1986

Cell formation in flexible manufacturing systems.

Gajanana. Nadoli
University of Windsor

CELL FORMATION IN FLEXIBLE MANUFACTURING SYSTEMS

by

Gajanana Nadoli

A Thesis
submitted to the
Faculty of Graduate Studies and Research
through the Department of
Industrial Engineering in Partial Fulfillment
of the requirements for the Degree
of Master of Applied Science at
the University of Windsor

Windsor, Ontario, Canada

ISBN 0-315-31996-8

I hereby declare that I am the sole author of this thesis. I authorize the University of Windsor to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Gajanana Nadoli

I further authorize the University of Windsor to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Gajanana Nadoli

The University of Windsor requires the signature of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

In recent years the concept of flexible manufacturing systems (FMS) has emerged as a viable answer to the problems of low volume, medium variety production. The technological sophistication and correspondingly high investment in these systems necessitate sufficient planning effort both in the implementation and the operation stages. This research deals with the initial specification decisions in the pre-production planning stage. The cellular configuration of FMS is considered, in which a group of machines is dedicated to the manufacture of a particular family of parts. Two of the problems in cell formation viz., part family formation and machine group allocation are formulated. A fractional programming model defined on zero-one integer variables has been proposed for the part family formation. The parts are grouped based on their processing similarity. The machine group allocation problem is formulated as a zero-one integer program, to maximize the routing diversity available for the parts in different families. The availability of alternative routings has been considered in cell formation. The application of the formulations has been illustrated through a number of examples using realistic data.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude towards Dr. S.P. Dutta and Dr. R.S. Lashkari for their guidance and support during the course of this research. I would like to thank Dr. Y. Aneja for sparing his time to give useful suggestions. A special note of thanks to Jacquie Mummery and Tom Williams for their help from time to time. I would also like to thank the consultants of the University Computer Centre for their help.

LIST OF FIGURES

Figure | Page

LIST OF TABLES

TABLE OF CONTENTS

Chapter | Page

Chapter I

INTRODUCTION

In recent years the concept of Flexible Manufacturing systems (FMS) has emerged as a viable answer to the problems of low volume, medium variety production. These systems offer automated and flexible operation coupled with the optimum exploitation of resourses. It is acknowledged that an integrated approach to parts manufacture from design conceptualization to operation stage is the pre-condition for the success of such systems.

The technological sophistication and the correspondingly high investment in these systems necessitate sufficient planning efforts both in the implementation and operation stages.

The efficient system design to facilitate the gradual implementation is very important. It can be achieved by conceiving the FMS to be made up of different groups of machines. In Group Technology terms these groups are known as cells. Chapter 2 briefly explains the flexible manufacturing systems, the types of arrangements of the manufacturing set ups, the advantages of cellular arrangement and the system configuration under consideration.

The objective of this research is to model two of the problems related to the cell formation in FMS.

i) The part family formation

ii) The machine group allocation.

A review of the previous research is given in Chapter 3.

The formulations of the above two problems are explained in Chapters 4 and 5 respectively. A fractional programming model for minimizing the processing dissimilarities between different part types has been proposed for the part family formation. A solution procedure is developed for this model taking into consideration the nature of the objective function. The procedure suitably adopts a general principle of search for finding the optimal solution. The infeasibility in allocation caused by restricting each of the machines to only one family (unique allocation) has been resolved. A simple mathematical model identifies the machines causing the infeasibility and the unique allocation constraint is relaxed for these machines.

Realistic data representing typical part and machine varieties have been considered in solving a number of problems to illustrate the formulations. The results are explained in Chapter 6. A summary of the research findings has been presented in Chapter 7.

Chapter II

SYSTEM DESCRIPTION AND OBJECTIVES

2.1 Flexible manufacturing systems

2.1.1 Definitions

An FMS is an automated, batch manufacturing system consisting of a set of numerically controlled machine tools with automatic tool changing capabilities. A computer controlled material handling system transports the parts from machine to machine.

These systems have been given a variety of names - Computerized Manufacturing Systems (CMS) and Variable Manufacturing Systems (VMS), for example, and have in fact been designed in a variety of configurations.

The cellular configuration of FMS is considered in this research. The definitions of the terminology [22], with reference to this type of configuration are given below.

Flexible Manufacturing Module (FMM): An FMM is defined as a Numerically Controlled Machine augmented by a part buffer, a tool changer, a pallet changer etc. An FMM will be referred to as a machine throughout this report.

Flexible Manufacturing Cell (FMC): An FMC consists of several machines, capable of producing a range of parts. Each of these FMCs are organized as independent facility set-ups. The term cell has been borrowed from cellular manufacturing in the conventional systems. The FMCs are referred to as cells in the discussions to follow. An FMS can be considered to be consisting of cells. Many a times individual cells themselves are considered as systems, indicating the independent nature of these cells.

2.1.2 Cellular Configuration of FMS

There are various approaches to the arrangement of machines in a manufacturing system. In all the cases, it is necessary to conceive the system as a whole from design to installation.

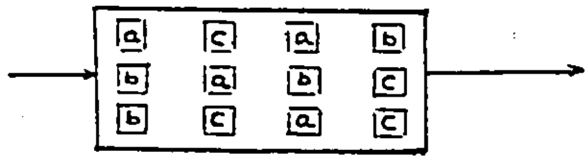
The typical arrangements of the machines in the manufacturing systems are (Fig. 1):

i) Random: A number of machines are arranged in a rectangular shop. The disadvantage of this lay-out is that with larger number of machines, transfer paths are complicated and are likely to be longer than necessary.

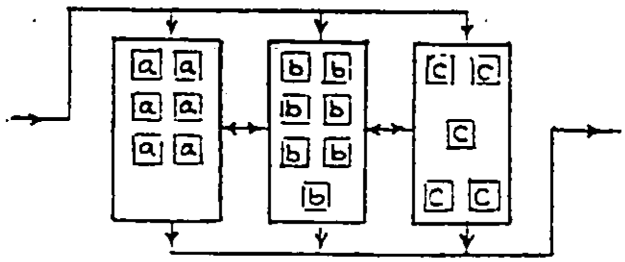
ii) Functional: The machines are arranged according to function, such as turning, milling, boring and grinding, so

Figure 1. TYPICAL ARRANGEMENTS OF MACHINES IN MANUFACTURING SYSTEMS

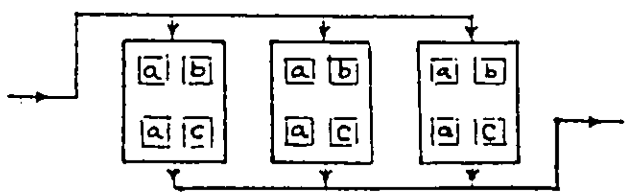
Random Layout



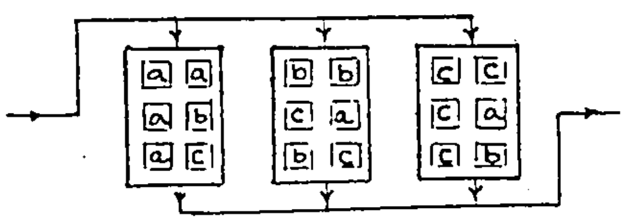
Functional Layout



Modular Layout



Cellular Layout



that the workpieces flow through the shop from one section to another. The workpieces have to be moved many times between the sections, and material handling paths in this type of arrangement may be excessively long.

iii) Modular: Here identical modules perform similar processes in parellel. This layout is likely to result in some redundant capacity, but can be an alternative to the functional layout, while redundancy may make it easy to cope with critical jobs or unexpected problems.

iv) Cellular: In this arrangement each cell is dedicated to a certain group of parts. It is an extension of the Group Technology (GT) concept. The cellular system is likely to give the best match of machining capability to the processing of various workpieces [18].

Group technology is a manufacturing philosophy that seeks to rationalize small and medium sized batch production by capitalizing on the similarity between the parts. GT is applied with respect to two aspects of part characteristics viz., geometric features and processing requirements.

The geometric feature based grouping has been mainly a part of design standardization effort for the various shapes of the parts. The concept has recently been considered in the computer aided process planning area, where an attempt to relate the processing steps to the geometric features is made to develop computerized systems for generating the process plans [9].

The grouping of parts with respect to the processing requirements forms the basis of cellular arrangement of the machines. A manufacturing cell is designed to produce the parts with similar machining requirements. Due to similarity of the parts, change over of processing from one part type to another on the machines causes minimal disruptions in terms of tooling requirements. Section 2.2.1 describes this issue. The cellular arrangement is an attempt to achieve the advantages of mass production in small batch production. Several conventional systems have been installed based on this principle [17].

2.1.3 Operation of the system

The FMSs can be viewed as highly automated job shops. A typical sequence of events involved in processing of a part in an FMS is as follows:

When a part is scheduled for an operation on one of the available alternative machines, the part is fixtured on a pallet and transported to the machine. The machine on which this part is to be processed receives the necessary part programs. If certain tools are not available on the machine, the handling system transports those tools also to the machine [18]. Once the machining for that operation is finished, the part is moved for its next operation.

Since different parts are in production simultaneously, conflicts in the requirements arise. Among other things, the automated control has to consider the important issues of scheduling of parts, queues for the machines and machine break downs.

The automated operation requires the proper operation logic to be programmed into the system prior to the start of production. The precise anticipation of all the operational exigencies is necessary in such an operation mode.

2.1.4 Advantages of Cellular Configuration

Dividing the system into smaller sub-systems (cells) is essential due to the complexity of operation as indicated in section 2.1.3. Such a division can be viewed as a method of aggregation leading towards a reduction in the size of the planning and scheduling problems [17]. It is a normal practice to install a small system first and then to build up the complete system in due course. Since apart from the machines, the peripheral equipment themselves constitute a large investment, a phased plan is necessary to implement these systems

An arrangement of machines in the form of a single system has the disadvantages of increased control problems, difficulty in keeping track of parts, increased part movement distances and complex scheduling requirements.

The cellular arrangement of the machines in FMS has the following advantages [17]:

i) Implied reduction in control.

ii) Reduced material handling.

iii) Quick change over of part types within a range of parts.

iv) Better tooling control.

v) Reduced in process inventory.

vi) Reduced expediting.

A schematic diagram of the cellular FMS is shown in Fig. 2.

2.1.5 Design and operation problems in flexible manufacturing systems

The design and operation of any FMS involves a variety of problems. Many of these problems are typical of any manufacturing system. The classification and description of these problems have been given in [29,35]:

i) Strategic Decision Problems

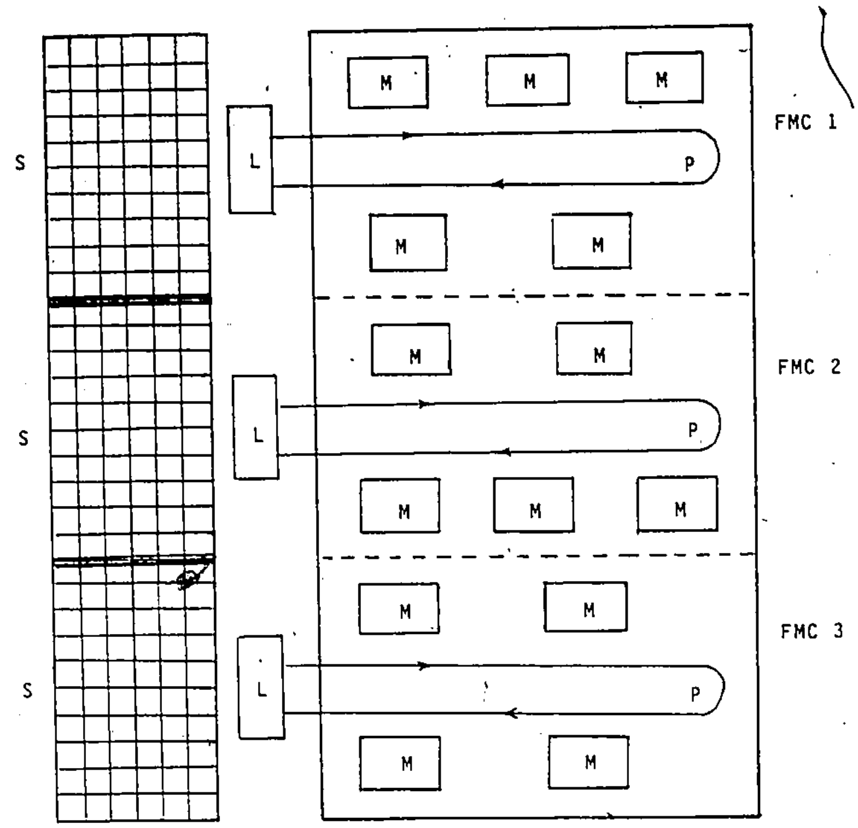
ii) Facility Planning Problems

iii) Intermediate decisions

iv) Dynamic Operations

The strategic decisions are concerned with such problems as the financial and policy decisions in

Figure 2. SCHEMATIC DIAGRAM OF CELLULAR FMS



M - MACHINES
L - LOADING STATIONS
S - STORAGES
P - MATERIAL HANDLING PATHS

implementing FMS.

The facility planning problems are concerned with decisions about initial specification and implementation of the production system. The initial specification decisions include the selection of the parts to be produced, machines & other peripherals and material handling system. The subsequent implementation decisions include layout of the machines, software development and the design of fixtures.

The intermediate range problems are the pre-production decisions on operation allocation, part mix ratio and allocation of other resourses.

The dynamic operations refer to the control problems due to conflicts in the production requirements. These are the in-production decisions on part release rules into the system, scheduling, sequencing, etc.

2.2 Objectives of the Research

2.2.1 Statement of Objectives

The objective of this research is to solve two problems related to the cell formation in flexible manufacturing systems:

- Grouping of the parts into families.
- Allocation of machine groups to families.

These are the initial specification issues in the pre-production planning stage of FMS.

It is important to consider the relevant criterion for grouping the parts into families. The criterion for organizing the cells for manufacturing the parts is based on the processing similarity of the parts. The lack of such similarity has an adverse effect on the operation of the cellular system.

The tools have to be changed intermittently in the tool magazines if parts with different processing requirements are manufactured in the cells. Each tool change puts a certain demand on the system resources for the following activities:

The measurement of cutter compensation and tool offsets (to be supplied to the machines) may have to be carried out when a tool is loaded on to or transferred between the machines. This puts a load on the metrology facilities in the system.

The tool loading is usually done manually in the present systems (although there is an attempt to make this automatic, the use of such automation is not yet widespread [18]) and the frequent tool changing interrupts the operation of the machines.

Frequent tool changing also results in a constant flow of tools within the shop competing with the parts for the resources(trolley, scheduling time on computer etc.). It has been found in some cases that the flow of tools through the shop caused more problems than the flow of workpieces [18].

Hence the processing similarity is considered for grouping parts first to form the part families.

Once the parts constituting different families are determined, the machine group allocation problem will be solved. The objective of such allocation is to provide maximum number of alternative routings for the parts. The diversity in part routing is known to be a very helpful strategy in the operation of the system. It is possible to divert the parts to different machines when the designated machines break down or are busy serving some other parts.

The part family formation and machine group allocation problems are formulated in Chapters 4 and 5.

2.2.2 Typical Problem Situation

Most of the modern manufacturing plants have NC/CNC machines located randomly within the factory. Even though many such machines may be in operation, the net effect on production may not be as significant as can be expected with these versatile machines. The individual machines are really very efficient, but the way in which they are placed in the system may result in low utilization levels. They may be restricted by limitations such as production bottlenecks at other machines and material handling delays.

In this situation, since the NC/CNC machines, which are the major components of FMS are already available, there is an opportunity to reorganize the system into an independent cellular FMS. The machines in the cells can be linked together through a material handling system. Once such a strategic decision is taken, it becomes essential to analyse the part range under consideration to form families and then allot the available machines to each of the families. As mentioned earlier, the implementation can be done in phases, organizing one cell at a time.

2.2.2.1 Part range manufactured in FMS

When the FMS capacity augments the conventional capacity, the part variety chosen for manufacturing in FMS is restricted keeping in mind the need to utilize other high cost plant and auxiliary equipment. The parts chosen are high value, critical components required in the downstream production facilities. A fabrication shop supplying the finished parts to an assembly section is an example of such a situation, where certain parts in the final assembly are invariably in short supply due to the difficulties encountered in manufacturing them in the conventional shops.

This is clearly illustrated by the reports on the existing systems and the restricted component variety they encompass [2].

The literature on FMS [2,18] and experience in a light engineering industry indicates that the parts selected for manufacturing on CNC machines(and hence in FMS) have, in general, the following characteristics:

i) The parts require a large number of processing steps. If loaded in a conventional machine shop these parts have to visit several machines, in most cases one machine carrying out one processing step. This results in a tremendous amount of handling and subsequently a tardy output from the shop. These parts are the right candidates to be manufactured in an FMS, since the CNC machines allow for a number of processing steps to be completed in one visit to the machine.

ii) Heavy emphasis on the milling, drilling, boring and tapping. The existing systems indicate their strength in these processes basically due to the corresponding capabilities offered by the machining centres. 'Difficult' processes such as grinding and honing, mass production oriented processes such as broaching and not-so-common production processes such as planing and shaping (shaper), if required on a part, are usually carried out on the facilities operating in tandem with, but outside, the FMS.

iii) Apart from the problem of excessive handling, the sheer difficulty involved in achieving the complicated process requirements of some parts (in conventional shops) makes them the automatic choice for manufacturing in FMS.

iv) The parts are mostly finished from raw casting state.

Chapter III

LITERATURE SURVEY

The problems of FMS design and operation have been considered using different Operations Research approaches. The major approaches used in the literature are Networks of Queues, Simulation, and Mathematical Programming.

The facilities design problem has two issues as mentioned earlier, the initial specification decisions and the subsequent implementation decisions. These decisions are generally one time decisions, especially the ones concerning the machines constituting the FMS cells. The implementation decisions about the number of pallets and the number of fixtures can be spread over the time of operation of the system.

The queueing network models provide some aggregate results and are perhaps helpful in the decision issues such as the number of pallets and the number of fixtures required in the system. The aggregation may not be acceptable for more specific decisions such as sequencing and scheduling of the parts and the number of buffer spaces required. Simulation is the approach for such problems.

The mathematical models are appropriate for the static decision issues of facility design, operation allocation in the planning stage and fixture & pallet allocation. In such pre-production planning decisions some criteria are used which have been proved to be effective either by experience or by theoretical research in the operation of the system. Providing alternative routings for parts, balancing workloads between machines, minimizing part handling distances, launching similar parts for production, etc., are some examples of such criteria. These would be basically indirect measures, which are recommended as static problem objectives.

Wilhelm and Sarin [35] provide a review about the issue of suitability and limitation of different modelling approaches.

In this research mathematical modelling has been adopted. The criteria adopted in this research are the processing similarity concept for part family formation and routing diversity concept for machine group allocation.

Buzacott and Shantikumar [5] have reported some simple models for the understanding of the FMS. Their approach is to consider the system as an automated job shop. The models are simple and aggregate in nature, but they demonstrate amongst other aspects the importance of diversity in job routing.

Chatterjee et. al [10] have developed a general framework for manufacturing system specification. They present some scheme for manufacturing systems to identify critical distinctions between various types of manufacturing capabilities. They define manufacturing flexibility and identify the number of routings available for a part within a system as the routing flexibility.

Stecke [30] gives an analysis of FMS cell using the queueing network theory. It has been shown that the pooling of machines in FMS cells improves the output of the system. Under a seperate study of a real system through simulation [31], the same result was obtained. The system showed maximum output through the pooling in combination with some scheduling rule. The pooling of machines with reference to an operation means that there is more than one machine available for that operation and the part routing can be through one of the available machines depending on the scheduling decisions in real time.

Thus, providing maximum number of alternative routings has been proved to be a good strategy in operating the system.

One of the principles in Group Technology is to restrict a machine to only one part family (unique allocation). Thus, a certain machine group is made available to the parts in a particular family. However, in practice, some exceptions do exist. The scarcity of certain machines may force the sharing of those machines by more than one family of parts. Certain overlapping referred to as ´cascading´ is allowed in these situations. This possibility has been incorporated in the formulation of machine group allocation.

The literature on the grouping procedures is mostly limited to the conventional systems.

There are two issues in the grouping; part representation and grouping procedure based on this representation of the part. However, as pointed out by King and Nakornchai [20], in the past decade the emphasis has slowly shifted from classification schemes per se to the problem of developing methods for grouping. This has happened mainly due to the realization that most of the classification schemes have to be industry-specific anyway.

