PART NP(REGULAR MACHINES)
C
      NOP=NOP+NOPRN(MCHN,IWORD,I2,NW,M,NWORD)
      ELSE
C
C  COUNT THE NUMBER OF MACHINES VISITED BY PART NP(DUPLICATED MACHINES)
C
      MCHN1=MCHND(MCHN)
      NOP=NOP+NOPRN(MCHN1,MCHNS,I2,NW,M,NWORD)
      END IF
C
C  COUNT THE NUMBER OF EXCEPTIONAL PARTS VISITING MACHINE MCHN
C
      IF(I2.LT.0) NBTLK(MCHN)=NBTLK(MCHN)+1
1010  CONTINUE
C
C  IDENTIFY THE MACHINES DUPLICATED IN CELL N
C
      MB=MCHNB(N)
1015  MCHN=INPAK(MB,IM)
```

```
      IF(MCHN.GT.0) THEN
C
C  COUNT THE NUMBER OF DULICATED MACHINES(IN CELL N) VISITED BY PART NP
C
      NOP=NOP+NOPRN(MCHN,IWORD,I2,NW,M,NWORD)
      GO TO 1015
      END IF
C
C  DETERMINE MNOP AND THE RELATED CELL FOR EACH PART
C
      IF(NOP.GT.MNOP) THEN
      IDCLP=N
      MNOP=NOP
      END IF
1020  CONTINUE
C
C  IDENTIFY EXCEPTIONAL PARTS
C
      IF(NC.GT.1) THEN
      NBPN=NBPN+1
      IBTLK(NBPN)=NP
      END IF
C
C  DETERMINE THE MACHINES DUPLICATED IN CELL IDCLP
C
      IF(ICELP(NP).GT.0.AND.ICELP(NP).NE.IDCLP) THEN
      MB=MCHNB(IDCLP)
1025  MCHN=INPAK(MB,IM)
C
C  DETERMINE THE CELLS(IN ADDITIN TO THE PARENT CELL) VISITED BY NP
C
      IF(MCHN.GT.0) THEN
      IF(ICELL(MCHN).EQ.IDCLP) GO TO 1025
      IST=NP
      IFN=NP
C
C  DETERMINE THE STARTING LOCATION OF MACHINE VECTOR MCHN IN ARRAY
C  MCHNS
C
      IBMS=MCHND(MCHN)
C
C  UPDATE THE MACHINE VECTOR OF THE DUPLICATED MACHINE
C
      CALL MODIF(IWORD,MCHNB,MCHND,MCHNS,IPART,IPRTC,NLINE,IFINH,NP,MCHN
     .,NWORD,NBITS,IM,IST,IFN,IBMS,2)
      GO TO 1025
      END IF
      END IF
C
C  ASSIGN PART NP TO CELL IDCLP
C
      ICELP(NP)=IDCLP
C
C  UPDATE THE POINTER TO THE LAST PART IN CELL IDCLP
```

```
C
      JFINH(IDCLP)=JFINH(IDCLP)+1
      IST=IFINH(IDCLP-1)+1
      IFN=IFINH(IDCLP)
C
C  ADJUST NBTLK,S FOR MACHINES IN IDCLP(NBTLK,S FOR THESE MACHINES
C  WERE PREVIOUSLY INCREMENTED BY ONE)
C
      DO 1030 J=IST,IFN
      MCHN=NLINE(J)
      NWI=(MCHN-1)*NWORD+NW
      IW=IWORD(NWI)
      I2=ISHFT(IW,M)
1030  IF(I2.LT.0) NBTLK(MCHN)=NBTLK(MCHN)-1
      RETURN
      END
C
C                    *******************************************
C                    **                                       **
C                    **          SUBROUTINE DUPLT             **
C                    **                                       **
C                    *******************************************
C
C  THIS SUBROUTINE DEALS WITH THE DUPLICATION PROCESS OF THE BOTTLENECK
C  MACHINES. IT CALLS SUBROUTINE DATA,INIT,MODIF,AND BOTLK.IT ALSO
C  CALLS FUNCTION NAVAL.
C
      SUBROUTINE DUPLT(IWORD,IPRTC,IPART,NPRTC,NPRTS,NTRIP,IFINH,JFINH,
     .ICELL,NLINE,MCHNB,MCHND,MCHNS,IBTLK,JBTLK,NCELS,NMCHN,NPART,NWORD,
     .NBITS,NBP,LB,MBTLK,IM)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION IWORD(1),IPRTC(1),IPART(1),NPRTC(1),NPRTS(1),NTRIP(1),
     :IFINH(1),JFINH(-1:16),ICELL(1),NLINE(1),MCHNB(1),MCHND(1),MCHNS(1)
     .,IBTLK(1),JBTLK(1),NBTLK(16),ICELP(44),ITRIP(20),NPRTJ(20)
      DIMENSION TIMES(20)
C
C  INITIALIZE
C
      L=0
      K=0
C
C  CHOOSE THE CRITICAL BOTTLENECK MACHINE(MBTLK)
C
      MCHN=MBTLK
C
C  FIND THE STARTING LOCATION OF MACHINE VECTOR MBTLK IN IWORD
C
      NWI=(MCHN-1)*NWORD
C
C  FIND POINTERS TO THE FIRST AND LAST PART IN CELL N
C
      DO 1035 N=1,NCELS
      IST=JFINH(N-2)+1
      IFN=JFINH(N-1)
```