A review of the various grouping procedures is given by King and Nakornchai [20]. Recent work in this area includes [6], [8], [21] and [33]. The classification of the available techniques is as follows:

i) Similarity Coefficient methods
ii) Set theoretic methods
iii) Evaluative methods
iv) Other analytical methods.

Similarity coefficient is an approach drawn from numerical taxonomy, and first suggested by McAuley [24]. The basis of the method is to measure the similarity between each pair of machines and then to group the machines based on their similarity measurement.

These methods are called 'hierarchical clustering methods' and are based on some 'threshold value' of coefficients. If a coefficient is less than a predetermined value, the coefficient will be ignored in the next stage of the algorithm. The selection of the threshold values is arbitrary. Rajagopalan and Batra [27] suggest a more systematic method of finding the threshold value; however, the arbitrary nature of the procedure still persists. The hierarchical grouping methods can be explained as follows:

First two parts are selected which have the greatest similarity to form the nucleus of the first group. A third part is added which has the most similarity with the first two. The fourth is added which has the most similarity with the first three and so on. At any stage, if there is no part which has a similarity above a particular level with the parts in the first cluster, a new cluster is formed with the remaining parts in the same manner.

Set theoretic method has been developed by Purcheck [25]. This method considers the lists of machines required for the parts as sets and does set union operations on them. This is a heuristic method for grouping the machines and parts.

Evaluative methods are based on the Production Flow Analysis [4], and basically use the judgement of the analyst. The main feature of the evaluative approach is that it involves listing of components in different ways in the expectation that the groups can be found by careful inspection. This requires manual intervention to identify groups at each stage.

The other analytical methods are based on machine component matrix manipulation. King and Nakornchai [20] and Chan and Milner [7] reported algorithms using this approach. The procedure developed in [33] for finding the bottleneck machines also is based on the matrix representation. Some critisism about King and Nakornchai's algorithm is given in [33]. The principle used is to improve a criterion starting from initial grouping, through some manipulations in the grouping using graph theory. Figure 3 (a) illustrates the typical machine-component matrix used by these methods. In this example, the machines are labelled from A to E and the parts from 1 to 6. An entry of 1 in cell (i,j) indicates that some operation of part j requires processing on machine i, whereas a blank entry means that it does not. The cell entries of 1 are spread around the matrix in a random fashion, so that no particular pattern of machine component grouping is apparent.

Figure 3 (b) shows the same matrix, but after several exchanges of the relative positions of both rows and columns. It will be seen that the original cell entries of

MATRIX MANIPULATION METHODS

Figure 3(a)

PARTS

MACHINES	1	2	3	4	5	6
A	1	1				
B			1		1	
C	1	1		1		
D			1		1	1
E	1	1		1		

Figure 3(b)

PARTS

MACHINES	6	5	3	2	4	1
B		1	1			
D	1	1	1			
E				1	1	1
A				1		1
C				1	1	1

Fig. 3 (a) are preserved unchanged, but now, two machine component groupings (B,D,6,5,3) and (E,A,C,2,4,1) emerge naturally along the diagonal of the matrix as a result of a particular arrangement of the rows and columns of the original matrix. In some cases for geometric feature based grouping instead of having machine-component matrix, a code-component matrix is formed with the same basic idea of grouping the components and features together. The matrix manipulation methods are not mathematically rigorous [23].

Clustering is basically a yes/no type décision of allotting a part to a cluster. A 0-1 integer programming approach for the grouping of the parts has been reported by Kusiak [23], which employs a statistical clustering method [1]. This method considers the 'distance' between the parts and then considers each part as the 'median' of the cluster in the formulation. These concepts of 'distance' and 'median' are vaguely defined. The integer programming approach has also been used for grouping based on the geometric features.

The literature survey indicates that:

i) In the grouping methods reported it is assumed that the operations are restricted to one machine. Based on this single and fixed machine allocation information for the operations a one to one relationship between the parts and machines is defined in the form of a matrix. This leads to a simultaneous grouping of parts and the machines.

ii) The (dis)similarity coefficients and measures of (dis)similarity between two parts, have been adopted with the arbitrary specification of some 'cut off' values as the basis for grouping.

In conventional systems assumption (i) can be justified by noting that each operation on a part is generally restricted to one machine. The machine assignment for different operations of the parts acts as the basis for simultaneous grouping of parts and machines

This assumption is not applicable to flexible manufacturing systems since each operation on a particular part can be performed on alternative machines. In this case, the processing similarity between the parts should be determined by using the basic information about the processsing steps required to manufacture the parts.

The necessity of achieving homogeneity amongst the parts to be produced in an FMS cell, as explained in Chapter 2 is incorporated in the model by defining a dissimilarity coefficient. This coefficient is defined using 0-1 decision variables and is used as the objective function in a fractional programming model.

The procedure developed for cell formation groups the parts first based on the similarity of the processes and subsequently allots the machines to each of the part groups(families).

In the problem of allocation of machines to the part families the concept of providing routing diversity for the parts has been used as the objective function.

Chapter IV

PART FAMILY FORMATION

The mathematical formulation of the part family formation problem is discussed in this Chapter. First, the criterion for grouping based on the manufacturing attributes is explained. The objective function of the formulation is fractional defined on zero-one integer variables. A solution procedure for this situation is outlined. Due to the computational difficulty in solving this model for larger problems, an approximation procedure that yields a good initial solution is developed.

4.1 Formulation

4.1.1 Statement of the Problem

The part family formation is considered with respect to manufacturing attributes for eventually forming the cells.

The objective is to group the parts into part families based on their processing similarities. The similarity based grouping is done to achieve minimum disruptions in the production within the cells as the batches of different parts are launched into production.

Notations Used

N	= Number of parts to be grouped
K	= Number of families
ij	= Index for the part pair (i,j) (i=1,2...N-1 ; j=i+1,i+2...N)
d_{ij}	= Number of dissimilar processes between part i and part j
s_{ij}=	= Number of similar processes between part i and part j
DIS_{ij}	= Dissimilariy coefficient between part i and part j
DC_k	= Dissimilarity coefficient for family k
D_k	= Contribution of family k to the value of CDC
MF	= Family having highest value of D_k
CDC	= Dissimilarity coefficient for a configuration of part families
A(X)	= Linearized numerator of the coefficient CDC
B(X)	= Linearized denominator of the coefficient CDC
P(.)	= Minimization problem of CDC
R	= Parameter in the search procedure for optimal solution to P(.)

P(R)	= Transformed minimization problem with parametric objective function
Z(R,X)	= Objective function of the problem P(R)
C	= Constraint set of the problems P(.) and P(R)
C^1	= Reduced constraint set for problem P(R)
$\underline{Z}(R,X)$	= Minimum of the objective function Z(R,X) subject to C
$\overline{Z}(R,X)$	= Maximum of the objective function Z(R,X) subject to C
LB_R	= Bound on function Z(R,X) for minimization
UB_R	= Bound on function Z(R,X) for maximization
(L,U)	= A range of values of R established such that $L < R^* < U$
C^R_{ij}	= Coefficients of M_{ijk} variables in function Z(R,X).(For convenience the superscript is dropped and the coefficient is denoted by C_{ij}).
J	= Set of C_{ij} s for all (i,j)
NP	= Number of positive C_{ij}s in the set J
NN	= Number of negative C_{ij}s in the set J

<u>Decision variables:</u>

X_{ik} $\begin{cases} = 1 \text{ if the part i is included in family k} \\ = 0 \text{ if the part i is not included in family k} \end{cases}$

These variables for all (i,k) are denoted by (X).

M_{ijk} = Linearization variable introduced to replace the product term $X_{ik} \cdot X_{jk}$

4.1.2 Criterion for grouping

As stated earlier, the objective of the part family formation problem is to group the parts with similar processing requirements. For a part pair (i,j) in a particular family, we would like to have a low ratio of d_{ij}/s_{ij}, indicating that the parts i and j have more operations in common than dissimilar operations.

An Example:

Consider the part pair (i,j) having the processing requirements as shown:

Processes ->	1	2	3	4	5	6	7
part i	1	0	1	1	0	1	1
part j	1	1	0	1	1	0	1

For this part pair, s_{ij}= 3 (processes 1,4 and 7) and
d_{ij}= 4 (processes 2,3,5 and 6).

The dissimilarity between two parts is relevant only when they are grouped together into the same part family. The dissimilarity of two parts in different families is of no concern, since these parts are manufactured in different cells. The grouping should be done such that within the families formed, parts have the minimum dissimilarities and the maximum similarities in terms of processing requirements.

Based on this concept, the coefficient of dissimilarity between part i and part j is defined as:

$$DIS_{ij} = \sum_{k=1}^{K} [d_{ij}/s_{ij}] \cdot X_{ik} \cdot X_{jk} \quad (1)$$

This coefficient is used as the basis for defining the objective function for grouping the parts. The value of DIS_{ij} would be d_{ij}/s_{ij} or 0 respectively, depending on whether parts i and j are grouped in same family k or not.

4.1.3 Definition of Dissimilarity Coefficients

The objective function for the part family formation would be the minimization of an overall measure of processing dissimilarity between the parts. The definition of such a measure considered in this research is explained next. It represents the overall average of the pairwise dissimilarity coefficients. This is similar to the coefficient considered in [12]. Alternative representations for the overall measure of processing dissimilarity are indicated in Appendix A.

The dissimilarity coefficients are defined for each of the families and for the overall partitioning of the parts into families.

i) Dissimilarity coefficient for the family:

The average of the pairwise dissimilarity coefficients of all the parts in family k is given in Eqn.

(2). A high value of this coefficient indicates that the family k contains parts which are highly dissimilar to each other.

$$DC_k = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \cdot X_{ik} \cdot X_{jk}}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \cdot X_{ik} \cdot X_{jk}} \qquad (2)$$

ii) Dissimilarity coefficient for the configuration:

The average of the dissimilarity coefficient of all the part pairs in the configuration can be expressed as:

$$CDC = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \cdot X_{ik} \cdot X_{jk}}{\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \cdot X_{ik} \cdot X_{jk}} \qquad (3)$$

This coefficient is taken as a measure of the overall dissimilarity between the parts in different families in a particular grouping. A high value may indicate the possibility of decreasing the dissimilarity by reallocating some parts from the present configuration. This idea is used in the approximation procedure for allocating the parts between the families to get a good initial solution.

4.1.4 Formulation

The problem could be formulated as follows:

Minimize the overall dissimilarity coefficient:

$$\text{Minimize} \quad Z_1 = CDC \tag{4}$$

Subject to the following constraints:

i) Each part is allocated to only one family:

$$\sum_{k=1}^{K} X_{ik} = 1 \quad \text{for } i=1,2,3.......N \tag{5}$$

ii) Each part family should at least have some specified number of parts say, L. This constraint may or may not be specified.

$$\sum_{i=1}^{N} X_{ik} \geq L \quad \text{for } k=1,2,...K \tag{6}$$

iii) $X_{ik} = 0$ or 1 for $i=1,2,...N$ and $k=1,2,...K$

Let constraints (i),(ii) and (iii) be denoted by C_1 .

The objective function in (4) is a ratio of two non-linear functions. As a first step in solving the problem, the numerator and denominator of the objective functions are linearized. The linearization scheme [14] is explained next.

Consider the term $X_{ik}.X_{jk}$; both X_{ik} and X_{jk} are 0-1 integer variables.

Each of the terms $X_{ik} . X_{jk}$ can be replaced by M_{ijk} with the addition of the following constraints:

iii) $X_{ik} + X_{jk} - M_{ijk} \leq 1$ (7)

iv) $M_{ijk} \leq X_{ik}$ (8)

v) $M_{ijk} \leq X_{jk}$ (9)

The above constraints force the variable M_{ijk} to assume the values 0-1. Let the set of constraints (iii), (iv) and (v) for all i,j and k be denoted by C_2.

With the linearized numerator and denominator, the formulation can be written as follows:

$$\text{Minimize } Z_2 = \frac{A(X)}{B(X)} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \cdot M_{ijk}}{\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \cdot M_{ijk}} \qquad (10)$$

Subject to:

C_1 and C_2

Let C denote the constraint sets C_1 and C_2.

4.2 Solution Procedure

The objective function in (10) is a ratio of two linear integer functions A(X) and B(X). This type of problem is referred to as fractional programming in the literature. Methods have been reported for solving the fractional programming models with continuous decision variables [11,32,34]. The objective function in (10) being defined on zero-one integer variables, does not lend itself to these methods. Hence, in this case, a general search principle [9] has been adopted which involves solving a series of linear/non-linear problems to arrive at the optimal solution to a fractional programming problem. A description of this principle is given in section 4.2.2.

Section 4.2.2 also describes a method developed for this model to restrict the search area and subsequently to reduce the search time. This method identifies a range of values in which the optimal value for the ratio A(X)/B(X) exists.

4.2.1 Parametric Search Principle

The objective function (10) can be expressed as:

$$P(.): \quad \min_{X \in C} A(X)/B(X) \qquad (11)$$

Let (X^*) be the optimal solution to P(.)

Then,

$$\min_{X \in C} A(X)/B(X) = A(X^*)/B(X^*) = R^*$$

and

$$A(X^*) - R^*.B(X^*) = 0$$

Consider the following problem:

$$P(R): \min_{X \in C} Z_3 = \min_{X \in C} [A(X) - R.B(X)] = \min_{X \in C} Z(R,X) \qquad (12)$$

$$= \underline{Z}(R)$$

The function Z(R,X) for a particular (X) decreases with increasing values of R, since, both the functions A(X) and B(X) have only positive coefficients (d_{ij} and s_{ij} respectively) and are defined over the same set of non-negative variables (M_{ijk}'s). It follows that the optimal value of Z(R,X) will also behave in a similar manner with respect to changes in R. This characteristic of Z(R,X) helps in deciding the direction of search for the optimal ratio A(X)/B(X). The value of the parameter R which gives a value of $\underline{Z}(R) = 0$ is the optimal ratio A(X)/B(X). This will be clear from the following:

a) <u>Suppose $R=R_0$ and Optimal (X) = (X_0)</u>

Then, $Z(R_0,X_0) = A(X_0) - R_0 . B(X_0) = 0$ (say)

Since $\min_{X \in C} A(X) - R.B(X) = A(X_0) - R_0 . B(X_0) = 0$

$$\Rightarrow A(X_0)/B(X_0) = R_0$$

Now, $A(X) - R_0 . B(X) \geq 0$ for all $X \in C$

$$\Rightarrow A(X)/B(X) \geq R_0$$

$$\Rightarrow R^* = R_0 = A(X^*)/B(X^*) \qquad (13)$$

b) <u>Suppose $R=R_1$ and Optimal (X) = (X_1)</u>

Then, $Z(R_1,X_1) = A(X_1) - R_1 . B(X_1) > 0$ (say)

Since $\min_{X \in C} A(X) - R.B(X) = A(X_1) - R_1 . B(X_1) > 0$,

$A(X) - R_1 . B(X) > 0$ for all $X \in C$

$$\Rightarrow A(X)/B(X) > R_1$$

$$\Rightarrow R^* = A(X^*)/B(X^*) > R_1 . \qquad (14)$$

c) <u>Suppose $R=R_2$ and Optimal (X) = (X_2)</u>

Then, $Z(R_2,X_2) = A(X_2) - R_1 . B(X_2) < 0$ (say)

$$\Rightarrow A(X_2)/B(X_2) < R_2$$

Since $A(X_2)/B(X_2) < R_2$,

The optimal solution,

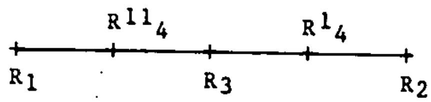
$$\underset{X \in C}{\text{Min}}\ A(X)/B(X) = A(X^*)/B(X^*) = R^* < R_2$$

i.e $$A(X^*)/B(X^*) < A(X_2)/B(X_2) < R_2 \quad (15)$$

Hence from (b) and (c),

$$R_1 < R^* < R_2 \quad (16)$$

Now, consider $R_3 = (R_1 + R_2)/2$



If $\underline{Z}(R_3) = \underset{X \in C}{\text{Min}}\ A(X) - R_3.B(X) > 0$

then, $R_3 < R^* < R_2$
(from similar arguements in (b) and (c))

Now consider $R^1{}_4 = (R_2 + R_3)/2$ and continue the search.

If $\underline{Z}(R_3) = \underset{X \in C}{\text{Min}}\ A(X) - R_3.B(X) < 0$

then, $R_1 < R^* < R_3$
(from a simialr arguement in (b) and (c))

Now consider $R^{11}{}_4 = (R_1 + R_3)/2$ and continue the search.

The solution for problem P(.) is obtained from a binary search for the parameter R which gives $\underline{Z}(R) = 0$.

In other words, the search for R^* can be carried out by solving a series of problems P(R) with different values of R, each time selecting the value of R depending on the optimal solution of the previous problem.

4.2.2 Finding an Interval (L,U) Such that $L < R^* < U$

By initially choosing a value of R too far away from R^*, a considerable amount of computation will be required to converge on R^*. Hence it is necessary to identify a range of R in which R^* lies. This can be done by finding the upper bound and lower bound for the function Z(R,X) at different values of R with some constraints relaxed. If for a particular R, both these bounds are positive, the problem P(R) need not be solved, since it is known beforehand that the optimal solution to P(R) cannot be zero. A similar argument holds for the case when both the upper bound and the lower bound are negative.

Consider P(R) : $\min_{X \in C} Z(R,X) = \underline{Z}(R)$

and $\max_{X \in C} Z(R,X) = \overline{Z}(R)$

Let C^1 be any subset of set C (Constraint set C has been defined earlier). Consider the minimization and the maximization of Z(R,X) under C_1 (i.e., fewer number of constraints).

Let $\min_{X \in C^1} Z(R,X) = LB_R$, the lower bound.

$\max_{X \in C^1} Z(R,X) = UB_R$, the upperbound.

Now, for all R,

$$LB_R \leq \underline{Z}(R) \leq UB_R \qquad (17)$$

$$LB_R \leq \overline{Z}(R) \leq UB_R \qquad (18)$$

It is evident that only those values of R which give a negative LB_R and a positive UB_R have to be considered in the search for R^*. The changes in values of LB_R and UB_R with respect to the changes in R are indicated in Table 1. R^* lies in the region (L,U). In this region the binary search principle outlined in Section 4.2.1 can be applied with R as the parameter.

Another point to be noted here is that, although a strict binary search plan requires the whole region (L,U) to be searched, actually it is possible to restrict to the lower end of the region (L,U). The basic strategy of the search is to solve the problems with different values of R, looking for a value of R that gives the value of $\underline{Z}(R)$ equal to zero. Since LB_R is a relaxed solution to the minimization of Z(R,X), it can be expected that this will occur (i.e., $\underline{Z}(R) = 0$) at those values of R giving LB_R value closer to zero on the negative side.

Establishing the interval (L,U) will reduce guesswork and the computational requirements of the search.

4.2.3 Establishing Lower Bound(LB_R) And Upper Bound(UB_R) for Z(R,X)

4.2.3.1 Constraints on the function Z(R,X)

Consider P(R) : $\min_{X \in C} \; A(X) - R\,B(X)$

TABLE 1

Illustration of the Region (L,U) for P(R)

Value Of R	Sign of LB_R	Sign of UB_R	
0	+	+	
0+s	+	+	Case I
0+2s	+	+	
.	.	.	
.	.	.	
$0+k_1s$	−	+	L ——
$0+(k_1+1)s$	−	+	
.	.	.	Case III
.	.	.	
$0+k_2s$	−	+	U ——
$0+(k_2+1)s$	−	−	
.	.	.	
.	.	.	Case II
$0+k_3s$	−	−	

s is a small increment.

i.e., $\min_{X \in C} Z(R,X)$

i.e., $\min_{X \in C} \sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}.M_{ijk} - R.[\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij}.M_{ijk}]$

i.e., $\min_{X \in C} \sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij} - R.s_{ij}) M_{ijk}$ (19)

Let $(d_{ij} - R.s_{ij}) = C^R_{ij}$. In the discussions to follow the superscript R has been dropped for convenience.

The following conditions are implied by the constraint set C.

(i) M_{ijk}'s are zero-one variables (forced to assume values of 0/1).

(ii) Let $S_{ij} = [M_{ij1}, M_{ij2}, M_{ij3} \ldots M_{ijK}]$.
Since each of the parts i and j can be allotted to only one family, at most one variable in the set S_{ij} can assume a value 1
This can be illustrated by Table 2.

(iii) At least IP number of variables M_{ijk} should have a value 1 where IP is given by the following expression:

$$IP = [N/K].\{[N/K]-1\}/2 * K + (N - K.[N/K]) * [N/K]$$

where, [N/K] is the largest integer less than or equal to N/K.

This expression represents the minimum number of part pairs (which in turn corresponds to the minimum number of M_{ijk} variables taking the value 1) that have to be formed while grouping N parts into K families. It is impossible to form K families out of N parts without forming at least IP part pairs. The expression for IP has been derived by trial and error.

This can be illustrated by the number of distinct possible groupings possible for different values of N given in Table 3:

For N=2 and K = 3 IP=0

N=3 and K = 3 IP=0

N=4 and K= 3 IP=1

Since this is true for all N, it follows that at least a minimum number of M_{ijk}'s must take the value of 1 in any feasible solution to P(R).

(iv) Some combinations of M_{ijk}'s cannot take the value of 1 in the same solution.

For example, consider the part pair (1,2) has been in family 1.

Then $M_{121} = X_{11} \cdot X_{21} = 1 \cdot 1 = 1$

In this case, the variable $M_{132} = X_{32} \cdot X_{12} = 0$

Since, $X_{11} + X_{12} = 1$ & $X_{11}=1$ $\Rightarrow X_{12}=0$

i.e., M_{121} and M_{132} cannot take a value 1 at the same time.

TABLE 2

Part pair (1,2)

K, the number of families = 2

No.	Allocation F1	Allocation F2	Value of M_{121}	Value of M_{122}
1	1	2	0	0
2	2	1	0	0
3	1,2	-	1	0
4	-	1,2	0	1

Note: $M_{121} = X_{11} \cdot X_{21}$ and

$M_{122} = X_{12} \cdot X_{22}$

It is clear that at most one value in the set

S_{12} takes a value of 1

TABLE 3

IP, The Minimum Number Of Part Pairs

K, the number of families = 3

N	Distinct groupings	F1	F2	F3	No of part pairs	
2	1	1	2	-	0	*
	2	1,2	-	-	1	
3	1	1	2	3	0	*
	2	1	2,3	-	1	
	3	1,2,3	-	-	2	
4	1	1	2	3,4	1	*
	2	1,2	3,4	-	2	
	3	1	2,3,4	-	2	
	4	1,2,3,4	-	-	3	

If any one of these conditions are violated by a variable M_{ijk}, some of the constraints in the set C will be violated. The constraint set C minus the violated constraints is denoted as C^1.

The procedure explained in the next section neglects condition (iv) and finds the maximum (UB_R) and the minimum (LB_R) of the function P(R) under a reduced set of constraints C^1. As shown in Section 4.2.2, UB_R and LB_R will be the bounds on the objective function P(R) for maximization and minimization respectively subject to the constraint set C.

4.2.3.2 Coefficients of the function Z(R,X)

The M_{ijk}'s are 0-1 variables. Hence the objective function Z(R,X), is the sum of all C_{ij}'s corresponding to the M_{ijk}'s taking a value of 1. From conditions (i), (ii), (iii) and (iv) it follows that:

1) If C_{ij}'s are positive for all (i,j), then both the upper bound and lower bound would be positive (Case I).

2) If C_{ij}'s are negative for all (i,j), then both the upper bound and lower bound would be negative (Case II).

3) If some C_{ij}'s are positive and some C_{ij}'s are negative, then the value of R may correspond to

Case III. It is necessary to find the LB_R and UB_R in this situation.

4.2.3.3 Algorithm for finding LB_R and UB_R

Although each C_{ij} is a coefficient for K linearization variables, from condition (ii) in Section 4.2.3.1 it can be counted in only once for any solution.

The lower limit (upper limit) of Z(R,X) can simply be found by counting in all the negative (positive) C_{ij} s. However if the condition (iii) is not satisfied some positive (negative) terms should be counted in.

Algorithm

1. Sort the set J in ascending order.

2.0 $Sum_L = 0$

 2.1 Add all the negative C_{ij}'s to Sum_L

 2.2 If NN > IP go to step 3.0

 2.3 If NN < IP then add to Sum_L the first (IP-NN) positive terms in set J

3.0 $Sum_U = 0$

 3.1 Add all the positive C_{ij}'s to Sum_U

 3.2 If NP > IP go to step 4.0

 3.3 If NN < IP then add to Sum_U the last (IP-NN) negative terms in set J

4.0 $LB_R = Sum_L$ and $UB_R = Sum_U$

As indicated earlier, this algorithm neglects condition (iv) implied by the constraint set C.

4.2.4 Summary of the steps for solving the formulation

In brief, the steps in solving problem P(.) are:

1. Set up the objective function P(R).
2. For different values of R starting with 0 find LB_R and UB_R.
3. Establish the interval (L,U) by the method explained in Section 4.2.2 .
4. Carry out the binary search for the value of R^* in this interval (L,U).

   Or

   Choose a smaller interval (R_1,R_2) in the lower end of the range (L,U) and carry out the search for R^*(since R^* is expected to be at the lower end of the region (L,U)).
5. Stop when a value of R yields $\underline{Z}(R,X)=0$

4.3 Approximation Procedure

4.3.1 Need for Finding a 'Good' Initial Solution

The grouping problem is combinatorial in nature.

Each of the N parts can be allocated to one of the K families independently. Hence the number of feasible solutions to the family formation problem is K^N, which becomes too large with increasing values of N. The solution time required for the problem P(R) will also increase rapidly.

It is clear that the time required for the search as outlined in section 4.2.1 is dependent on the number of problems P(R) solved in the process. The number of problems P(R) required to be solved depends on the interval (R_1,R_2) chosen initially.

If the time required for each problem P(R) is high, it is desirable to limit the number of such problems solved to as few as possible. This means that a tight interval (R_1,R_2) has to be selected.

The approach suggested in this case is to find a 'good' solution and choose the corresponding CDC as the upper limit on the value of R. Any procedure for finding such an initial solution should be expected to satisfy the following requirements:

a. The solution should be 'good'. An initial solution is considered to be better than others if the corresponding CDC is nearer to the value of L (L is the lower limit on the value of R). This results in a shorter search interval.

b. The time required to arrive at that solution should be justifiable.

An approximation procedure has been developed for this purpose. Both the above requirements have been found to be satisfied by the procedure in the several problems solved.

4.3.2 Principle

The approximation procedure also uses the IP formulation described in section 4.1.4. In this case however, a number of smaller problems are solved instead of a single large problem. The procedure is based on a method of clustering first reported by Friedman and Rubin [13]. Whereas the principle in [13] is single reallocation based, the procedure developed in this section is multiple reallocation based.

A random partitioning of N parts into K families is considered initially to start off the approximation procedure. Let $n_1, n_2, \ldots n_K$ be the number of parts in families 1,2,.....K, respectively.

4.3.2.1 Single Reallocation

The principle as applied to part grouping problem is given below:

Start with a random partitioning of parts into K families. The parts are considered in a particular order for moving into other families. The part selected is moved to some other family such that it brings about the maximum favourable change in the objective function. This reallocation generates a new configuration, and causes the coefficients to assume new values. The procedure restarts each time a reallocation move is made. If the reallocation fails to bring about a favourable change, the part is retained in its present family and the next part in the order is selected. This continues until no part can be moved from its present family to another.

This approach to part grouping has been applied by Dutta et.al [12], for the part family formation. Different trials were conducted with varying starting partitions. The final objective function values were very close to each other irrespective of the starting configurations.

4.3.2.2 Multiple Reallocation

It can be noted that in the single move algorithm, each time a part is considered for reallocation, a decision is taken with respect to each family about moving the part to that family. The value of the objective function is calculated for all the possible reallocations. This, in effect, means that a problem with 1 x K integer variables (0-1) is solved each time by complete enumeration.

Extending the same principle, all the parts of a particular family 's' (to be chosen based on some criterion) can be considered for reallocation.

If there are n_s parts in family 's', the number of feasible reallocations is $n_s \times K$, which is quite large even for small values of n_s, if a complete enumeration has to be attempted. However, the reallocation of these n_s parts can be considered using the formulation in section 4.1.4. The allocation of all the other parts in other families is fixed and the reallocation of the n_s parts into K families is considered.

A series of smaller sub-problems are solved, until a stopping criterion is reached. The criterion for choosing the family to be considered for reallocation is the value of D_k for different part families(k=1,2,...K).
We define for a family k,

$$D_k = \left[\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \; X_{ik} \cdot X_{jk} \right] / B(X) \qquad (20)$$

A high value of D_k indicates that the parts in family k are such that the contribution from the part family k to the value of CDC is very high, which suggests the presence of highly dissimilar parts in that family. This means that the parts from this family are the candidates for reallocation. The family with highest D_k (family, MF) is considered for reallocation of parts at an iteration of the algorithm.

As indicated, a sub-problem of the form P(.) with n_{MF} x K integer variables is solved in an iteration. After each successful iteration (iteration causing an improvement in the objective function), the algorithm returns to a stage similar to the initiad configuration with an improved bound on the value of the objective function. The algorithm terminates when the reallocation of parts from any family fails to bring about an improvement in the objective function

4.3.3 **Algorithm**

The flowchart of the algorithm is s(own in Fig6 1. A brief explanation of the flowc`art blocks follows.

- Block (a) initialize the algorithm by computing the values of the objective function Z_1 and D_k for all the families.
- An iteration of the procedure involves reallocation cosideration of all the parts in a family for improving the value of the objective function Z_1.
- Blocks (b), (c) and (d) represent the main steps in an iteration. The reallocation subproblem is solved as an IP of the same form as formulated in section 4.1.4

- Decision block (e) indicates whether a reallocation of the parts has to be made or not.
- The loop (e)-(f)-(g)-(h)-(a) represents the steps involved when a decision to reallocate the parts from family 'MF' is made.
- When it is impossible to reallocate the parts in family 'MF' under consideration, the loop (e)-(i)-(j)-(b) represents the choice of another family for considering reallocation of parts.
- Block (i) identifies the families from which the parts could not be reallocated within an iteration.
- The procedure terminates when the decision block (j) returns a result YES.

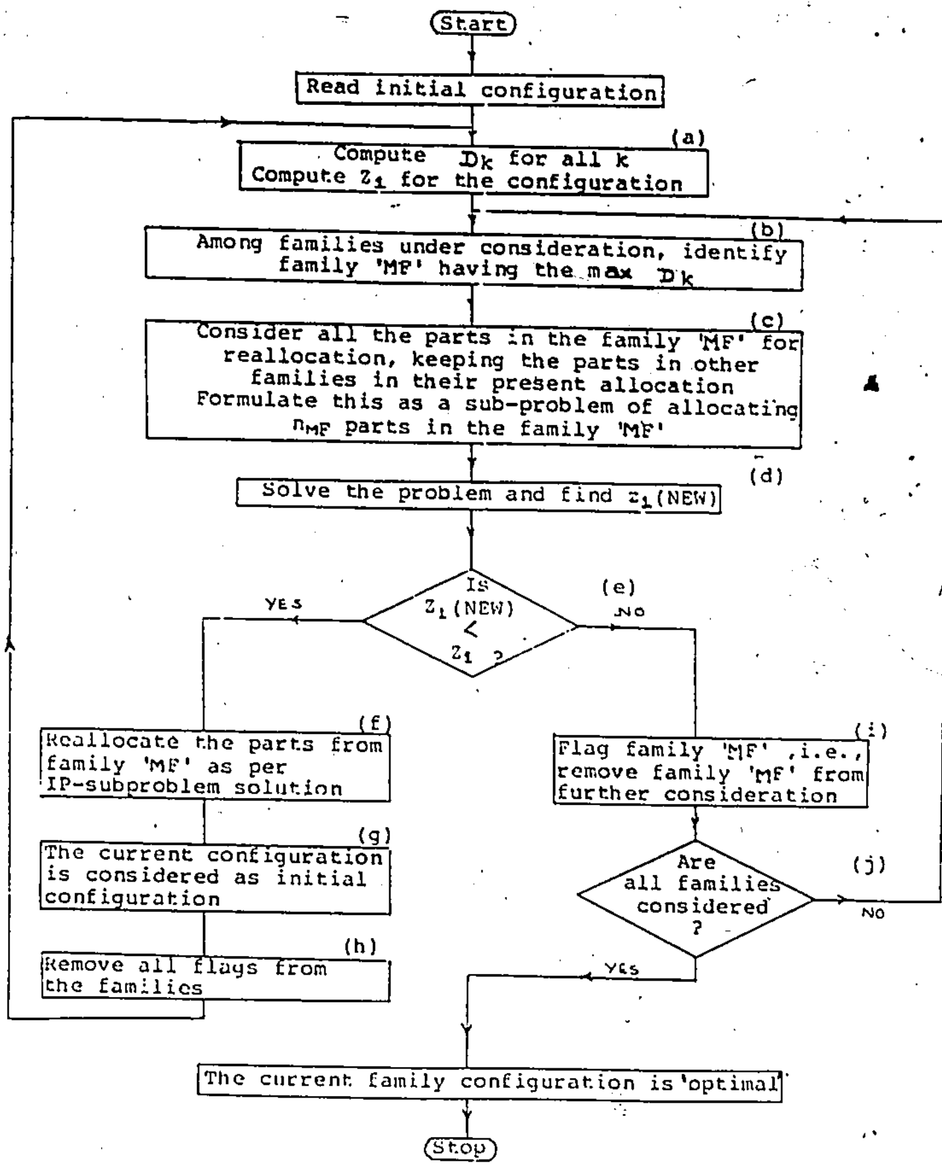


FIGURE 4. FLOWCHART OF THE ALGORITHM

Chapter V

MACHINE GROUP ALLOCATION

The assignment of machine groups for the production of the parts segregated into part families is discussed in this chapter. The availability of alternative machines for each of the operations on the parts is considered. Our objective is to maximize the number of available alternative routings for the parts within the cellular system. The formulation allows for the individual machines to be allocated to only one part family. When such allocation is infeasible, the machine(s) causing this infeasibility is(are) identified through a mathematical model. The condition of allocation to only one family is relaxed for these machines.

5.1 Formulation

5.1.1 Statement of the Problem

All the parts of a particular family have to be processed completely within the corresponding machine group. These machine groups constitute the FMCs or cells. The aim of this problem is to allocate a group of machines to each of the part families.

5.1.2 Objective

The objective is to provide the maximum possible number of alternative routings for the parts within their respective machine groups. The availability of alternative routings is known as the routing flexibility. Routing flexibility is maximized taking into account the operation requirement of the parts in different families. The routing for a part is defined as a sequence of machine visits needed to complete the operations required. Consider a part with three operations. A sequence of visits to the machines 3,2 and 5 represents a routing for that part.

5.1.3 Concept of Alternative Routings

Notations

N = Total number of parts

K = Number of part families

= Number of machine groups to be formed

n_k = Number of parts in family k.

Therefore, $\sum_{k=1}^{K} n_k = N$

jk = Index for part j in family k ($j=1,2,3..n_k$)

$O_{(jk)}$ = Number of operations on a part indexed by the part identity jk

M = Number of available machines

PR_{jk} = Maximum possible number of routings for part jk

FP_{jk} = Number of alternative machines available for operation p of part jk.

NR_{jk} = Number of alternative routings available for part jk

S_k = Product terms of the decision variables (to be defined later) to indicate the allocation of individual machines to k families (k=2,3...K). $P_{i,k}$ indicates the product term i in set S_k.

W_k = Penalty weight to k family allocation of a machine in the model to identify the machines causing infeasibility in the machine group allocation formulation.

The feasible routing for the operations is represented in the form a matrix as explained below: Consider a part jk with $O_{(jk)} = 3$. Assume $M = 9$. The matrix A_{jk}, with elements $a_{(jk)pm}$ indicates the feasibility of operation p of part jk on different available machines.

Op	Mc 1	2	3	4	5	6	7	8	9	
1	1	0	1	0	1	0	0	0	1	4 feasible machines
2	0	1	0	0	0	0	1	0	1	3 feasible machines
3	0	0	0	1	1	1	0	1	0	4 feasible machines

For example, operation 1 can be done on machines 1, 3, 5, or 9.

This information is assumed to be available based on the technological capabilities of the available M machines, which can be expressed by a matrix for each part jk.

Decision variables

$$I_{mk} \begin{cases} = 1 & \text{if machine } m \text{ is allocated to cell } k. \\ = 0 & \text{otherwise.} \end{cases}$$

Referring to the previous example, if all the machines 1, 2,...., 9 are assigned to family k, all the possible alternative routings, PR_{jk} will be available for the part jk.

$$PR_{jk} = \begin{bmatrix} \text{\# of machines} \\ \text{available for} \\ \text{first operation} \end{bmatrix} \cdot \begin{bmatrix} \text{\# of machines} \\ \text{available for} \\ \text{second operation} \end{bmatrix} \cdot \begin{bmatrix} \text{\# of machines} \\ \text{available for} \\ \text{third operation} \end{bmatrix}$$

$$PR_{jk} = [\sum_{m=1}^{M} a(jk)1m] \cdot [\sum_{m=1}^{M} a(jk)2m] \cdot [\sum_{m=1}^{M} a(jk)3m] \quad (21)$$

$$= 4 \cdot 3 \cdot 4 = 48$$

This may be the most desirable situation as far as the production of part jk is concerned. But, all the machines cannot be allocated to family k, since there will be a requirement of these machines for parts in other families. Due to these requirements, allocation of the machine groups has to be done for each of the families with the objective of maximizing the number of routings available.

5.1.4 Formulation

Consider the decision variable I_{mk} which represents the allocation of machine m to family k. Each machine can be allocated to only one family.

If a machine m is allocated to family k, it offers a routing possibility for all the operations in that family that can be done on machine m.

A formulation with the objective function to maximize the number of alternative routings available will try to allocate those machines to a particular family, which offer routing to a large number of operations.

For operation p on part jk the number of alternative machines available is given by:

$$FP_{jk} = \sum_{m=1}^{M} a(jk)pm \cdot I_{mk} \tag{22}$$

The number of available routings for part jk is then given by:

$$NR_{jk} = F^1_{jk} \cdot F^2_{jk} \cdot F^3_{jk} \cdot F^4_{jk} \cdots F^{O}_{jk} \qquad (23)$$

The total number of alternative routings over all the parts in all the families is given by:

$$Z = \sum_{k=1}^{K} \sum_{j=1}^{n_k} [F^1_{jk} \cdot F^2_{jk} \cdot F^3_{jk} \cdot F^4_{jk} \cdots F^{O}_{jk}] \qquad (24)$$

The objective function can be stated as :

P1 Maximize Z (25)

Subject to the following constraints:

1. Each operation p on the part jk must have at least one feasible machine in the corresponding group. This ensures that for operation p at least one routing is provided in the corresponding group of machines,

$$\sum_{m=1}^{M} a_{(jk)pm} \cdot I_{mk} \geq 1 \quad \text{for} \quad \left.\begin{array}{l} k=1,2,3\ldots\ldots K. \\ j=1,2,3\ldots\ldots n_k \\ p=1,2,3\ldots\ldots O_{jk} \end{array}\right\} \qquad (26)$$

2. Each of the available machines can be allotted to one cell only.

$$\sum_{k=1}^{K} I_{mk} = 1 \qquad \text{for } m=1,2,3\ldots\ldots M \qquad (27)$$

3. Integrality constraints:

$I_{mk} = 0 \text{ or } 1$ (28)

for $m = 1,2,3.......M$

$k = 1,2,3.......K$

<u>Example of the expression for NR_{jk}</u>

The objective function of this formulation is the sum of alternative routings available for all the parts.

The expression for the number of alternative routings available for part jk is developed as follows:

$$F1_{jk} = \sum_{m=1}^{M} a(jk)pm \cdot I_{mk}$$

$$= 1 \cdot I_{1k} + 0 \cdot I_{21k} + 1 \cdot I_{3k} + 0 \cdot I_{4k} + 1 \cdot I_{5k} + 0 \cdot I_{6k} + 0 \cdot I_{7k} + 0 \cdot I_{8k} + 1 \cdot I_{9k}$$

$$= I_{1k} + I_{3k} + I_{5k} + I_{9k}$$

Similarly,

$F2_{jk} = I_{2k} + I_{7k} + I_{9k}$

$F3_{jk} = I_{4k} + I_{5k} + I_{6k} + I_{8k}$

NR_{jk}, the number of alternative routings available for the part jk =

$$(I_{1k}+I_{3k}+I_{5k}+I_{9k}) \cdot (I_{2k}+I_{7k}+I_{9k}) \cdot (I_{4k}+I_{5k}+I_{6k}+I_{8k})$$

5.2 Solution Procedure

The objective function Z is non-linear in integer variables. The non-linear terms, each of which represents the possibility of a routing are the product terms of the decision vearlables I_{mk}.