```
      NTRIP(N)=0
      DO 1032 J=IST,IFN
C
C  FIND PARTS IN CELL N (NP)
C
      NP=NPRTC(J)
      NW=NP/NBITS
      IF(NW*NBITS.LT.NP) NW=NW+1
C
C  FIND THE BIT IN IWORD RELATING TO NP (BIT M)
C
      M=NP-(NW-1)*NBITS
      IW=IWORD(NWI+NW)
      I2=ISHFT(IW,M)
      IF(I2.LT.0) THEN
C
C  IF M=1 INCREMENT THE NUMBER OF INTER-CELLULAR TRIPS(FOR CELL N)
C  CREATED BY MBTLK
C
      NTRIP(N)=NTRIP(N)+1
C
C  RECORD THE PART NUMBER OF THE EXCEPIONAL PART
C
      L=L+1
      IPART(L)=NP
      END IF
1032  CONTINUE
      ITRIP(N)=NTRIP(N)
1035  CONTINUE
C
C  GET DATA FOR EXCEPTIONAL PARTS
C
      CALL DATA(IPART,NPRTJ,TIMES,K,L,MCHN,2)
      MARK=0
      DO 1040 N=1,NCELS
      IF(N.EQ.ICELL(MCHN).OR.NTRIP(N).EQ.0) GO TO 1040
C
C  DETERMINE THE MACHINE REQUIREMENTS OF THE EXCEPTIONAL PARTS IN
C  CELL N
C
      NAVL=NAVAL(IWORD,NPRTC,NPRTS,JFINH,NPRTJ,N,MCHN,NTRIP,IPART,TIMES,
     .IST,IFN)
      IF(NAVL.GT.0) THEN
C
C  RECORD THE CELLS IN WHICH MACHINE MCHN IS DUPLICATED
C
      MCHNB(N)=IPACK(MCHNB(N),MCHN,IM)
C
C  INITIALIZE THE VARIABLES RELATED TO DUPLICATED MACHINES
C
      IF(MARK.EQ.0) CALL INIT(MCHNB,MCHNS,MCHND,NCELS,NWORD,MCHN,NMCHN,
     .LB,2,IWORD,NPART,NBP,IM,MBTLK,MARK,NPRTC)
      IBMS=IB
C
```

```
C  MODIFY THE MACHINE-COMPONENT CHART
C
      CALL MODIF(IWORD,MCHNB,MCHND,MCHNS,IPART,IPRTC,NLINE,IFINH,N,MCHN,
     .NWORD,NBITS,IM,IST,IFN,IBMS,1)
      END IF
1040  CONTINUE
C
C  DETERMINE THE NEXT CRITICAL BOTTLENECK MCHINE-MBTLK
C
1050  CALL BOTLK(IWORD,NLINE,IPRTC,NPRTC,IFINH,JFINH,MCHNB,MCHND,
     .MCHNS,IBTLK,JBTLK,ICELL,IPART,NMCHN,NPART,NWORD,NBITS,NCELS,NBP,
     .MBTLK,IM)
      RETURN
      END
C
C                   *********************************************
C                   **                                         **
C                   **            FUNCTION NOPRN               **
C                   **                                         **
C                   *********************************************
C
C  THIS FUNCTION DETERMINES THE NUMBER OF MACHINES VISITED BY NP
C  IN EACH CELL
C
      FUNCTION NOPRN(MCHN,JWORD,I2,NW,M,NWORD)
      DIMENSION JWORD(1)
C
C  FIND THE STARTING LOCATION OF MACHINE VECTOR MCHN IN IWORD
C
      NWI=(MCHN-1)*NWORD+NW
      IW=JWORD(NWI)
      I2=ISHFT(IW,M)
      NOPRN=0
      IF(I2.LT.0) NOPRN=1
      RETURN
      END
C
C                   ********************************************
C                   **                                        **
C                   **            FUNCTION NAVAL              **
C                   **                                        **
C                   ********************************************
C
C  THIS FUNCTION DETERMINES THE MACHINE REQUIREMENTS OF THE EXCEPTIONAL
C  PARTS IN EACH CELL. IT ALSO DETERMINES THE MATERIAL HANDLING COST
C  DUE TO THE INTER-CELLULAR TRIPS CREATED BY A BOTTLENECK MACHINE.
C  FUNCTION IVALT IS CALLED BY NAVAL
C
      FUNCTION NAVAL(IWORD,NPRTC,NPRTS,JFINH,NPRTJ,N,MCHN,NTRIP,IPART,
     .TIMES,IST,IFN)
      COMMON /BLOK1/ PRICE,COSTI,ULIFE,SVALU,R,OCOST,UFCTR,H,ADIST
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      DIMENSION IWORD(1),NPRTC(1),NPRTS(1),JFINH(-1:16),NTRIP(1),
     .IPART(1),ITRIP(1),NPRTJ(20)
```

```
      DIMENSION TIMES(1)
C
C  INITIALIZE
C
      WHOUR=5.0*H
      TIME=0
      PRCNT=.5
      ACAST=1
      IST=1
      IFN=0
C
C  DETERMINE THE POINTERS TO THE FIRST AND LAST EXCEPTIONAL PART
C  IN CELL N
C
      DO 1010 I=1,N
      IF(I.GT.1) IST=IST+NTRIP(I-1)
      IFN=IFN+NTRIP(I)
1010  CONTINUE
      COST=0
      TIME=0
C
C  DETERMINE THE MACHINE REQUIREMENT OF EXCEPTIONAL PARTS IN CELL N
C
      DO 1020 J=IST,IFN
C
C  FIND PART NUMBER OF THE JTH EXCEPTIONAL PART(IN CELL N)
C
      NP=IPART(J)
C
C  FIND THE TOTAL PROCESSING TIME OF THE EXCEPTIONAL PARTS ON MACHINE
C  MCHN
C
      TIME=TIME+FLOAT(NPRTJ(J))*TIMES(J)
C
C  FIND THE MATERIAL HANDLING COST DUE TO MACHINE MCHN
C
      COST=COST+FLOAT(NPRTJ(J))*ACAST
      WRITE(IO,1015) MCHN,N,IST,IFN,J,NP,NPRTJ(J),TIMES(J),TIME
1015  FORMAT(1X,7I4,2F8.4)
1020  CONTINUE
C
C  FIND THE REQUIRED NUMBER OF MACHINES
C
      AVL=TIME/(WHOUR*UFCTR)
      NAVAL=AVL
C
C  FIND THE FRACTION OF MACHINE REQUIRED
C
      DEV=AVL-FLOAT(NAVAL)
C
C  IF THE FRACTION IS LARGE ENOUGH TO JUSTIFY THE ASSIGNMENT OF
C  ONE MACHINE ,INCREMENT NAVAL
C
      IF(DEV.GT.PRCNT) THEN
```