These terms are linearized by introducing additional variables using a scheme suggested in [12,13]. This is similar to the method adopted in the part family formation problem, where product terms of two integer variables were considered. In this case however, each term corresponding to a routing for a part jk will be a multiplication of $O_{(jk)}$ integer variables.

5.2.1 Linearization of Product Terms

The scheme for linearizing the product terms of zero-one variables is [12,13]:

Let Q be the index set of the variables in a particular product term.

- Replace each of the product terms of the type $(x_j)^k$ by x_j.
- Replace each of the product terms of the type : $\prod_{j \in Q} x_j$ by x_Q and add the constraints,

$$\sum_{j \in Q} x_j - x_Q \leq q-1$$

and,

$$x_Q \leq x_j$$

where q is the number of elements in Q.

The linearization strategy adopted for the problem of part family formation is a specific case of this with q = 2.

This formulation is straightforward once the product terms are linearized.

5.2.2 Some Reductions in the Number of Product Terms

The number of variables in the formulation is problem specific, depending on the number of operations for the parts, number of possible machines for each operation and the number of machines. It was mentioned that each routing possibility for a part is denoted by a product term of $O_{(jk)}$ decision variables. However a careful consideration while generating the problem can result in a reduction of the actual number of terms.

For example, a routing for a particular part in the family k may be through machines 1-2-3. This routing will be identified by the product term $I_{1k} \cdot I_{2k} \cdot I_{3k}$. It can be noted that, this product term will also represent the routings 1-3-2 and 3-2-1 for any other operation for the parts in that family. Taking care of these situations while generating the problem for input to an IP routine would be helpful.

5.3 Infeasibility in Machine Group Allocation

It is assumed in Section 5.2 that one or more allocations of the machine groups to the families exist, such that each machine is allocated to only one family.

The condition that one machine should be allocated to only one family may not be possible sometimes due to the problem data.

The reason for infeasibility is the absolute necessity of some machine(s) to be in more than one family. The infeasibility can be removed from the problem by relaxing the assignment constraint (27) on the machine(s). These machines are allowed to be allocated to more than one family.

5.3.1 Multiple Family Allocation of Some Machine(s)

As mentioned earlier, the requirement of some machines in more than one family causes the infeasibility in the machine group allocation problem, P1. The possible cases are the requirement of some machine(s) in two, three,..... or K families.

The problem P1 can be made feasible by relaxing the allocation constraints on the machine(s) as follows:

Allow for some machine(s) to be allocated to two families and check for the feasibility of problem P1. If the problem is not feasible, then allow for some machine(s) to be allocated to three families and check for the feasibility of problem P1, and so on.

The rationale of the above strategy is to allow for sharing of some machine(s) by the least possible number of part families to make problem P1 feasible.

An objective function is defined in the next Section that implements this strategy and identifies the machine(s) for which the assignment constraints have to be relaxed.

Consider the constraints of problem P1, with constraint (27) modified as follows:

$$\sum_{k=1}^{K} I_{mk} \leq K \qquad (27\text{-a})$$

Let S_1 denote the decision variables I_{mk} (Number of decision variables = M.K). The possible multiple allocation variables defined by the original decision variables are listed in Table 4.

5.3.2 Mathematical Model to Identify the Machines Causing Infeasibility

All the product terms in S_2, S_3... S_K take the

TABLE 4

Product Terms Indicating Multiple Allocation Of Machines

Description of Allocation	Product Terms	Example of a product term	Number of product terms
Two families	S_2	$I_{mk} \cdot I_{ml}$	$M \cdot \binom{K}{2}$
Three families	S_3	$I_{mk} \cdot I_{ml} \cdot I_{mj}$	$M \cdot \binom{K}{3}$
. . .	. . .	. . .	. . .
K families	S_K	$I_{mk} \cdot I_{ml} \cdot \; I_{mK}$	$M \cdot \binom{K}{K}$

values 0-1 depending on the value of the decision variables in each of these product terms. Consider the minimization of an objective function Y involving the above product terms.

Based on the strategy explained earlier, the coefficients of the product terms in Y should be such that:

- No term in S_2 takes a value of 1, if the constraints can be satisfied by having the decision variables I_{mk} to assume the value of 1 without any multiple assignments.
- No term in S_3 takes a value of 1 if the constraints can be satisfied by having the terms in S_1 (without any multiple assignments) and S_2 to take a value of 1.

.

.

- No term in S_K takes a value of 1, if the constraints can be satisfied by having the terms in S_1 (without any multiple assignments), $S_2, \ldots\ldots S_{(K-1)}$ to take a value of 1.

Consider a weightage of zero for the terms in S_1, indicating no penalty to the objective function value for any single family allocation of a machine.

The terms in S_1, $S_2, \ldots. S_K$ are given increasing values of penalty weightages. An example is given in Table 5, which satisfies the conditions listed above. Any non-negative value for D will give the same solution to the

TABLE 5

Penalty Weights to Multiple Allocation Of Machines

Product Terms	Weightage Equation
S_1	0
S_2	$W2 = O \cdot M \cdot K + D$
S_3	$W3 = O \cdot M \cdot K + W2 \cdot M \cdot \binom{K}{2} + D$
S_4	$W4 = O \cdot M \cdot K + W2 \cdot M \cdot \binom{K}{2} + W3 \cdot M \cdot \binom{K}{3} + D$
.	
.	

minimization of Y i.e., the formulation is independent of the value of D chosen.

The mathematical model to identify the machines causing the infeasibility can now be written as follows:

<u>INF</u> $$\text{Minimize } Y = \sum_{k=1}^{K} \sum_{P_{i,k} \in S_k} W_k \cdot P_{i,k}$$

Subject to constraints (26), (27-a) and (28).

5.4 <u>Summary of steps for solving the formulation Allocation Problem</u>

In brief the steps to be followed are:

i) Solve the problem <u>INF</u> to identify the machines which have to be allocated to more than one family.

ii) Relax the assignment constraints in the formulation P1 for the machine(s) identified in (i).

iii) Solve problem P1 for the allocation of machines to maximize the number of routings available for the parts.

Chapter VI

APPLICATION OF THE FORMULATIONS

The application of the formulations of part family formation and machine group allocation is illustrated in this chapter. Section 6.1 gives a description of the problem data. Details about the software written for generating the input problem matrix for a the integer programming routine of SAS/OR (Version 5) [29] are provided in section 6.1.2. Solution procedures for the two problems are discussed in detail in Sections 6.2 and 6.3 . A discussion about the application of the formulations and the scope for future work has been included in Section 6.4.

6.1 The Problem Data

6.1.1 Parts and Machines

The problem data considered represent the typical part spectrum characteristics and the machine tool variety in the Flexible Manufacturing systems.

6.1.1.1 **Parts Spectrum**

A set of fifteen parts suitable for manufacturing on CNC machines and hence the natural choice for manufacturingin an FMS are considered. Sketches of these parts are given in Appendix B.

Process details required for the part family formation have been written for these parts and are also provided in Appendix B. A summary of the process requirements is given in Table 6.

When a part visits a machine, a number of processing steps can be carried out and this set of processing steps constitutes an operation. Referring to the process details for Part #1 (HOUSING), 'Rough Mill Surface (A)', is a processing step whereas, (1) which is a combination of nine processing steps is an operation.

These parts have the characterestics explained in Section 2.2.2.1 .

6.1.1.2 Machines

The machines assumed to be available for allocation to part families are basically the variety of machining centres found in FMSs. The two major types of machining centres are Horizontal Spindle and Vertical Spindle. Heavy boring operations are done on designated

TABLE 6

SUMMARY OF PROCESS REQUIREMENTS

PART #	NUMBER OF TOOLS AND THE TOOL REQUIREMENTS	(TOOL CODES ASSUMED)
1	26	M501 M502 M701 M602 M101 M401 M102 M103 M301 M702 M603 M108 M503 M504 D201 D142 D109 D202 D130 R142 R130 R148 B108 B109 B101 B115
2	18	M101 M103 M104 M105 M901 M602 M501 M506 M508 M402 D201 D125 R128 S104 B101 B109 B105 B104
3	23	M403 M404 M701 M401 M412 M405 M406 M702 M712 M101 M102 D202 D128 D203 R130 T130 B108 B109 B101 B102 B115 B106 B112
4	23	M501 M518 M502 M701 M712 M211 M101 M102 M212 M503 M301 M302 M401 M405 D202 D150 D201 D120 D202 B108 B112 S109 T120
5	29	M102 M401 M103 M105 M602 M603 M501 M513 M502 M503 M101 M701 M702 M402 M508 M406 M610 M301 M305 D201 D108 D150 D203 D170 R108 B106 B109 B108 S101
6	18	M501 M504 M502 M415 M412 M416 M901 M902 M401 M507 M509 D201 D118 D130 B108 B112 B109 S102
7	28	M102 M401 M103 M105 M404 M511 M402 M512 M407 M408 M502 M701 M704 M702 M413 M506 M508 M601 D201 D108 D120 D115 D202 D105 S102 S101 T108 R120
8	20	M101 M108 M501 M401 M102 M502 M702 M301 M703 D201 D125 D140 D203 B105 B108 B109 B112 B110 S102 T125
9	24	M102 M401 M103 M105 M402 M101 M106 M501 M506 M502 M508 M301 M603 M604 M404 M302 D203 D115 D130 S101 T115 B107 B109 B112
10	24	M101 M103 M104 M503 M105 M408 M106 M403 M415 M407 M501 M509 M703 M502 M401 M402 M901 M902 M701 D201 D115 D202 S101 T115
11	15	M112 M401 M104 M412 M405 M402 M410 D201 D130 D202 D116 B101 B109 B105 T116
12	21	M112 M401 M104 M105 M101 M106 M402 M505 M710 M107 M702 M704 M502 D201 D130 D202 D125 B108 B109 R130 S102
13	30	M701 M101 M702 M102 M401 M413 M406 M301 M302 M306 M315 M801 M802 M402 M405 M501 M502 M506 M508 D202 D140 D201 D120 D130 R140 R120 T140 T120 S102 B108
14	27	M102 M401 M103 M105 M508 M402 M509 M415 M412 M101 M701 M702 M403 M405 M406 M601 M602 M603 M604 D202 D125 D201 R125 B102 B103 S102 T125
15	21	M105 M708 M102 M702 M108 M302 M401 M412 M402 M408 M416 M417 M508 D201 D150 B108 B109 D202 D145 B112 B118

machines, with sturdier structure. The light operations of drilling and tapping are done on special CNC drilling machines when necessary. In total, twelve machines of these different types are assumed to be available for allocation to the part families.

The physical dimensions of the problem under consideration can be summarized as follows:

15 parts.

12 Machines

4 Horizontal Spindle Machining Centers

3 Vertical Spindle Machining Centers

3 Boring Centers (Heavy Machining)

2 NC Drilling and Tapping Machines

Three cells

6.1.2 Generation of Problem Input to An IP Routine

The problems are solved using the integer programming routine of the SAS/OR package (Version 5) on an IBM 4381 computer.

The input problem matrix has to be generated through a program for each of the problems, since the problem sizes are too large for manual input.

A series of program modules in Fortran have been written for the generation of the problem matrix in SAS/OR format for different problems listed below.

a. P(R), the parametric objective function problem in the part part family formation.

b. Finding upperbound and the lowerbound for the problem P(R) for varying values of R (to find the interval (L,U)) by the algorithm in Section 4.2.3.3

c. Subproblems of type P(R) to be solved in the iterations of Approximation Procedure.

d. Finding the upperbound and lowerbound for the subproblems P(R) in the Approximation Procedure for varying values of R, to find the interval (L,U).

e. Problem P1, the machine group allocation formulation.

f. Problem INF, for identifying the machines causing infeasibility in the machine group allocation problem.

The program listings are given in Appendix C.

6.2 Part Family Formation - An Example

The solution procedure for the fractional programming formulation of the problem involves a search procedure as indicated in section 4.2. The problem size for the data in Appendix B is as follows.

N=15 and K=3.

Problem size:

# of integer X_{ik} variables	: 30
# of continuous X_{ik} variables	: 15
# of M_{ijk} variables	:315
TOTAL # OF VARIABLES	:360
# of type (i) constraints	: 15
# of type (ii) constraints	:NIL
# of type (iii) constraints	:315
# of type (iv) and (v) constraints	:630
TOTAL # OF CONSTRAINTS	:960

The computations involved in solving the problem are indicated Sections to follow.

6.2.1 Finding the interval (L,U)

The values of LB_R and UB_R were calculated using the algorithm in section 4.2.3.3 for the values of R from 0.05 to 10.50 in steps of 0.05. The partial listing of the values is tabulated in Table 7.

From the table:

L =2.45 ; $LB_{2.45}$ = -5.10 ; $UB_{2.45}$= 1041.59

U =6.10 ; $LB_{6.10}$ = -2272.2 ; $UB_{6.10}$= 0.20

Based on the proof in section 4.2.2 we have,

$$2.45 \leq R^* \leq 6.10$$

The argument about restricting to the lower end of the

TABLE 7

BOUNDS ON THE OBJECTIVE FUNCTION FOR DIFFERENT VALUES OF R

TOTAL NUMBER OF C_{IJ}'s = 315

R	# OF NEGATIVE C_{IJ}'s	UPPER BOUND	LOWER BOUND	
0.05	0	734.45	3076.54	
0.10	0	720.90	3033.09	
0.15	0	707.35	2989.64	
.	.	.	.	
.	.	.	.	
2.45	27	-5.10	1041.59	— L
2.50	33	117.50	1005.00	
.	.	.	.	
.	.	.	.	
6.10	282	-2272.29	0.20	— U
6.15	282	-2313.44	-7.70	
6.20	282	-2354.59	-26.80	
.	.	.	.	
.	.	.	.	
.	.	.	.	
9.95	309	-5536.75	-605.15	
.	.	.	.	
.	.	.	.	

region (L,U) in the search for R^* is evident from the values of LB_R and UB_R in Table 7.

6.2.2 Initial Solution through Approximation Procedure

The total solution time requirement for solving the part family formation problem is dictated by the number of problems P(R) solved during the search. Each of the paroblems P(R) to be solved in this case is of thee size indicated earlier.

Considering the problem size and the solution time required, the importance of starting with an initial solution nearer to the optimal solution is evident.

The approximation procedure as outlined in the section is applied in this case to find an initial solution. The result from a single trial of the approximation procedure with some random starting configuration solution is sufficient to get an initial solution.

As indicated earlier, the requirements of such an approximation procedure are arriving at a 'good' solution (near to optimal) and doing so in a reasonable amount of time (time comparable to the solution time of one problem P(R)). With a view to test the procedure, trials are carried out with different starting configurations. Table 8 provides a summary of these trials. The typical

TABLE 8

Summary of Trials with Different Starting Configurations for Fifteen Parts Example.

#	Starting Config.	Initial ODC	# of Iter	Final Config.	Final ODC
1.	[1,6,7,12,13] [2,5,8,11,14] [3,4,9,10,15]	3.6653	7	[1,3,4,8,15,13] [2,6,11,12] [5,7,9,10,14]	2.8158
2.	[1,6,7,12,13] [2,5,8,11,14] [3,4,9,10,15]	3.6971	8	[1,3,4,8,15,13] [2,6,11,12] [5,7,9,10,14]	2.8158
3.	[1,5,7,9,13,14] [3,6,11,15] [2,4,8,10,12]	3.1573	8	[1,3,4,5,8,13] [6,11,12,15] [2,7,9,10,14]	2.79999
4.	[1,5,8,10,15] [3,9,7,13,14] [2,4,6,11,12]	3.44	5	[1,3,4,8,15,13] [2,6,11,12] [5,7,9,10,14]	2.8158
5	[1,3 9,12,13] [4,5,7,10,11] [2,6,8,14,15]	3.66	7	[1,3,4,5,8,13] [6,11,12,15] [2,7,9,10,14]	2.79999

TABLE 9

Iteration Log for the Approximation Procedure

Number of Parts = 15

Random Starting Partition = 1

ITR No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
0	[1,6,7,12,13] [2,5,8,11,14] [3,4,9,10,14] MF=1 CDC=3.665	* Based on range analysis choose R=3.55				
		3.55	[2,5,8,11,14, 1,12] [3,4,9,10,15, 7,13] [6]	-73	3.349	High, Choose 3.349
		3.349	Same as above	0	3.349	OK
1	[6] [2,5,8,11,14,1, 12] [3,4,9,10,15,7, 13] MF=3 CDC=3.349	* Based on range analysis choose R=3.349				
		3.20	[6,3,15] [1,2,5,8,11, 12,14] [4,7,9,10,13]	0.4	3.201	OK
2	[3,6,15] [1,2,5,8,11,12, 14] [4,7,9,10,13] MF=2 CDC=3.201	* Based on the range analysis choose R = 2.90				
		2.90	[3,6,15,8] [2,11,12] [4,7,9,10,13, 1,5,14]	-1.6	2.895	High Choose 2.895
		2.895	Same as above	0	2.895	OK

(Contd.)

TABLE 9 (continued)

Itr No.	INITIAL ALLOCATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
3	[3,6,8,15] [2,11,12] [1,4,5,9,10,13, 14] MF=3 CDC=2.895	Based on range analysis choose R=2.70				
		2.70	[3,6,8,15, 1,4] [2,11,12] [5,7,9,10, 13,14]	54.9	2.869	Low, Choose 2.869
		2.869	Same as above	-0.01	2.869	OK
4	[3,6,8,15, 1,4] [2,11,12] [5,7,9,10,13, 14] MF= 3 CDC=2.869	* Based on range analysis choose R = 2.85				
		2.85	Same as Initial	6.45	2.869	Low, Choose 2.869
		2.869	Same as initial	-0.01	2.869	OK
		Family configuration not changed				
		Choose the family with next highest D_k. i.e MF=1 R = 2.85				
		2.85	[1,3,4,8, 15] [2,6,11,12] [7,9,10,5,14] 13,1,5,14]	-0.05	2.845	OK

(Contd.)

TABLE 9 (Continued)

Itr No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
5	[1,3,4,8,15] [2,11,12,6] [5,7,9,10,14] MF=3 CDC=2.845	* Based on range analysis choose R=2.845				
		2.845	[1,3,4,8, 13,15] [2,11,12,6] [5,7,9,10, 14]	-9.3	2.815	Hogh, Choose 2.815
		2.815	Same as above above	-0.01	2.815	OK
6	[1,3,4,8,15 13] [2,11,12,6] [5,7,9,10,14] MF=1 CDC=2.815	* Based on range analysis choose R = 2.815				
		2.815	Same as Initial Family configuration has not changed.	0	2.815	OK
		Choose the family with next highest D_k . i.e., MF=3 Based on the range analysis choose R =2.815				
		2.815	Same as initial Family configuration has not changed	0	2.815	OK
		Choose the next family i.e., MF=1 Based on the range analysis, choose R= 2.815				
		2.815	Same as Initial Family configuration is not changed	0	2.815	OK
		All the families are considered. STOP				

iteration log maintained for the first trial is given in Table 9. The iteration logs for other trials are listed Appendix D.

From Table 8. two points are evident:

i) The final CDC's obtained by the procedure are very close to each other (Similar result has been reported in [10]).

ii) The solution obtained is close to the optimal solution(R^* should be in the range 2.45 - 2.7999, since L = 2.45).

The solution times required for different subproblems in each of the iterations are given in Table 10. The total times required are :

Starting Config.	Approximate Total Time
1	2.48 Min
2	8.03 Min
3	2.31 Min
4	3.90 Min
5	1.53 Min

6.2.3 Search Log

The problem is then solved using R = 2.7999 (the value obtained through the approximate solution procedure) as the upper limit on the value of R^*.

i.e., the search range chosen is [2.45,2.799]

First the problem is solved at the midpoint of this range, i.e, at 2.625.

The optimal solution to the problem P(2.625) = 173.25

=> $R^* > 2.625$

Hence, the value 2.7999 is chosen as the R for next P(R).

The problem P(2.7999) gives the optimal value as 0.

=> $R^* = 2.7999$

The solution times required for the above two problems P(R) are:

P(2.625) ----- 21.28 Mins

P(2.799) ----- 22.29 Mins

It also turns out that the optimal solution is obtained in some of the trials of the approximation procedure, and close to optimal solution in other trials. Thus the solution obtained through the approximation procedure is 'good' (in fact optimal in this case. It can not be gauranteed to be optimal, however). Also, the solution time required for the problems P(R) confirms the importance of starting the search at a 'good' solution. If the problem were to be attempted with some other initial value as the upper limit, say with R=3.00, then the number of problems P(R) solved would have been more (each of them taking a time of about 20 Minutes) resulting in a larger solution time.

6.2.4 Some computational considerations

The solution times (CPU) for the series of problems solved in the course of approximation procedure iterations are listed in Table 10 along with the number of integer variables in each of the problems.

Whilst these times should be strictly associated with a specific Package-Computer combination (SAS/OR and IBM 4381), they are indicative of the computational behaviour of these problems viz.,

i) The solution times increase even for a small increase in the number of integer variables.

ii) For the same number of integer variables, different problems require different solution times, sometimes varying widely from each other.

Several problems have been solved using variations of the original data. It is observed that in all the cases the solution procedure converges to the exact solution. And, it appears that for the values of R close to the value of R^*, the optimal allocations obtained by P(R) would also be the optimal allocation corresponding to R^*.

The problems P(R) are solved steps, halting the IP routine intermittently to check the sign of the objective function any intermediate solution that might have been found. If the sign turns out to be negative, problem P(R)

TABLE 10

TYPICAL SOLUTION TIMES FOR THE APPROXIMATION PROCEDURE ITERATIONS

Starting Config #	Iter-ation	Problem No.	# of Integer Variables	Solution time Min : Sec
1	0	1	10	0 : 3.38
		2	10	0 : 3.38
	1	1	14	0 :26.43
	2	1	14	0 : 7.52
	3	1	16	0 :36.27
		2	16	0 :36.27
	4	1	12	0 : 5.10
		2	12	0 : 5.09
		3	12	0 : 4.89
	5	1	12	0 : 4.91
		2	12	0 : 4.89
	6	1	12	0 : 5.20
		2	10	0 : 3.38
		3	8	0 : 2.12
			TOTAL	2.48 Min
2	0	1	12	0 : 4.89
		2	12	0 : 4.79
	1	1	16	0 :36.27
		2	16	0 :36.27
	2	1	20	1 :32.22
		2	20	1 :35.41
		3	20	1 :37.85
	3	1	16	0 :21.77
		2	16	0 :18.80
		3	10	0 : 3.09
		4	10	0 : 3.09
	4	1	16	0 :17.13
		2	16	0 :14.61
	5	1	12	0 : 5.10
		2	12	0 : 5.09
		3	12	0 : 4.89
	6	1	12	0 : 4.91
		2	12	0 : 4.89
	7	1	12	0 : 5.20
		2	10	0 : 3.38
		3	8	0 : 2.12
			TOTAL	8.0295 Min
3	0	1	12	0 : 5.61
		2	12	0 : 5.58
	1	1	16	0 :25.87
		2	16	0 :20.05

TABLE 10(Contd.)

Starting Config	Iteration	Problem No.	# of Integer Variables	Solution time Min : Sec
	2	1	16	0 :16.14
		2	16	0 :26.26
	3	1	12	0 : 5.15
		2	10	0 : 3.38
		3	10	0 : 3.38
	4	1	12	0 : 5.58
		2	10	0 : 3.50
		3	10	0 : 3.50
	5	1	14	0 : 7.75
3		2	14	0 : 7.75
	6	1	12	0 : 5.05
		2	10	0 : 3.50
		3	8	0 : 1.91
			TOTAL	2.314 Min
	0	1	10	0 : 2.98
		2	10	0 : 2.92
	1	1	18	1 :19.61
		2	18	1 :57.18
	2	1	12	0 : 5.04
4		2	10	0 : 3.00
		3	10	0 : 2.81
	3	1	12	0 : 4.91
		2	12	0 : 4.89
	4	1	12	0 : 5.20
		2	10	0 : 3.38
		3	8	0 : 2.12
			TOTAL	3.900 Min
	0	1	10	0 : 3.25
		2	10	0 : 3.38
		3	10	0 : 3.29
	1	1	14	0 : 8.03
		2	14	0 : 8.21
	2	1	12	0 : 6.72
		2	10	0 : 2.88
	3	1	14	0 : 8.14
		2	14	0 : 8.69
5		3	10	0 : 3.03
		4	10	0 : 2.90
	4	1	14	0 : 7.45
		2	14	0 : 7.19
	5	1	12	0 : 5.28
		2	10	0 : 2.85
	6	1	12	0 : 3.05
		2	10	0 : 3.50
		3	8	0 : 1.91
			TOTAL	1.529 Min

is terminated since the optimal solution must be negative. This approach is helpful in cutting down the actual time required to solve the problem. For this reason, intutively it is better to approach the value of R from the negative side.

The solutions obtained by solving the problems P(R) can be used to get a short-cut in search procedure compared to binary principle. At each value of R, at least one feasible solution is found before diciding on the next course of action. Each feasible solution gives a particular allocation of parts to the families. For this allocation, the value of A(X)/B(X) can be calculated. The value of R^* should be less than or equal to the calculated value for this feasible solution. The calculated value A(X)/B(X) thus is helpful in the search for R^*.

6.3 Machine Group Allocation- An Example

The machine group allocation problem is illustrated by an example. The infeasibility occuring in the problem is resolved using the formulation in Section 5.3.2.

6.3.1 Routing Information

The routing information in the form of matrix A_{jk} as described in section 5.1.3 is given in Table 11.

this type of information about the feasibility of certain operations on the available machines is assumed to be available.

6.3.2 Solution Procedure

The machine group allocation problem has been solved for the part families formed with fifteen parts.

The family configuration obtained by the part grouping in Section 6.2 is given below:

Family 1 : parts 1, 3, 4, 5, 8, 13

Family 2 : parts 6, 11, 12, 15

Family 3 : part 2, 7, 9, 10, 14

From the routing matrix for these parts (Table 11) a list of multiplication terms for each family is generated as explained in Section 5.1.4 . For example, the operation 1 and 2 of the part 6 can be done on machines 1 and 4 respectively, and the product term $I_{12} \cdot I_{42}$ represents this routing. The sample of such a list is given in Table 12.

The size of the problem is :

M=12 K=3

- Number of integer Variables = 12 x 2 = 24
- Number of Free variables = 12
- Number of product term variables representing the routings for the parts =117

TABLE 11

MACHINE ROUTING DATA

Part No.	# Of Opns	Opn #	H1 1	H2 2	H3 3	H4 4	V1 5	V2 6	V3 7	B1 8	B2 9	B3 10	D1 11	D2 12	Total
1	3	1	0	0	0	0	1	0	1	0	0	0	1	0	3
		2	1	0	0	0	0	0	0	0	1	0	0	0	2
		3	0	0	1	0	0	0	0	0	0	0	1	0	2
2	2	1	1	0	0	1	0	1	0	0	0	0	0	0	3
		2	1	0	0	0	1	0	0	0	0	1	0	0	3
3	2	1	0	0	0	0	1	1	0	1	0	1	0	0	4
		2	0	0	0	0	0	1	0	0	0	1	0	1	3
4	3	1	0	0	0	0	0	0	1	0	1	0	0	0	2
		2	0	1	0	0	1	0	0	0	0	0	0	1	3
		3	0	0	0	1	0	1	0	0	0	0	0	0	2
5	3	1	0	1	0	0	0	1	0	0	0	0	0	0	2
		2	0	0	1	0	0	1	0	0	0	0	0	0	2
		3	0	0	0	0	0	1	0	0	0	0	0	0	1
6	2	1	1	0	0	0	0	0	0	0	0	0	1	0	2
		2	0	0	0	1	1	1	0	0	1	0	0	0	4
7	4	1	0	1	0	0	0	0	0	0	0	1	0	0	3
		2	0	0	0	0	0	0	1	0	0	0	0	1	2
		3	0	0	0	1	0	0	1	0	0	0	0	0	2
		4	0	0	0	0	0	0	1	0	0	0	0	0	1
8	3	1	1	0	1	0	0	0	0	0	0	0	0	0	2
		2	1	0	0	0	0	0	0	1	0	0	0	0	2
		3	0	1	0	0	0	0	0	0	1	0	0	0	2
9	3	1	0	1	0	0	0	1	0	0	0	1	0	0	3
		2	0	1	0	0	0	1	0	0	0	0	0	0	2
		3	0	0	1	0	0	0	0	0	0	1	0	0	2
10	2	1	0	1	0	0	0	1	0	0	0	1	0	1	4
		2	0	0	0	1	0	0	1	0	0	0	0	0	2
11	3	1	1	0	1	0	0	0	0	0	0	0	0	0	2
		2	0	0	0	1	0	0	0	0	1	0	0	0	2
		3	1	0	0	0	0	0	0	0	1	0	0	0	2
12	3	1	1	0	0	1	0	0	0	0	0	0	0	0	2
		2	0	0	0	1	0	0	0	0	0	0	0	0	2
		3	1	0	0	0	0	0	0	0	1	0	0	0	2
13	3	1	0	1	0	0	1	0	0	0	0	0	0	0	2
		2	0	0	0	0	0	0	0	1	0	0	0	0	2
		3	0	0	1	0	0	0	0	0	1	0	0	0	2
14	4	1	0	0	0	0	0	1	1	0	0	0	0	0	2
		2	0	0	0	0	0	1	0	0	0	0	0	0	1
		3	0	0	1	0	0	0	1	0	0	1	0	0	3
		4	0	1	0	1	0	0	0	0	0	0	0	0	2
15	3	1	1	0	1	0	0	0	0	0	0	0	0	1	2
		2	0	1	0	0	1	0	0	0	0	0	0	0	2
		3	0	0	1	0	0	0	0	0	0	0	1	0	2

TABLE 12

LIST OF PRODUCT TERMS INDICATING THE ROUTINGS FOR PARTS PART FAMILY 2

No.	Product Term	Number of routings represented for the part number			
		6	11	12	15
1	$I_{12} \cdot I_{42}$	1	1	2	0
2	$I_{12} \cdot I_{52}$	1	0	0	0
3	$I_{12} \cdot I_{62}$	1	0	0	0
4	$I_{12} \cdot I_{92}$	1	2	0	0
5	$I_{112} \cdot I_{42}$	1	0	0	0
6	$I_{112} \cdot I_{52}$	1	0	0	0
7	$I_{112} \cdot I_{62}$	1	0	0	0
8	$I_{112} \cdot I_{92}$	1	0	0	0
9	$I_{12} \cdot I_{42} \cdot I_{92}$	0	1	1	0
10	$I_{32} \cdot I_{42} \cdot I_{12}$	0	1	0	0
11	$I_{32} \cdot I_{42} \cdot I_{92}$	0	1	0	0
12	$I_{32} \cdot I_{92} \cdot I_{12}$	0	1	0	0
13	$I_{32} \cdot I_{92}$	0	1	0	0
14	$I_{12} \cdot I_{122}$	0	0	1	0
15	$I_{12} \cdot I_{122} \cdot I_{92}$	0	0	1	0
16	$I_{42} \cdot I_{92}$	0	0	1	0
17	$I_{42} \cdot I_{122} \cdot I_{12}$	0	0	1	0
18	$I_{42} \cdot I_{122} \cdot I_{92}$	0	0	1	0
19	$I_{12} \cdot I_{22} \cdot I_{32}$	0	0	0	1
20	$I_{12} \cdot I_{22} \cdot I_{112}$	0	0	0	1
21	$I_{12} \cdot I_{52} \cdot I_{32}$	0	0	0	1
22	$I_{12} \cdot I_{52} \cdot I_{112}$	0	0	0	1
23	$I_{32} \cdot I_{22}$	0	0	0	1
24	$I_{32} \cdot I_{22} \cdot I_{112}$	0	0	0	1
25	$I_{32} \cdot I_{52}$	0	0	0	1
26	$I_{32} \cdot I_{52} \cdot I_{112}$	0	0	0	1

The problem is infeasible with the constraint of allocating one machine to one family only. The mathematical model developed in Section 5.3.2 is applied to find out the machine causing infeasibility. The value of the optional weightage D is chosen to be 1. The optimal value of the objective function is 2. Two of the machines are identified to be the ones causing infeasibility.

Machine 2 is absolutely essential for families 2 1nd 3. Machine 6 is absolutely essential for families 1 and 3. The allocation constraint for these two machines was relaxed by giving a RHS value of 2 in the corresponding constraints of type (27). Problem P1 is then solved to find the optimal solution to the allocation problem. The optimal machine group allocation(after the relaxation for machines 2 and 3) to the families is given below.

Maximum number of routings acchieved = 38

Machines allotted to family 1 : 3,5,6,8,9,12

Machines allotted to family 2 : 2,1,4,11

Machines allotted to family 3 : 2,6,7,10,

6.4 Discussion of Results

The cell formation is an initial specification problem in the pre-production planning stage. This research presents new formulations for this problem.

The examples illustrating the formulations involve grouping of fifteen parts into three families and then allocating twelve machines to these families. The problems are solved on an IBM 4381 computer using the integer programming (IP)routine of SAS/OR package.

The part family formation example for the above data has 15 . 3 = 45 decision variables. In the input to the IP routine 15 . 2 = 30 variables are explicitly stated to be of 0-1 integer type. The remaining 15 are forced to take 0-1 values due to constraints of type (i) in the formulation. The solution time for the problems solved during the search procedure (Problem P(R)) is about 20-25 Minutes.

The solution times for several problems of smaller size are listed in Table 10.

An example of twentyfive parts using variations in the data given in Appendix B has been attempted. The solution time for the problems solved during the search procedure has been found to be in excess of 150 Min. The experimentation with larger problems has been limited. However, based on the results it appears that with a comparable computer-package combination, problems with 40-50 variables could be solved in a 'reasonable' time. In physical terms the corresponding problem size is 20-25 parts to be grouped into 2-3 families.

Further computational experimentation is necessary for the larger problems considering the following issues.

i) Branch and Bound Strategy: Several search heuristic options are available for the rules of selecting branching nodes and branching variable at the nodes. The experience with the smaller problems can not be directly extrapolated to larger problems, since the strategy that works well for a particular problem size may not work well as the problem size increases [29].

ii) Linearization Strategy: Some variations involving reductions in the number of extra constraints generated have been suggested by Glover and Woolsey [15]. A brief discussion of these is provided by Stecke [30]. The experimentation with these different strategies could be a possibility for the larger problems. (The program written to generate the input for the problem P(R) provides options for implementing different strategies as indicated in [30]).

The bounds established on the objective function could be used to determine the maximum possible variation of the objective function value for any feasible solution from the optimal value. This would be useful especially for large problems.

The approximate solution procedure developed is an extension of the 'single move' heuristic to a 'multiple move' case. The results from the procedure have been found to be near-optimal in the problems solved. Also, starting with different configurations the procedure converged to solutions with objective function values very close to each other. This result is comparable to the 'single move' implementation of the heuristic in [12].

The solution obtained through the approximation procedure provides an upper limit on the objective function value.

The formulation of machine group allocation has 12 . 3 = 36 decision variables. In the input to the IP routine 12 . 2 = 24 variables are explicitly stated to be of 0-1 type. The remaining 12 are forced to take 0-1 values due to the assignment constraints. The solution time for the problem is about 2-3 Min. The formulation for identifying the machines causing infeasibility in the problem required less than 1 Min.

The implementation of the formulations gives the system specification in terms of the cell configuration and the parts manufactured in the cells. The contributions of this research are :

- Defining a dissimilarity coefficient as the objective function of the part grouping formulation.
- Developing an algorithm for finding the bounds on the above objective function.

- Extension of a clustering method of ´single move´ type to a ´multiple move´ case.
- Consideration of the availability of alternative machines and the routing diversity in machine grouping problem.
- Developing a mathemativcal model to identify the machines causing infeasibility in the machine group allocation problem.

The computational aspects for larger problems have to be further tested for larger problems as explained earlier. Also a matter of consideration could be to impose other constraints on grouping. For example, one of the constraints could be to consider the quantities of the parts to be manufactured with a view to balance the work load in the cells.

Some of the direct consequences of grouping are reduction in the work in progress, reduced lead times and reduced scheduling complexity. A study by Purcheck [26] confirms that the cellular systems have better ´operating charecterestics´ as measured by these factors. The reduced work in progress and reduced lead times result in financial gains. An approach for analysing such gains due to grouped system has been suggested by Boucher [3]. Such an analysis could be done after the cell formation problem has been solved using the formulations presented.

CHAPTER VII

SUMMARY

This research deals with the initial specification decisions in the pre-production planning stage for Flexible Manufacturing Systems. The problems of part family formation and machine group allocation have been formulated as 0-1 integer programming models.

The formulation of part family formation is a fractional program. The dissimilarity between the parts in terms of processing requirements has been represented by a coefficient and is defined as a function of 0-1 variables. By identifying the specific nature of the objective function a general search principle has been suitably adopted for solving the formulation.

As a method for providing a starting solution to the search procedure, an extension to a clustering principle reported in the literature has been developed. This extension is based on the fractional model.

The concept of routing flexibility, or the number of available routes for the parts within the cellular system has been adopted in the machine group allocation.

This aspect has not been considered by the Group Technology researchers in conventional systems.

The formulations have been applied to a set of realistic problem data. Several problems have been solved.

The computational experience with these problems indicates that the formulations are applicable to FMS installations manufacturing low or medium variety of parts.

As indicated in Chapter III, many of the systems which operate in tandem with conventional facilities have been, in general, used for the manufacture of critical, high value parts. Many of the FMSs reported in the literature fall into this category. The proposed procedure is applicable for these systems.

The solution procedure developed for the fractional programming model is also applicable in other clustering applications where the pairwise ratio criteria could be used.

In summary, the main contributions of this thesis are the development of a new formulation for part family formation, extension of a heuristic procedure in clustering and adopting the availability of alternative routings for the parts as the criterion in machine grouping.

REFERENCES

1. Arthanari, T.S., and Dodge, Y., Mathematical Programming in Statistics, John Wiley, NY. (1981).

2. Bilalis, N.G., and Mamalis, A.G., The Flexible Manufacturing Systems (FMS) in Metal Removal Processing: An Overview, J. Applied Metal Working, 3, 400-409 (1985).

3 Boucher, T.O., and Muckstadt, J.A., Cost Estimating Methods for Evaluating the Conversion from a Functional Manufacturing Layout to Group Technology, AIIE Trans., 17,268-275 (1985).

4. Burbidge, J.L., Group Technology in the Engineering Industry, Mechanical Engineering Publications Ltd., London (1975).

5. Buzacott, J.A., and Shantikumar, J.G., Models for Understanding Flexible Manufacturing Systems, AIIE Trans., 12, 339-349 (1980).

6. Chakravarthy, A.K., and Shtub, A., Selecting Parts and Loading Flexible MAnufacturing Systems, Proceedings of First Operations Research Society of America/The Institute Of Management Sciences Special Interest Conference on Flexible Manufacturing Systems, Ann Arbor, Michigan, 284-289 (1984).

7. Chan, H.M., and Milner, D.A., Direct Clustering Algorithm for Group Formation in Cellular Manufacture, Journal of Manufacturing Systems, 1, 65-75 (1982).

8. Chandrasekharan, M. P., and Rajagopalan, R., An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing, Int. J. Prod. Res., 24, 451-464 (1986).

9. Chang, T.C., and Wysk, R.A., An Introduction to Automated Process Planning Systems, Prentice-Hall, New Jersey (1985).

10. Chatterjee, A., Cohen, M.A., Maxwell, W.J., Miller, L.W., Manufacturing Flexibility: Models and Measurements, Proceedings of First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems, Ann Arbor, MI, 49-64 (1984).

11. Dinkelbach, W., On Nonlinear Fractional Programming, Management Science, 13, 492-498 (1967).

12. Dutta, S.P., Lashkari, R.S., Nadoli, G., and Ravi, T., A Heuristic Procedure for Determining Manufacturing Families from Design Based Grouping for Flexible Manufacturing Systems, To appear in Computers and Industrial Engineering (1986).

13. Friedman, H.P., and Rubin, J., On some invariant criteria for grouping data, J. Am. Statist. Ass., 62, 1159-1178 (1967).

14. Glover, F., and Woolsey, Converting the 0-1 Polynomial Problem to a 0-1 linear program, Operations Research, 22, 180-182 (1984).

15. __________., and ________, Further Reduction of Zero-One Polynomial Programming Problems to Zero-One Linear Programming Problems, Opns. Res., 21, 156-161 (1973)

16. Gongavare, T.A., and Ham, I., Cluster Analysis applications for group technology manufacturing systems, Cited in Hyer, N. L(ed.)., Group Technology at Work, Society Of Manufacturing Engineers, Dearborn, MI. (1984).