```
      NAVAL=NAVAL+1
      RETURN
      END IF
C
C  IF THE FRACTION IS TOO SMALL ,NO ADDITIONAL MACHINE IS REQUIRED
C
      IF(DEV.LT.PRCNT/10.) RETURN
C
C  DETERMINE THE PORTION OF COST RELATED TO THE FRACTION
C
      COST=(COST/AVL)*DEV
C
C  PERFORM COST-BENEFIT ANALYSIS
C
      NAVAL=IVALT(COST)
      RETURN
      END
C
C                 ********************************************
C                 **                                        **
C                 **             FUNCTION IVALT             **
C                 **                                        **
C                 ********************************************
C
C  THIS FUNCTION EVALUATES THE COST INCURRED BY AND THE BENEFIT
C  RESULTED FROM THE DUPLICATION OF A BOTTLENECK MACHINE
C
      FUNCTION IVALT(COST)
      COMMON /BLOK1/ PRICE,COSTI,ULIFE,SVALU,R,OCOST,UFCTR,H,ADIST
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      WWEEK=52
C
C  FIND THE ANNUAL COST INCURRED DUE TO PURCHASE OF A NEW MACHINE
C
      AC=(PRICE-SVALU)*(R*(1.+R)**ULIFE)/((1.+R)**ULIFE-1.)+SVALU*R+
     .OCOST
C
C  FIND WEEKLY COST
C
      WCOST=AC/WWEEK
      IVALT=0.0
C
C  BUY A NEW MACHINE IF WCOST<COST
C
      IF(WCOST.LT.COST) IVALT=1
      WRITE(IO,1010) COST,WCOST,IVALT
1010  FORMAT(1X,2F10.2,I5)
      RETURN
      END
C
C                 *********************************************
C                 **                                         **
C                 **             FUNCTION IPACK              **
C                 **                                         **
```

```
C                   ***********************************************
C
C  THIS FUNCTION PACKS SEVERAL CELL NUMBERS( RELATING TO DUPLICATED
C  MACHINES ) INTO A SINGLE COMPUTER WORD
C
      FUNCTION IPACK(M1,M2,IM)
      MC=0
C
C  FIND THE LOCATION OF BITS(IN NUMBR) TO BE USED FOR STORING CELL
C  NUMBER
C
1010  IF(M1.GE.10**MC) THEN
      MC=MC+IM
      GO TO 1010
      END IF
C
C  STORE THE CELL NUMBER
C
      IPACK=M1+M2*10**MC
      RETURN
      END
C
C                   **********************************************
C                   **                                          **
C                   **            FUNCTION INPAK                **
C                   **                                          **
C                   **********************************************
C
C  THIS FUNCTION UNPACKS THE CELL NUMBERS PACKED BY IPACK
C
      FUNCTION INPAK(M1,IM)
      IDC=M1/10**IM
      INPAK=M1-IDC*10**IM
      M1=IDC
      RETURN
      END
C
C                   *********************************************
C                   **                                         **
C                   **            SUBROUTINE DATA              **
C                   **                                         **
C                   *********************************************
C
C  THIS SUBROUTINE READS THE DATA RELATING TO EXCEPTIONAL PARTS . IT
C  ALSO READS THE AVERAGE TRAVELLING DISTANCE AND THE COST DATA OF THE
C  BOTTLENECK MACHINES.
C  INPUT FORMAT:F5.2/24F3.1/24I3/9F8.2/
C
      SUBROUTINE DATA(IPART,NPRTJ,TIMES,K,L,MCHN,ID)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      COMMON /BLOK1/ PRICE,COSTI,ULIFE,SVALU,R,OCOST,UFCTR,H,ADIST
      DIMENSION IPART(1),NBTLK(16),ICELP(44),NPRTJ(20)
      DIMENSION TIMES(1)
```

```
      IF(ID.EQ.1) THEN
C
C  READ THE AVERAGE TRAVELLING DISTANCE FOR AN INTER-CELLULAR TRIP
C
      READ(UNIT=IN,FMT=1005) ADIST
1005  FORMAT(F5.2)
      RETURN
      END IF
C
C  READ THE PROCSSING TIMES AND PRODUCTION REQUIREMENT OF THE EXCEP-
C  TIONAL PARTS
C
      READ(UNIT=IN,FMT=1010) (TIMES(I),I=1,L)
      READ(UNIT=IN,FMT=1020) (NPRTJ(I),I=1,L)
1010  FORMAT(20F3.1)
1020  FORMAT(20I3)
C
C  READ COST DATA OF THE BOTTLENECK MACHINES
C
      READ(UNIT=IN,FMT=1030) PRICE,COSTI,ULIFE,SVALU,R,OCOST,UFCTR,H,
     .ADIST
1030  FORMAT(9F8.2)
      PRICE=PRICE+COSTI
      RETURN
      END
C
C                  *********************************************
C                  **                                         **
C                  **            SUBROUTINE INIT              **
C                  **                                         **
C                  *********************************************
C
C  THIS SUBROUTINE INITIALIZES THE VARIABLES OF THE BOTTLENECK UNIT
C
      SUBROUTINE INIT(MCHNB,MCHNS,MCHND,NCELS,NWORD,MCHN,NMCHN,LB,IND,
     .IWORD,NPART,NBP,IM,MBTLK,MARK,NPRTC)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION MCHNB(1),MCHNS(1),MCHND(1),IWORD(1),NBTLK(16),ICELP(44),
     .NPRTC(1)
      GO TO (1010,1040,1060) IND
1010  NBP=NPART
C
C  INITIALIZE THE VARIABLES RELATING TO DUPLICATED MACHINES
C
      LIMIT=2
      IM=2
      LB=0
      IB=0
      MBTLK=2
      DO 1030 N=1,NPART
      NPRTC(N)=N
      IF(N.LE.NCELS) MCHNB(N)=0
      IF(N.LE.NMCHN) MCHND(N)=0
1030  ICELP(N)=0
```