17. Green, T.J., and Sadowski, R.P., Cellular Manufacturing Control, Journal Of Manufacturing Systems, 2, 137-145 (1983).

18 Hartley, J., FMS at Work, IFS (Publications Ltd.,) U.K., (1984).

19. Jagannathan, R., On Some Properties of Programming Problems in Parametric Form Pertaining to Fractional Programming, Management Science, 12, 609-615 (1966).

20. King, J.R., and Nakornchai, V., Machine component group formation in group technology:Review and extention, Int. J. Prod. Res, 20, 117-133 (1982).

21. Kumar, K.R., Kusiak, A., and Vannelli, A., Grouping of Parts and Components in Flexible Manufacturing Systems, European Journal of Operations Research, 24, 387-397 (1986).

22. Kusiak, A., Flexible Manufacturing Systems: A Structural approach, Int. J. Prod. Res., 23, 1057-1073 (1985).

23. __________., The Part Families Problem In Flexible Manufacturing Systems, Proceedings of First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems, Ann Arbor, MI, 237-242 (1984).

24. McAuley, J., Machine grouping for efficient production, Prod. Engr., 51, 53-57 (1972).

25. Purcheck, G.F.K., Machine Component Group Formation: An Heuristic Method for Flexible Production Cells and Flexible Manufacturing Systems, Int. J. Prod. Res., 23, 911-943 (1985).

26. _______________., Computer Aided Organization for Manufacture, Int. J. Prod. Res., 23, 887-910 (1985).

27. Rajagopalan and Batra, J.L., Design of cellular production systems: A graph-theoritic approach, Int. J. Prod. Res., 13, 567-579 (1975).

28. Ranky, P.G., The Design and Opeartion Of FMS, IFS (Publications) Ltd., North Holland Publishing Company, U.K (1984)

29. SAS Institute Inc., SAS/OR^{TM} User's Guide, Version 5 Edition. Cary, NC: SAS Institute Inc., pp 347 (1985)

30. Stecke, K.E., Production Planning for Flexible Manufacturing Systems, Ph.D thesis, Purdue University, University Microfilms International, (1981).

31. _____________., Loading and Control Policies for a Flexible Manufacturing System, Int. J. Prod. Res., 19, 481-490 (1981).

32. Swarup, K., Linear Fractional Functionals Programming, Operations Research, 13, 1029-1036 (1965).

33. Vannelli, A., and Ravikumar, K., A method for finding minimal bottle-neck cells for grouping part-machine families, Int. J. Prod. Res., 24, 387-400 (1986)..

34. Wilhelm, W.E., and Sarin, S.C., Models for the Design of Flexible Manufacturing Systems, Proceedings of Institute Of Industrial Engineers Annual Conference, Atlanta, (1985).

35. Wagner, H.M., and Yuan S.C.J., Algorithmic Equivalance in Linear Fractional Programming, Management Sciences, 14, 301-106 (1968).

APPENDIX A

ALTERNATIVE DEFINITIONS FOR THE OVERALL DISSIMILARITY COEFFICIENT

ALTERNATIVE DEFINITIONS FOR THE OVERALL DISSIMILARITY COEFFICIENT

The overall dissimilarity coefficient defined as the objective function in the part family formation problem is an average of all the pairwise dissimilarity coefficients. The overall measure also can also be defined as follows:

a) $$CDC_1 = \frac{1}{N.(N-1)} \cdot \left[\sum_{k=1}^{K} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij}/s_{ij}) \cdot X_{ik} \cdot X_{jk} \right]$$

b) $$CDC_2 = \frac{1}{K} \cdot \left[\sum_{k=1}^{K} \left[\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \cdot X_{ik} \cdot X_{jk}}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} \cdot X_{ik} \cdot X_{jk}} \right] \right]$$

c) $$CDC_3 = \frac{1}{K} \cdot \left[\sum_{k=1}^{K} \left[\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij}/s_{ij}) \cdot X_{ik} \cdot X_{jk}}{n_k} \right] \right]$$

where,

$$n_k = \sum_{i=1}^{N} X_{ik}$$

CDC_1, CDC_2 and CDC_3 are the other possible definitions of the objective of the minimization of dissimilarities and maximization of similarities between the parts.

A formulation with CDC_1 as the objective function would be similar to the problem P(R).

The formulations CDC_2 and CDC_3 as objective functions would be more complicated to solve. Both these functions can be simplified into a single ratio of non-linear integer functions. The difference would be that in these formulations, the functions would have polynomial terms of higher degree (unlike the formulation for CDC which has only the product terms of degree two). Hence, the method developed for establishing the region (L,U) would not be applicable to these formulations. The general search principle however, still holds in these cases.

The reasons for adopting CDC as the objective function are:

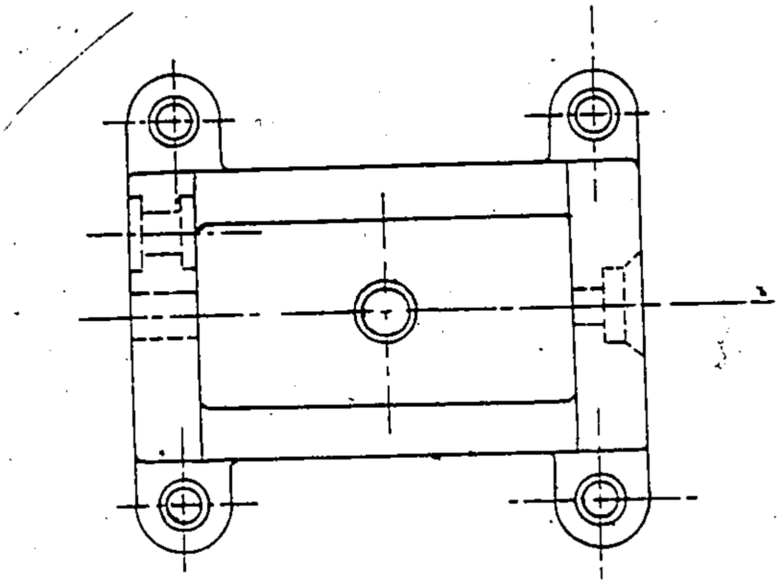
- Using as objective function a similar expression as reported in [12].
- The expression CDC incorporates a weighted average of all d_{ij}/s_{ij} values.

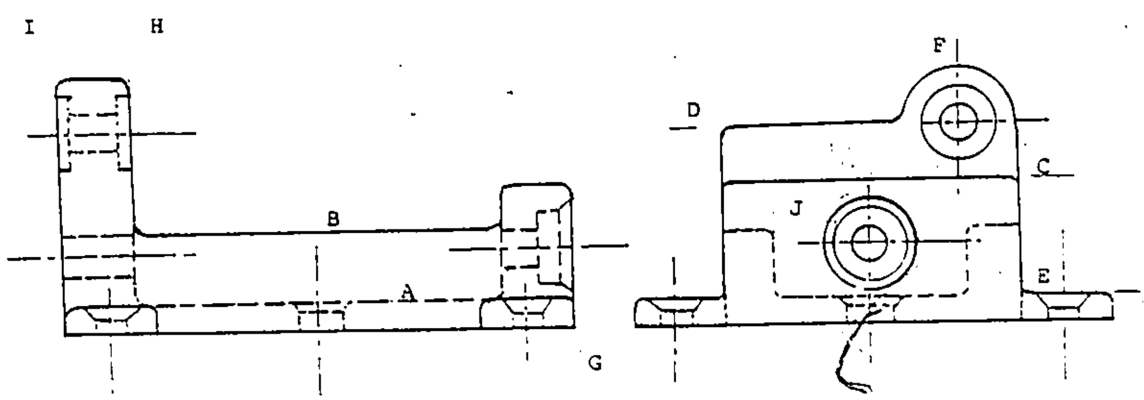
APPENDIX B

PART SKETCHES AND PROCESS DETAILS

PART #1 HOUSING

Scale 1:6





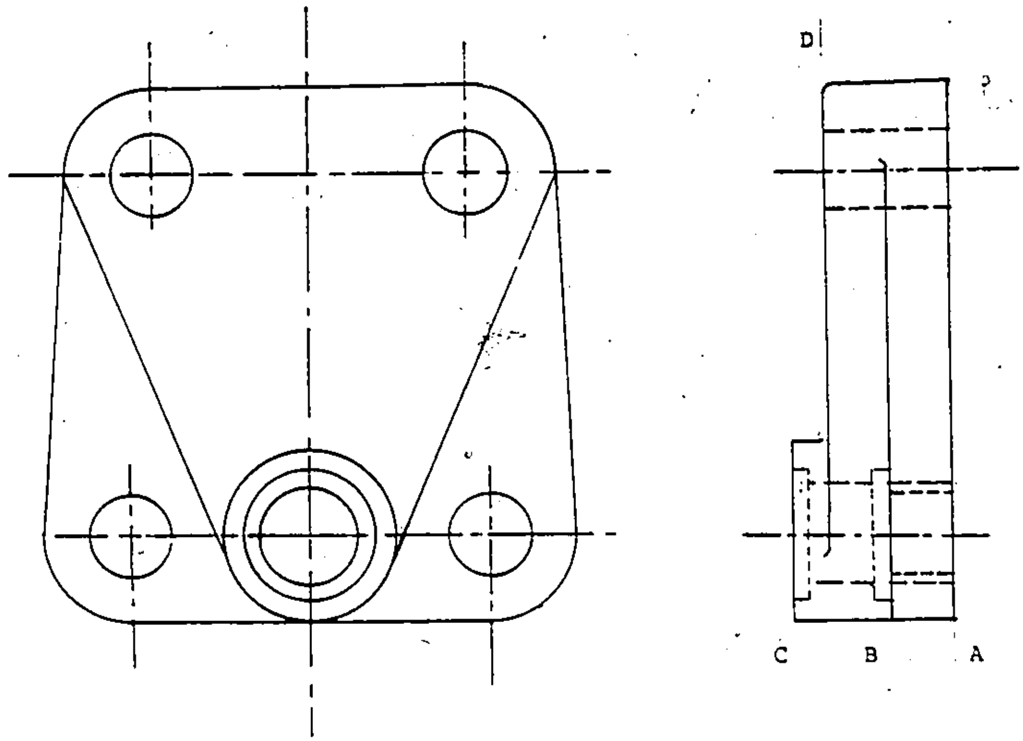
PROCESS DETAILS

PART NO. : 1 PART NAME: HOUSING

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Surface (A)	M501
	Finish Mill Surface (A)	M502,M701
	Center Hole (1)	D201
	Drill 42 Dia Hole Thro'	D142
	Ream 42 Dia Hole Thro'	R142
	Chamfer	D109,M602
	Rough Mill Edge (B)	M101,M401
	Rough Mill Edge (C)	M101,M401
	Rough Mill Edge (D)	M101,M401
(2)	Rough Mill Base Projections	M102
	Finish Mill Base Projections	M103
	Center Holes (4)	D202
	Drill 30 Dia Holes Thro'(4)	D130
	Ream 30 Dia Holes Thro' (4)	R130
	Chamfer (4)	D109
	Peripheral Mill Surface (J)	M301,M101
	Finish Mill Surface (J)	M302,M102
	Rough Mill Face (G)	M301,M102
	Finish Mill Face (G)	M301,M102
	Bore 42 Dia Hole Thro'	B108
	Finish Bore 42 Dia Hole Thro'	B109
	Counter Bore 72 Dia 36 Deep	B101
	Chamfer	M702
(3)	Contour Mill (F)	M603,M108
	Face Mill Surface (H)	M503
	Finish Mill Surface (H)	M504
	Rough Mill Surface (I)	M603,M108
	Finish Mill Surface (I)	M503
	Bore 48 Dia Holes Thro' (2)	B108
	Finish Bore 48 Dia Holes Thro' (2)	B115
	Ream 48 Dia Holes Thro' (2)	R148
	Counter Bore 78 Dia Inside	B101
	Counter Bore 78 Dia Outside	B106

PART #2 BASE BLOCK

Scale 1:2



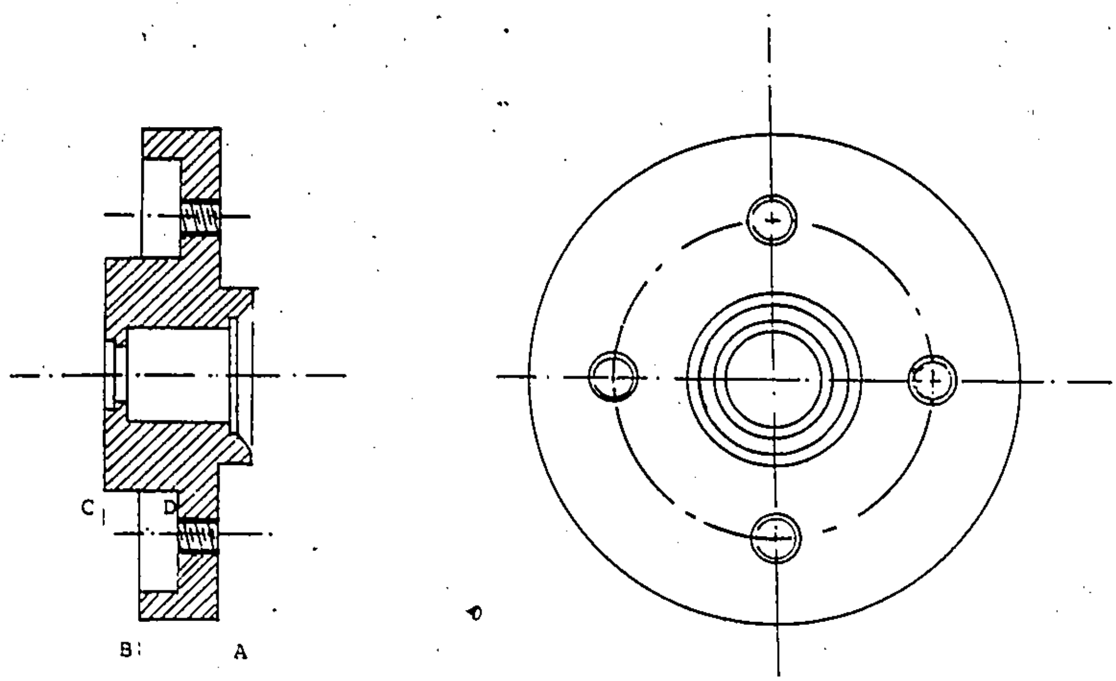
PROCESS DETAILS

PART NO. : 2 PART NAME: BASE BLOCK

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Face (A)	M101
	Finish Mill Bottom Face (A)	M103
	Rough Mill Sides	M104
	Finish Mill Sides	M105
(2)	Mill Contours (4)	M901,M602
	Face Mill Surface (D)	M501
	Finish Mill surface (D)	M501
	Face Mill Surface (B)	M506,M508
	Finish Mill Surface (B)	M506,M508
	Center Holes (4)	D201
	Drill Dia 28 Holes Thro' (4)	D125
	Ream Dia 28 Holes Thro' (4)	R128
	Deburr	S104
	Face Mill Boss	M101,M402
	Rough Bore 34 Dia Hole Thro'	B101
	Finish Bore 34 Dia Hole Thro'	B109
	Step Bore Outside Step (Rough)	B105
	Step Bore Inside Step (Rough)	B105
	Finish Bore Outside Step	B104
	Finish Bore Inside Step	B104

PART #3 PULLEY BLOCK

Scale 1:5



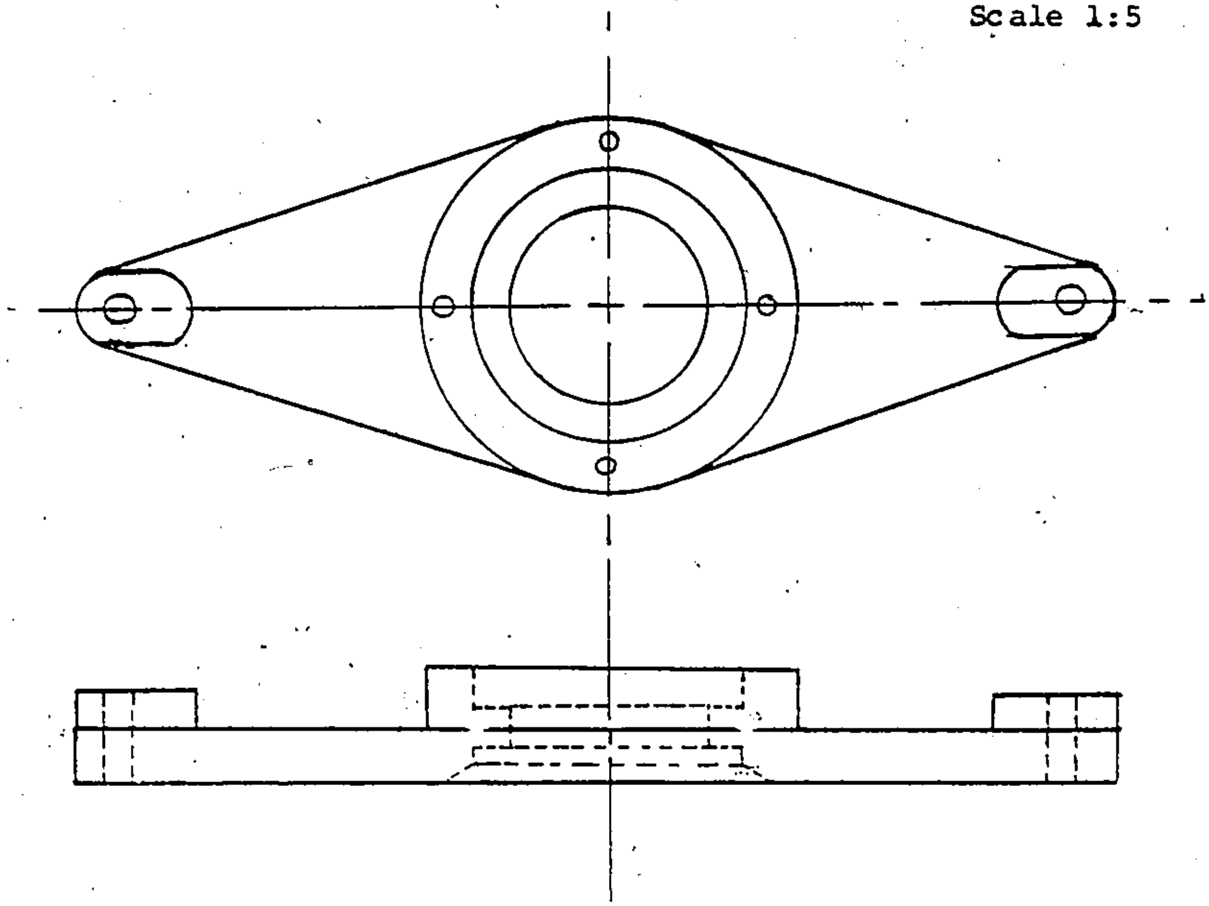
PROCESS DETAILS

PART NO. 3 PART NAME: PULLEY BLOCK

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Face Mill Periphery (B)	M403
	Finish Mill Periphery (B)	M404
	Rough Mill Face (C)	M701
	Finish Mill Face (C)	M702
	Rough Mill Grove (D)	M401
	Mill Peripheral Edges	M412
(2)	Bore 50 Dia Hole Thro´	B108
	Finish Bore 50 Dia Thro´	B109
	Counter Bore 35 Dia 8 Deep	B101
	Finish Counter Bore	B102
	Enlarge Bore Dia Thro´ from step	B115
	Rough Mill Side A	M405
	Finish Mill Side A	M406
	Center Holes (4)	D203
	Drill Holes 30 Dia Thro´(4)	D128
	Ream Holes 30 Dia Thro´ (4)	R130
	Tap Holes 30 Dia Thro´	T130
	Finish Bore 80 Dia	B106
	Counter Bore Dia 105	B112
	Chamfer edge	M702,M712
	Rough Mill Periphery	M101
	Rough Mill Outside Periphery	M102

PART #4 FLANGE

Scale 1:5



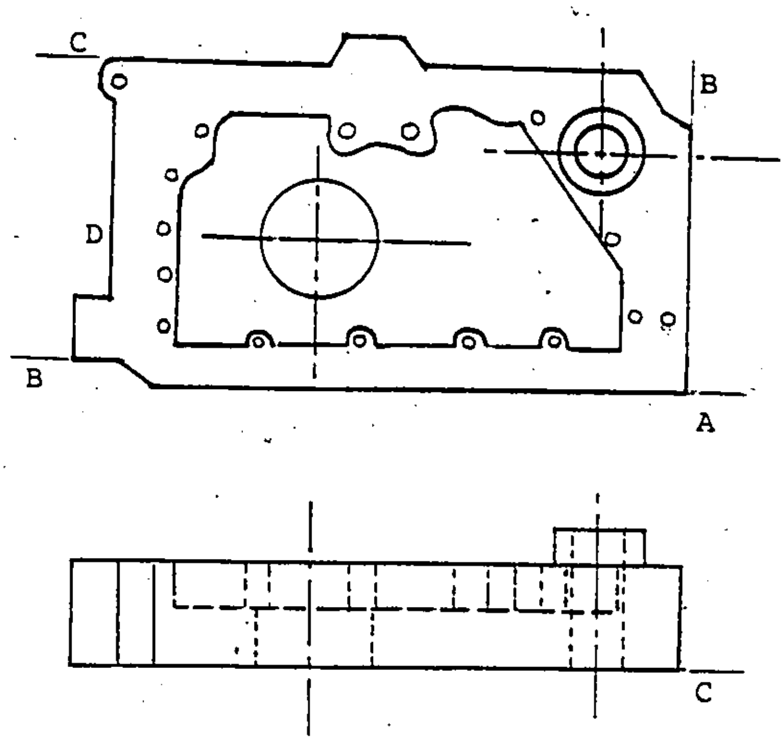
PROCESS DETAILS

PART NO. : 4 PART NAME: FLANGE

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
	Rough Mill Bottom Face	M501,M518
	Finish Mill Bottom Face	M502
	Center Hole (1)	D202
	Drill 50 Dia Hole Thro´	D150
(1)	Rough Bore 110 Dia Hole Thro´	B108
	Finish Bore 110 Dia Hole Thro´	B112
	Counter Bore 175 Dia Deep	B112
	Finish 175 Dia Bore Deep	B112
	Chamfer	M701,M712
	Rough Mill Sides	M211,M101
	Finish Mill Sides	M212,M102
	Face Mill Top Surface	M501,M502
	Counter Bore 175 Dia Deep	B112
	Finish Bore 175 Dia Deep	B112
	Center Holes (4)	D202
(2)	Drill 20 Dia Holes Thro´ (4)	D120
	Deburr	S109
	Tap 20 Dia Holes Thro´ (4)	T120
	Face Mill Bosses	M503
	Finish Mill Bosses	M503
	Center Holes (2)	D201
	Drill 20 Dia Holes Thro´ (2)	D120
	Mill Periphery of Projections	M301
(3)	Finish Periphery of Projections	M302
	Face Mill Top Surface(Rough)	M501
	Mill Periphery Of Bosses	M401,M405

PART #5 . BASE

Scale 1:8



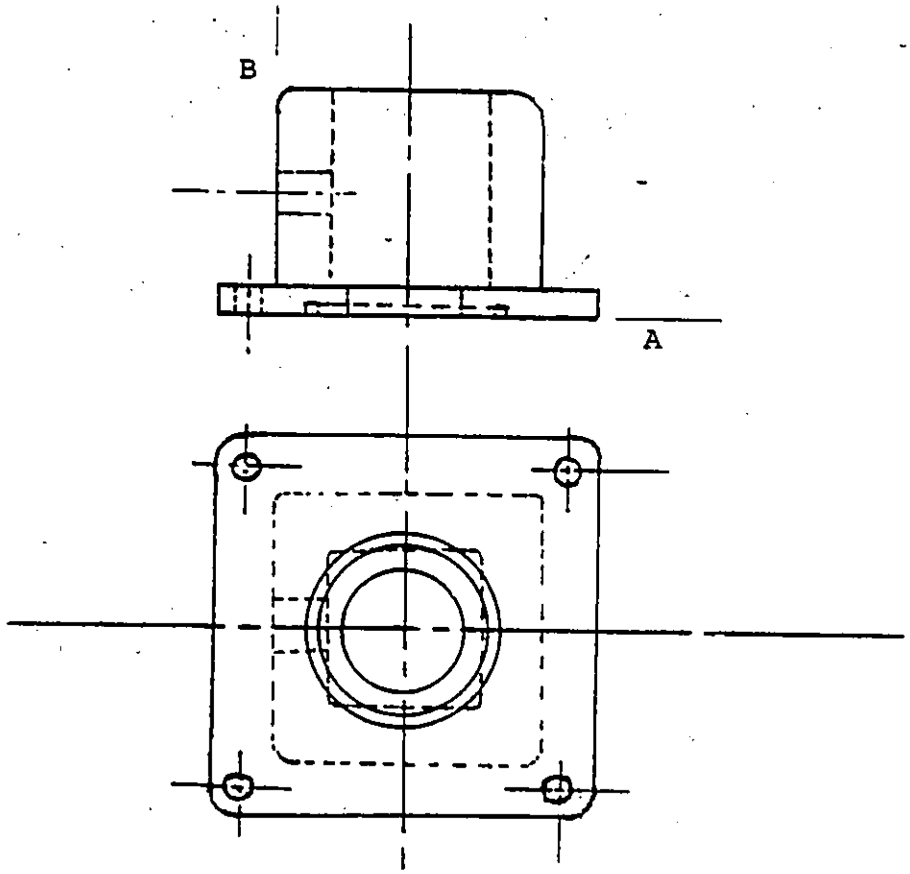
PROCESS DETAILS

PART NO : 5 PART NAME : BASE

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
	Face Mill Side (A)	M102,M401
	Finish Mill Side (A)	M103,M105
(1)	Angle Mill Side (B)	M602
	Finish Mill Side (B)	M603
(2)	Rough Mill Bottom Face (C)	M501,M513
	Finish Mill Bottom Face(C)	M503,M502
	Face Mill Top Edge	M101,M701
	Finish Mill Top Edge	M103,M702
	Peripheral Mill Boss	M401
	Face Mill Boss	M401
	Center Holes (15)	D201
	Drill 8 Dia Holes (15)	D108
	Deburr	S101
(3)	Ream 8 Dia Holes (15)	R108
	Center Hole (1)	D201
	Drill 50 Dia Hole Thro'	D150
	Bore 56 Dia Hole	B106
	Finish Bore 56 Dia Hole	B109
	End Mill Pocket (rough)	M501,M402
	Finish Mill Pocket	M508,M402
	Peripheral Mill Sides	M406
	Contour Mill Projections	M610,M602
	Center Hole (1)	D203
	Drill 50 Dia Hole (1)	D150
	Drill 70 Dia Hole	D170
	Bore 120 Dia Hole	B108
	Rough Mill Sides B,C and D	M301,M305

PART #6 CAP

Scale 1:6



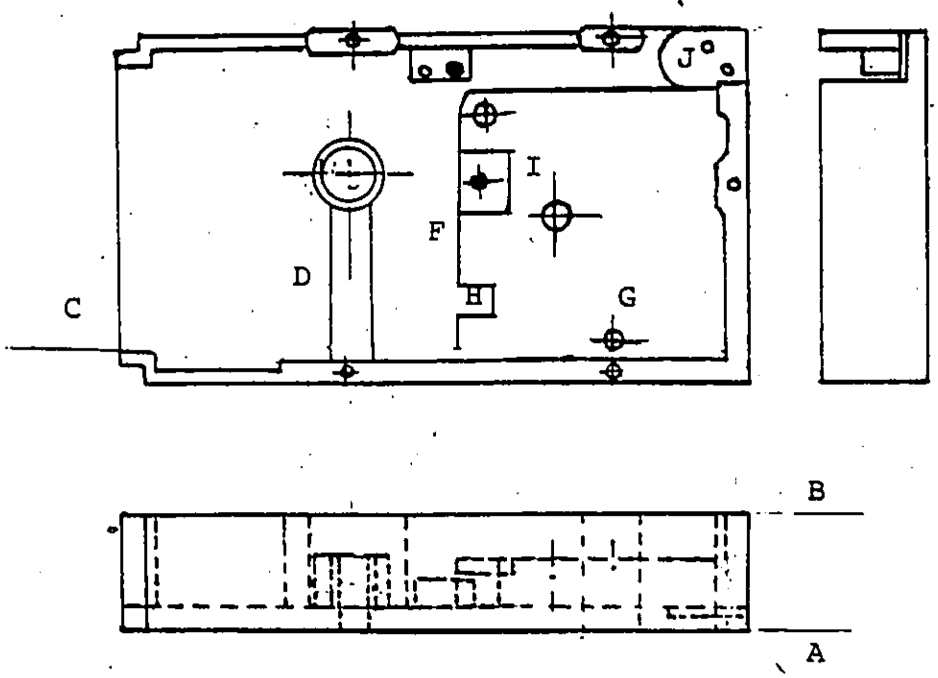
PROCESS DETAILS

PART NO. : 6 PART NAME: CAP

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Face (A)	M501, M504
	Finish Mill Bottom Face (A)	M502
	End Mill Grove	M415, M412
	Finish Mill Grove	M416
	Center Holes (4)	D201
	Drill 18 Dia Holes Thro' (4)	D118
	Deburr	S102
(2)	Rough Mill Top Face	M501
	Finish Mill Top Face	M504
	Bore 96 Dia Hole	B108,B112
	Finish Bore 96 Dia Hole	B109
	End Mill Base Edges	M401,M412
	Rough Mill Face C	M507
	Finish Mill Face c	M509
	Center Hole	D201
	Drill Dia 30 Hole Thro' Wall	D130

PART #7 **PANEL SIDE COVER**

Scale 1:5



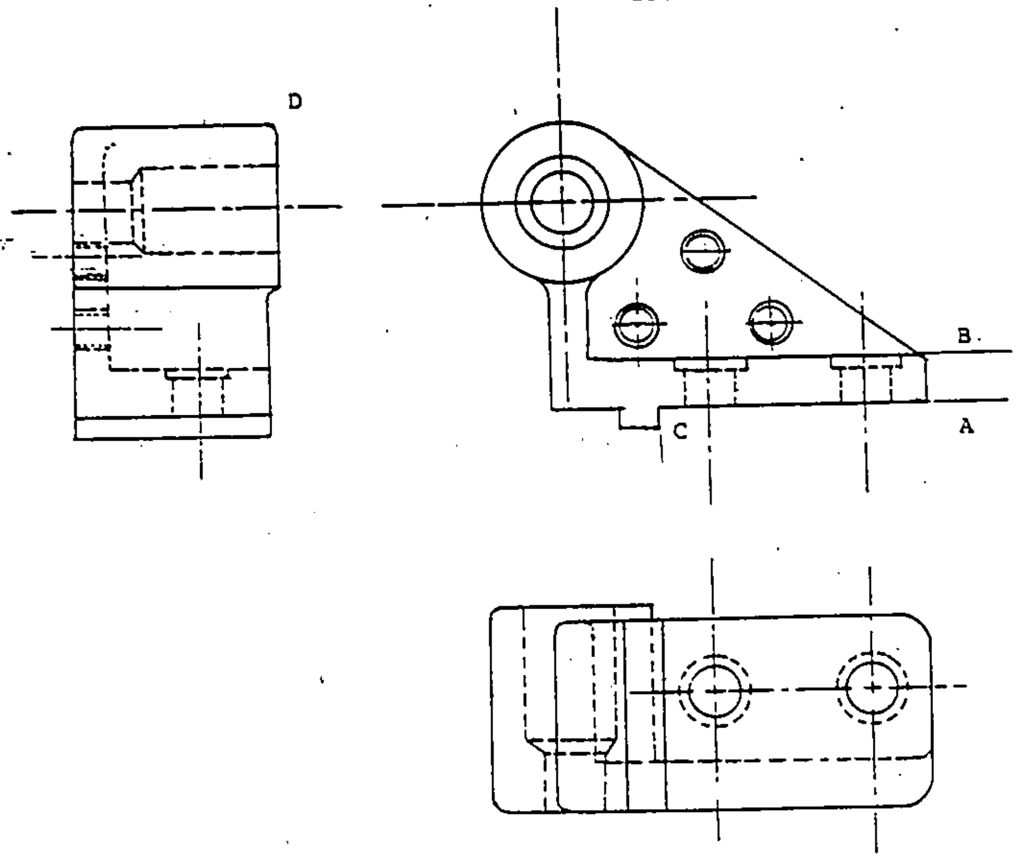
PROCESS DETAILS

PART NO. : 7 PART NAME: PANEL SIDE COVER

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Face (A)	M102,M401
	Finish Mill Bottom Face (A)	M103,M105
	Rough Mill Top Edges (B)	M404,M511
	Finish Mill Top Edges (B)	M402,M512
	Center Holes (5)	D201
	Drill 8 Dia Holes 25 Deep (5)	D108
	Deburr (5)	S102
	Tap 8 Dia Holes 25 Deep (5)	T108
	Box Mill Inside Edges (Rough) (C)	M401,M407
(2)	Finish Mill Inside Edges (C)	M402,M408
	Rough Mill Sides Of projection (D)	M404
	Face Mill Boss (E)	M401,M408
	Finish Mill Boss (E)	M402
	Center Hole (1)	D201
	Drill 20 dia Hole Thro' (1)	D120
	Ream 20 Dia Hole Thro' (1)	R120
	Deburr (1)	S102
	Side Mill ridge (Rough) (F)	M502
	Finish Mill Ridge (F)	M502,M701
(3)	Face Mill Surface (G)	M701,M704
	Finish Mill Surface (G)	M702
	Center Holes (3)	D201
	Drill 15 Dia Holes 10 Deep (3)	D115
	Drill 15 Dia Hole 25 Deep (for notch H)	D115
	Finish Mill Notch (H)	M413
	Rough Mill Pocket (I)	M506
	Finish Mill Pocket (I)	M508
	Center Hole (1)	D201
	Drill 8 Dia Hole 10 Deep (1)	D108
	Tap 8 dia Holes 10 Deep	T108
	Step Mill Edge (Rough) (J)	M506
	Finish Mill Edge (J)	M508
(4)	Contour Mill to Finish (J)	M601,M402
	Center Holes (2)	D202
	Drill 5 Dia Hole 5 deep	D105
	Deburr	S101

PART #8 BRACKET

Scale 1:5



PROCESS DETAILS

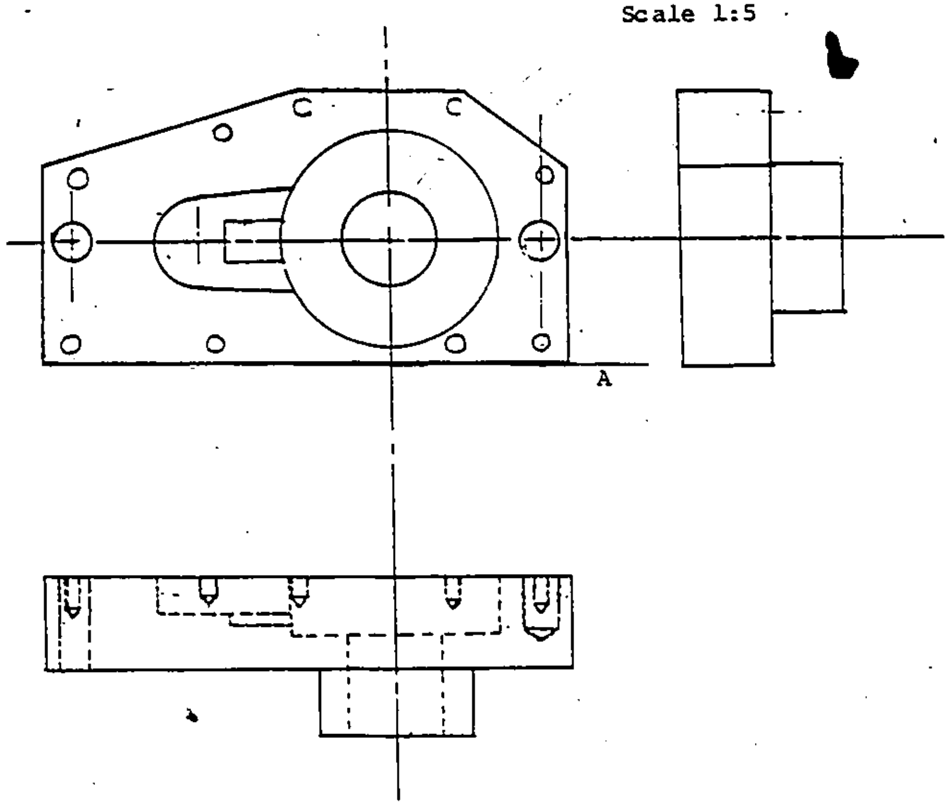
PART NO. : 8

PART NAME: BRACKET

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Face (A)	M101,M108, M501
	Rough Mill sides (C)	M401
	Finish Mill Bottom Face (A)	M102,M502
	Finish Mill Sides (C)	M401
(2)	Rough Mill Top Face (B)	M101
	Finish Mill Top face (B)	M102
	Center Holes (2)	D201
	Drill 25 Dia Hole Thro'	D125
	Drill 40 Dia Hole Thro'	D140
	Counter Bore	B105
	Deburr	S102
(3)	Rough Mill Face (D)	M702,M301
	Finish Mill Face (D)	M703
	Centering Holes (3)	D203
	Drill Dia 25 Hole (3)	D125
	Tap Dia 25 Hole (3)	T125
	Rough Bore Thro'	B108
	Enlarge Bore (Rough)	B109
	Chamfer	B110
	Finish Bore Thro'	B112

PART #9 COVER

Scale 1:5



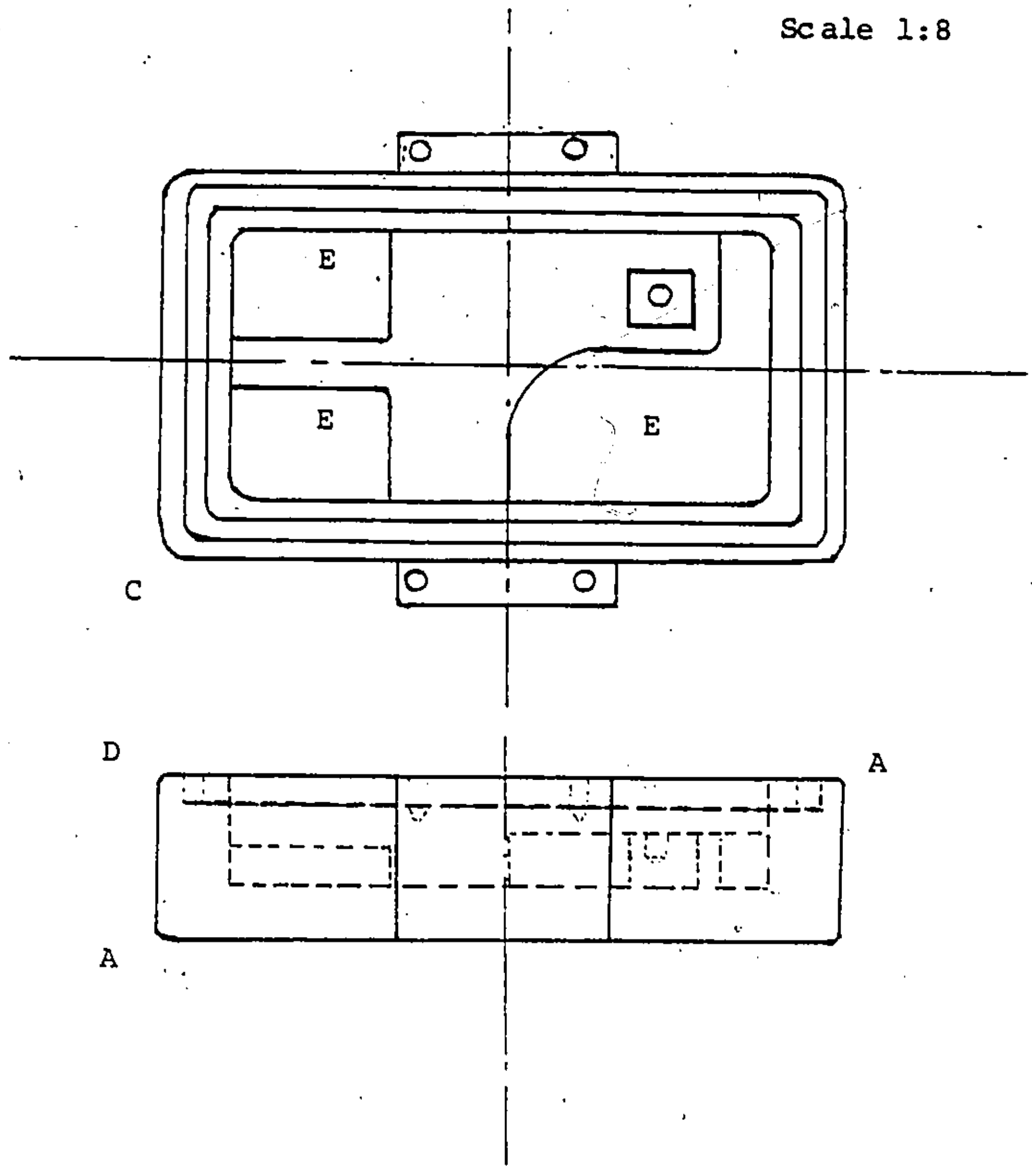
PROCESS DETAILS

PART NO. : 9 PART NAME: COVER

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Side (A)	M102,M401
	Finish Mill side (A)	M103, M105, M402
	Rough Mill Top Face	M101
	Finish Mill Top Face	M101
	Center Holes (9)	D203
	Drill 15 Dia Holes (9)	D115
	Center Holes (2)	D203
	Drill Dia 30 Holes (2) Thro'	D130
	Deburr	S101
	Tap 15 Dia Holes (9)	T115
(2)	Circular Milling (Rough)	M501,M506
	Circular Milling (Finish)	M502
	Thro' Bore 80 Dia (Rough)	B107
	Thro' Bore 80 Dia (Finish)	B109
	Counter Bore (Rough)	B112
	Contour Mill (Finish)	M402
	End Mill Pocket (Rough)	M501,M508
	End Mill Pocket (Finish)	M502
	Side Mill Pocket Edges	M301
	Shape Milling (Rough)	M603
	Shape Milling (Finish)	M604
(3)	Mill Pocket (Small)	M404,M402
	Finish Mill Pocket(Small)	M402
	Finish Mill All Sides	M302