```
      RETURN
C
C  INITIALIZE THE MACHINE VECTOR OF THE DUPLICATED MACHINE
C
1040  NW=(MCHN-1)*NWORD
      MARK=1
      IB=IB+1
      MCHND(MCHN)=IB
      LB=(IB-1)*NWORD
      DO 1050 J=1,NWORD
      MCHNS(LB+J)=IWORD(NW+J)
1050  CONTINUE
1060  RETURN
      END
C
C             ********************************************
C             **                                        **
C             **            SUBROUTINE MODIF            **
C             **                                        **
C             ********************************************
C
C  THIS SUBROUTINE MODIFIES THE MACHINE VECTORS OF THE DUPLICATED
C  MACHINES TO REFLECT THE RELATED CHANGES
C
      SUBROUTINE MODIF(IWORD,MCHNB,MCHND,MCHNS,IPART,IPRTC,NLINE,IFINH,
     .ID,MCHN,NWORD,NBITS,IM,IST,IFN,IBMS,IND)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION MCHNB(1),MCHNS(1),IPART(1),IPRTC(1),IWORD(1),NLINE(1)
     .,IFINH(1),MCHND(1),NBTLK(16),ICELP(44)
C
C  FIND THE STARTING LOCATION OF THE MACHINE VECTOR RELATED TO THE
C  DUPLICATED MACHINE
C
      NWI=(IBMS-1)*NWORD
C
C  FIND THE EXCEPIONAL PARTS VISITING THE DUPLICATED MACHINE
C
      NP=ID
      DO 1010 J=IST,IFN
      IF(IND.EQ.1) NP=IPART(J)
C
C  FIND THE BIT RELATING TO PART NP(BIT M) (IN THE MACHINE VECTOR RELAT
C  ING TO THE DUPLICATED MACHINE)
C
      NW=NP/NBITS
      IF(NW*NBITS.LT.NP) NW=NW+1
      M=NP-(NW-1)*NBITS
      M=NBITS-M
      IW=MCHNS(NWI+NW)
C
C  SET BIT M EQUAL TO ZERO
C
      MCHNS(NWI+NW)=IBCLR(IW,M)
1010  CONTINUE
```

```
      RETURN
      END
C
C                 *******************************************
C                 **                                       **
C                 **            SUBROUTINE UPDAT           **
C                 **                                       **
C                 *******************************************
C
C  THIS SUBROUTINE UPDATES THE DIFFERENT CELL VECTORS AFTER EACH
C  DUPLICATION
C
      SUBROUTINE UPDAT(IWORD,IPRTC,NLINE,MCHNS,MCHND,IFINH,IDC,NWORD)
      COMMON /IBLOK/ NBTLK,ICELP,LIMIT,MAXB,IB
      DIMENSION IWORD(1),IPRTC(1),NLINE(1),MCHNS(1),IFINH(1),
     .MCHND(1)
C
C  FIND THE STARTING LOCATION OF THE CELL VECTOR IDC IN ARRAYS
C  IWORD OR MCHNS
C
      NWI=(IDC-1)*NWORD
      DO 1020 I=1,NWORD
C
C  INITIALIZE THE CELL VECTOR
C
1020  IPRTC(NWI+I)=0
C
C  FIND THE POINTERS TO THE FIRST AND LAST MACHINES IN CELL IDC
C
      IST=IFINH(IDC-1)+1
      IFN=IFINH(IDC)
      DO 1040 J=IST,IFN
C
C  FIND THE MACHINES IN CELL IDC
C
      MCHN=NLINE(J)
C
C  CHECK FOR DUPLICATED MACHINES
C
      IF(MCHND(MCHN).GT.0) MCHN=MCHND(MCHN)
      NWJ=(MCHN-1)*NWORD
C
C  FIND THE NEW CELL VECTOR BY ORING THE RELATED MACHINE VECTORS
C
      DO 1030 I=1,NWORD
      IW=IWORD(NWJ+I)
      IF(MCHND(NLINE(J)).GT.0) IW=MCHNS(NWJ+I)
      JW=IPRTC(NWI+I)
      IPRTC(NWI+I)=IOR(IW,JW)
1030  CONTINUE
1040  CONTINUE
      RETURN
      END
C
```

```
C                    **********************************************
C                    **                                          **
C                    **               SUBROUTINE SEQNC           **
C                    **                                          **
C                    **********************************************
C
C  THIS SUBROUTINE DETERMINES THE NUMBER OF TIMES EACH PART MOVES
C  BETWEEN TWO CELLS OR TWO MACHINES-NMOVE
C
      SUBROUTINE SQUNC(MK,JPART,ICELL,NXP,NP,IM,NMOVE )
      COMMON /BLOK1/ PRICE,COSTI,ULIFE,SVALU,R,OCOST,UFCTR,H,ADIST
      DIMENSION MK(1),JPART(1),ICELL(1)
      NMOVE=0
C
C  FIND THE LOCATION OF THE EXCEPTIONAL PART NP IN ARRAY JPART
C
      DO 1010 I=1,NXP
      IF(JPART(I).EQ.NP) GOTO 1020
1010  CONTINUE
C
C  FIND THE SEQUENCE NUMBER FOR NP
C
1020  MKI=MK(I)
      ICONT=0
1030  IF(MKI.LE.0) RETURN
C
C  FIND THE CELL NUMBERS VISITED BY NP.
C
      MKII=MKI/IM
      IDI=MKI-MKII*IM
      MKI=MKII
      ICONT=ICONT+1
C
C  RECORD THE FIRST CELL NUMBER
C
      IF(ICONT.LT.2) THEN
      IDJ=IDI
      GO TO 1030
      END IF
C
C  IF PART NP VISTS MACHINES IN TWO DIFFERENT CELLS, INCREMENT
C  NMOVE FOR IT
C
      IF(ICELL(IDI).NE.ICELL(IDJ)) NMOVE=NMOVE+1
      IDJ=IDI
      GO TO 1030
      END
C
C                   ***********************************************
C                   **                                           **
C                   **               SUBROUTINE OTPUT            **
C                   **                                           **
C                   ***********************************************
C
```

```
C  THIS SUBROUTINE ARRANGES THE MACHINE-COMPONENT CHARTS IN APPROPRIATE
C  FORMATS AND PRINTS THEM OUT
C
      SUBROUTINE OTPUT(IWORD,NPRTC,MCHNB,MCHNS,MCHND,NLINE,IFINH,JFINH,
     .ITEMP,NCELS,NWORD,IM)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      DIMENSION IWORD(1),NPRTC(1),MCHNB(1),MCHNS(1),MCHND(1),NLINE(1),
     .IFINH(1),JFINH(-1:16),ITEMP(1)
C
C  PRINT THE PART NUMBERS FOR PARTS IN A MACHINE-COMPONENT CHART
C
      WRITE(IO,1010) (NPRTC(I),I=1,NPART)
1010  FORMAT(3X,43I2)
      DO 1030 N=1,NCELS
C
C  IDENTIFY THE MACHINES IN EACH CELL
C
      IST=IFINH(N-1)+1
      IFN=IFINH(N)
      DO 1015 J=IST,IFN
      MCHN=NLINE(J)
      MCHN1=MCHN
      IF(MCHND(MCHN).LE.0) THEN
C
C  ARRANGE THE MACHINE VECTOR MCHN(FOR REGULAR MACHINES)
C
      CALL BUFER(IWORD,NPRTC,ITEMP,MCHN,MCHN1,NWORD,NPART,NBITS)
      ELSE
      MCHN1=MCHND(MCHN)
C
C  ARRANGE THE MACHINE VECTOR MCHN (FOR DUPLICATED MACHINES)
C
      CALL BUFER(MCHNS,NPRTC,ITEMP,MCHN,MCHN1,NWORD)
      END IF
1015  CONTINUE
C
C  DETERMINE THE MACHINES DUPLICATED IN CELL N
C
      MB=MCHNB(N)
1020  MCHN=INPAK(MB,IM)
      IF(MCHN.GT.0) THEN
C
C  INITIALIZE ARRAY ITEMP
C
      DO 1023 I=1,NPART
1023  ITEMP(I)=0
      NWI=(MCHN-1)*NWORD
C
C  DETERMINE PARTS HAVING OPERATION ON THE DUPLICATED MACHINE AND
C  SET THE RELATED BIT IN ITEMP EQUAL TO ONE
C
      IST=JFINH(N-2)+1
      IFN=JFINH(N-1)
      DO 1025 J=IST,IFN
```

```
C
C  FIND THE PART NUMBER OF THE JTH PART(PART NP)
C
      NP=NPRTC(J)
      NW=NP/NBITS
      IF(NW*NBITS.LT.NP) NW=NW+1
      IW=IWORD(NWI+NW)
C
C  FIND THE BIT RELATING TO PART NP IN IWORD
C
      M=NP-NBITS*(NW-1)
      I2=ISHFT(IW,M)
      ITEMP(J)=0
C
C  FOR PART HAVING OPERATION ON MCHN SET M=1
C
      IF(I2.LT.0) ITEMP(J)=1
1025  CONTINUE
C
C  PRINT THE MACHINE VECTOR RELATED TO THE DUPLICATED MACHINE
C
      WRITE(IO,1027) MCHN,(ITEMP(I),I=1,NPART)
1027  FORMAT(1X,44I2)
      GO TO 1020
      END IF
1030  CONTINUE
      RETURN
      END
C
C                    ********************************************
C                    **                                        **
C                    **            SUBROUTINE BUFER            **
C                    **                                        **
C                    ********************************************
C
C  THIS SUBROUTINE ARRANGES THE DATA IN THE MACHINE VECTORS IN APPROPR-
C  IATE FORMAT AND PRINTS THEM OUT
C
      SUBROUTINE BUFER(JWORD,NPRTC,ITEMP,MCHN,MCHN1,NWORD)
      COMMON /LIST1/ IN,IO,NBITS,NMCHN,NPART
      DIMENSION JWORD(1),NPRTC(1),ITEMP(1)
C
C  FIND THE STARTING LOCATION OF MACHINE VECTOR MCHN1 (IN IWORD OR
C  MCHNS)
C
      NWI=(MCHN1-1)*NWORD
      DO 1020 I=1,NPART
C
C  FIND THE PART NUMBER OF THE ITH PART IN THE MACHINE-COMPONENT CHART
C
      NP=NPRTC(I)
C
C  FIND THE BIT IN IW RELATING TO NP(BIT M)
C
```

```
      NW=NP/NBITS
      IF(NW*NBITS.LT.NP) NW=NW+1
      M=NP-NBITS*(NW-1)
      IW=JWORD(NWI+NW)
      I2=ISHFT(IW,M)
C
C  SET ITEMP EQUAL TO THE VALUE OF BIT M
C
      ITEMP(I)=0
      IF(I2.LT.0) ITEMP(I)=1
1020  CONTINUE
C
C  PRINT THE MACHINE VECTOR
C
      WRITE(IO,1030) MCHN,(ITEMP(I),I=1,NPART)
1030  FORMAT(1X,44I2)
      RETURN
      END
```

VITA

Hamid Seifoddini

Candidate for the Degree of

Doctor of Philosophy

Thesis: COST BASED MACHINE-COMPONENT GROUPING MODEL IN GROUP TECHNOLOGY

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Koohbanan, Kerman, Iran, in October 1949, the son of Abdolhossein and Dorbebe Seifoddini.

Education: Graduated from Pahlavi High School, Kerman, Iran, 1969; received Bachelor of Science in Industrial Engineering from Ary-Mehr University of Technology, Tehran, Iran in 1974; received Master of Science in Industrial Engineering and Management from Oklahoma State University, Stillwater, Oklahoma in 1977; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in December 1984.

Professional Experience: Director of the Mechanical Workshop, Pabedana Coal Mines, 1974-1976; graduate assistant, Oklahoma State University, 1977-1979; instructor, Kerman University, Kerman, Iran, 1979-1980; industrial engineer, Arak Machine Manufacturing Corporation, Arak, Iran, 1980-1981; graduate assistant, Oklahoma State University, 1981-1982; instructor, Langston University, Langston, Oklahoma, 1983-1984.

Professional Organizations: American Institute of Industrial Engineering.