PART #10 **BOX**

Scale 1:8



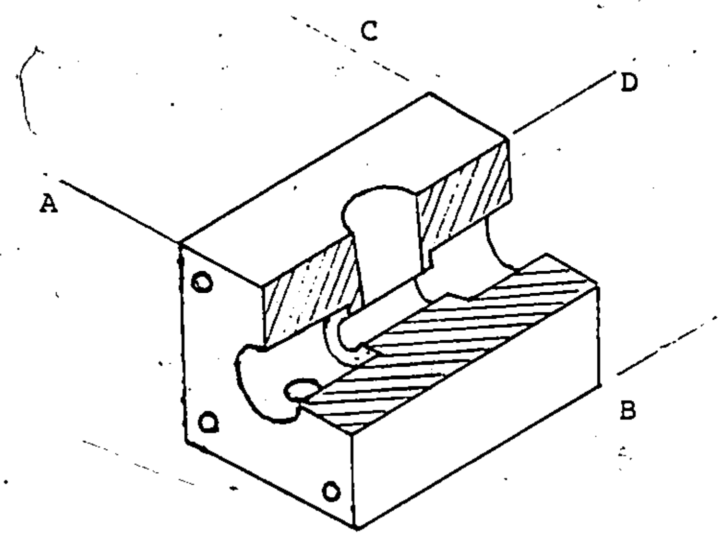
PROCESS DETAILS

PART NO. : 10 PART NAME: BOX

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Face	M101
	Finish Mil Bottom Face	M103
	Face Mill Side B (Rough)	M104,M503
	Face Mill Side B (Finish)	M105
	Face Mill Side C (Rough)	M104,M503
	Face Mill Side C (Finish)	M105
	Rough Mill Top Face (D)	M408,M106
	Finish Mill Top Face (D)	M403,M103
	Mill Grove (End Mill)	M415,M407
(2)	Center Holes (4)	D201
	Drill 15 Dia Holes (4)	D115
	Face Mill E (Rough)	M501,M509
	Face Mill E (Finish)	M502
	End Mill Edges (Rough)	M401
	End Mill Edges (Finish)	M402
	Contour Mill (Rough)	M501
	Contour Mill (Finish)	M902
	Face Mill Boss	M701
	Center Hole	D202
	Drill 15 Dia Hole 24 Deep	D115
	Deburr	S101
	Tap 15 Dia Hole 24 Deep	T115

PART #11 GUIDE

Scale 1:8



PROCESS DETAILS

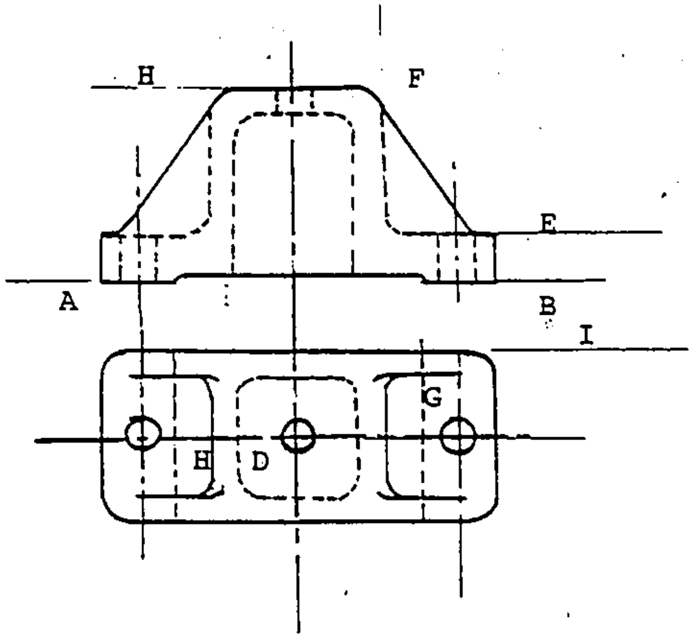
PART NO. : 11 PART NAME: GUIDE

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Face B	M112,M401
	Finish Mill Face B	M104,M412
	Center Hole	D201
	Drill 30 dia Hole	D130
(2)	Rough Mill Face A	M405
	Finish Mill Face A	M402,M410
	Rough Mill Face C	M405
	Finish Mill Face C	M402,M410
(3)	Bore Dia Hole Thro'	B101
	Finish Bore Dia Hole Thro'	B109
	Enlarge Dia Bore Deep	B105
	Center Holes (4)	D202
	Drill 16 Dia Holes 40 Deep	D116
	Tap 16 Dia Holes 40 Deep	T116
	Enlarge 80 Dia Bore 100 Deep	B109
	Center Hole (4) (Face C)	D202
	Drill 16 Dia Holes 40 Deep	D116
	Tap 16 Dia Holes 40 Deep	T116
	Rough Mill Face D	M112,M401
	Finish Mill Face D	M104,M412
	Enlarge 80 Dia Bore Thro' to Centre	B109

PART #12

BRACKET-A

Scale 1:10



PROCESS DETAILS

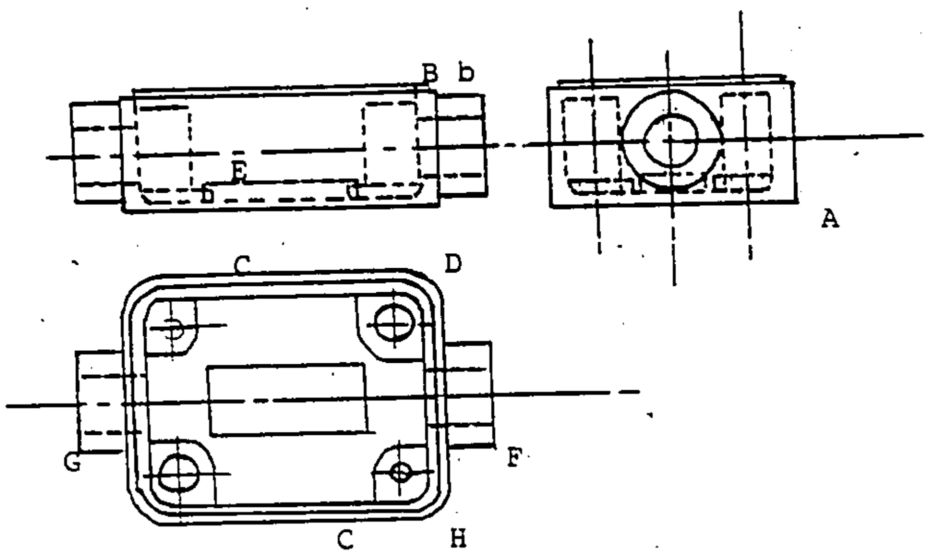
PART NO. : 12

PART NAME: BRACKET-A

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Bottom Edges A & B	M112,M401
	Finish Mill Bottom Edges A & B	M104,M105
(2)	Rough Mill Cavity D	M101,M106
	Semi-finish Cavity D	M104
	Finish Cavity D	M104
	Center Hole	D201
	Drill 30 Dia Hole Thro'	D130
	Ream 30 Dia Hole Thro'	R130
(3)	Mill faces E & F (Rough)	M401
	Finish Mill Feces E & F	M402
	Center Holes (2)	D202
	Drill 25 Dia Hole Thro'(2)	D125
	Deburr	S102
	Bore 30 Dia Holes Thro' (2)	B108
	Finish Bore 30 Dia Holes (2)	B109
	Face Mill Surface H	M505,M710
	Rough Mill Surface (H)	M107,M401
	Rough Mill Surface (G)	M702
	Rough MIll Surface (I)	M702,M704
		M502

PART #13 JUNCTION COVER

Scale 1:8



PROCESS DETAILS

PART NO. : 13 PART NAME: JUNCTION COVER

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Face A	M701,M101
	Finish Mill Face A	M702,M102
	Rough Mill Around Periphery B	M401,M413
	Finish Mill Around Periphery B	M402,M406
	Step Mill Around Periphery B	M301
	Finish Step	M302
	Rough Mill Surface C	M301,M306
	Finish Mill Surface C	M302,M315
	Rough Mill Round Edges	M801
	Finish Mill Round Edges	M802
	Rough Mill Bosses	M701
	Finish Mill Bosses	M702
	Center Holes (2)	D202
	Drill 40 Dia Holes (2)	D140
(2)	Ream 40 Dia Holes (2)	R140
	Tap 40 Dia Hole	T140
	Center Holes (2)	D201
	Drill 20 Dia Hole	D120
	Ream 20 Dia Hole	R120
	Tap 20 Dia Hole	T120
	Rough Mill Face E	M401
	Face Mill Face E	M402
	Rough Mill Inside Bottom Surface	M405
	Rough Mill Inside Edges	M301
	Face Mill F	M501
	Finish Mill F	M502
	Center Hole (1)	D201
	Drill 30 Dia Hole Thro' Wall	D130
	Deburr	S102
	Bore 56 Dia Hole Thro' Wall	B108
	Face Mill G	M501
(3)	Finish Mill G	M502
	Center Hole (1)	D201
	Drill 30 Dia Hole Thro' Wall	D130
	Deburr	S102
	Bore 56 Dia Hole Thro' Wall	D130
	Deburr	S102
	Bore 56 Dia Hole Theo' Wall	B108
	Rough Mill Face H	M501,M506
	Finish Mill Face H	M502,M508

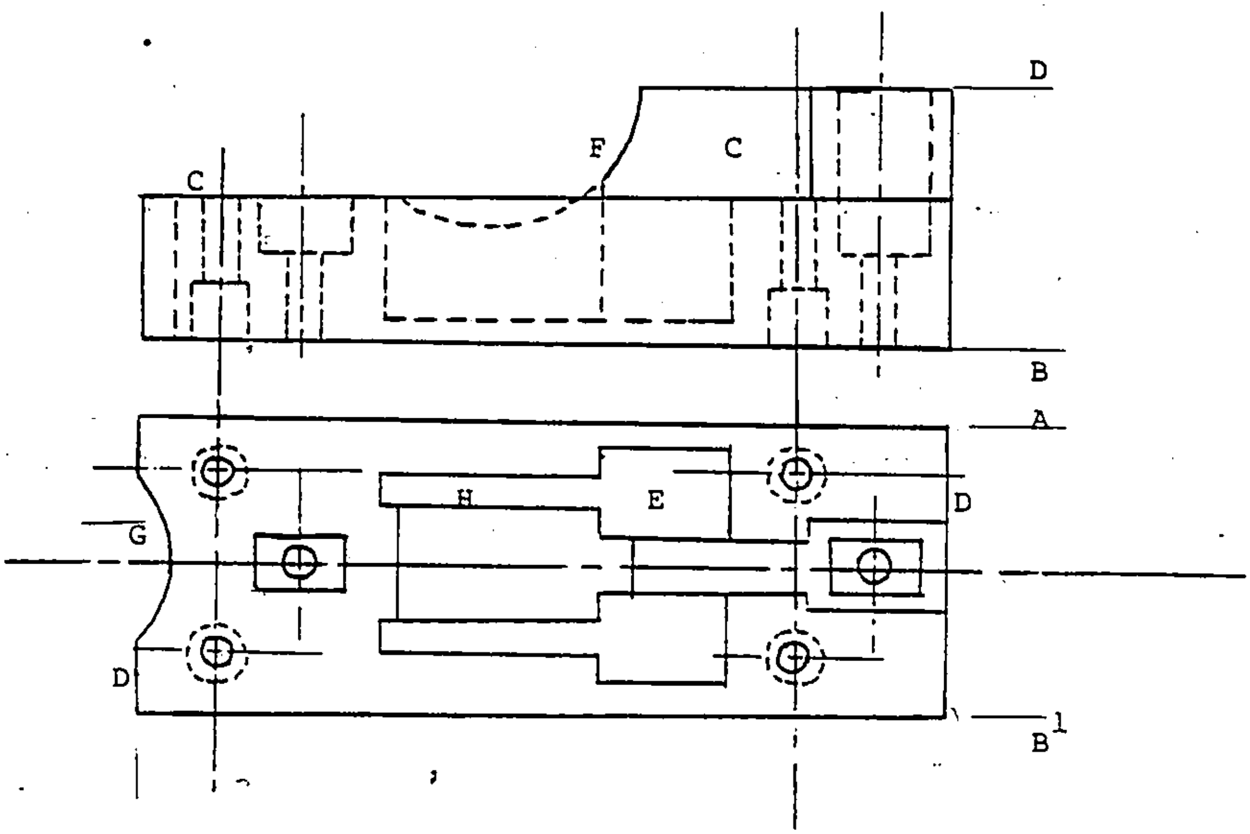
PROCESS DETAILS

PART NO. : 14 PART NAME: END SUPPORT

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Face A	M102,M401
	Finish Mill Face A	M103,M105
	Rough Mill Face B	M401,M508
	Semi Finish Face B	M402,M509
	Finish Face B	M402,M415
	Center Holes (4)	D202
	Drill 25 Dia Holes Thro' (4)	D125
	Ream 25 Dia Holes Thro' (4)	R125
	Bore 35 Dia Holes Thro' (4)	B102
	Center Holes (2)	D201
	Drill 25 Dia Holes Thro' (2)	D125
	Deburr	S102
	Ream 25 Dia Holes Thro' (2)	R125
	Tap 25 Dia Hole	T125
(2)	Face Mill Surface (C)	M401
	Finish Mill Surface (C)	M412
	Face Mill Surface (D)	M101
	Face Mill Surface (D)	M102
	Rough Bore Dia Holes (2)	B102
	Finish Bore Dia Holes (2)	B103
(3)	Rough Mill Face B1	M102,M401
	Finish Mill Face B1	M103,M105
	Face Mill C	M701
	Finish Mill C	M702
	Face Mill D	M701
	Finish Mill D	M702
(4)	Mill Bottom Face of Recess E	M401,M415
	Finish Mill Bottom of Recess E	M403,M705
	Rough Mill Walls of Recess E	M103
	Finish Mill Walls Of Recess E	M102
	Rough Mill Recess Projection H	M405
	Finish Mill Recess Projection H	M406
	Rough Mill Contour F	M601
	Finish Mill Contour F	M602
	Rough Mill Contour G	M603
	Finish Mill Contour G	M604

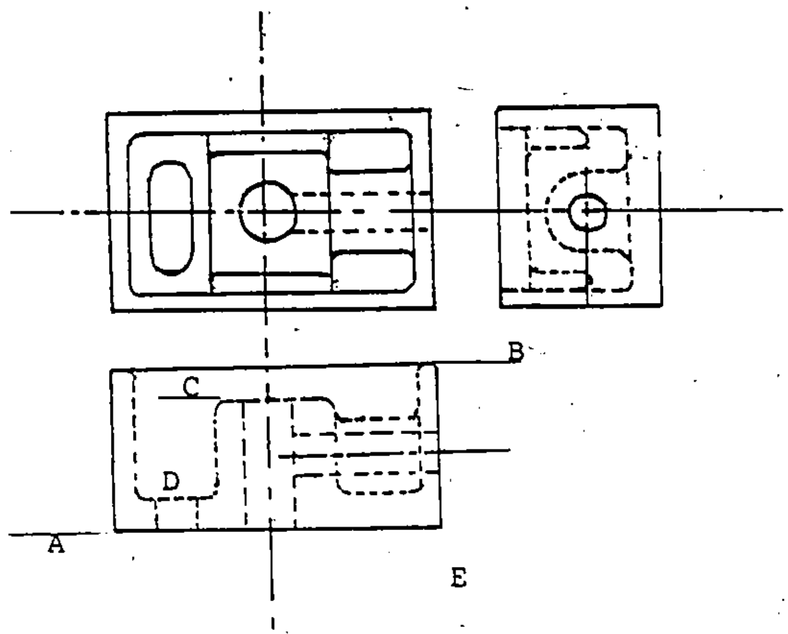
PART #14 END SUPPORT

Scale 1:5



PART #15 <u>CASING</u>

Scale 1:10



PROCESS DETAILS

PART NO. : 15 PART NAME: CASING

OPERATION NO.	DESCRIPTION OF THE OPERATION	TOOL(S) REQUIRED
(1)	Rough Mill Surface A	M105,M708
	Finish Mill Surface A	M102,M702
	Rough Mill Sides of Box	M108,M302
(2)	End Mill Edges B	M401,M412
	Finish Mill Edges B	M402
	Rough Mill Boss C	M401
	Finish Mill Boss C	M408
	Center Hole	D201
	Drill 50 Dia Hole Thro´	D150
	Bore 60 Dia Hole Thro´	B108
	Finish Bore 60 Dia Hole Thro´	B109
	Rough Mill Bottom D	M412
	End Mill Oblong Hole (Rough)	M416,M508
	Finish Mill Oblong Hole	M417
(3)	Finish Face (E)	M402
	Center Hole (1)	D202
	Drill 45 Dia Hole Thro´ to Centre	D145
	Bore 50 Dia Hole Thro´ to Centre	B112
	Finish Bore Hole Thro´ to Centre	B118

APPENDIX C

COMPUTER PROGRAM LISTINGS

```
//PA   JOB  (R240,NU7,10,5),'GAJ',CLASS=B,REGION=2048K
//       EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//    DISP=(NEW,KEEP),SPACE=(TRK,(80,10)),
//     DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN   DD  *
$JOB   WATFIV
C ----------------------------REF. SECTION 4.2-----------------
C CELL FORMATION IN FMS - PART FAMILY FORMATION PROBLEM
C ------------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C           GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
            UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 2Z2
C ------------------------------------------------------------
C THIS PROGRAM GENERATES THE INPUT FILE OF THE PROBLEM P(R)
C FOR THE SAS/OR (VERSION 5) INTEGER PROGRAMMING ROUTINE
C ------------------------------------------------------------
C   THE  VARIABLE DECLARATION SECTION ******************
        INTEGER NUMOP(40),SIM(40,40),DISIM(40,40)
        INTEGER OPI,OPJ,A
        INTEGER X(2500),XIND,XVARS,MVARS,XINDI,XINDJ
        INTEGER RHS,KX(2500),KM(2500)
        INTEGER MK(40,3),MEM(40,3),NNN(3)
        REAL LAMDA,M(2500),COEFF(40,40)
        CHARACTER CTYPE*8,FLAG*8,OP*4(40,50)
        COMMON OP,NUMOP,SIM,DISIM,OPI,OPJ,COEFF,I,J,IFAM,K,
     *  L,NM,MP,MN,N,K1
C   THE DATA INPUT AND CALCULATION OF SIMILAR AND DISSIMILAR PROCESSES
C   BETWEEN THE PARTS                    ******************
        READ(5,10) N,K
        READ,LAMDA
 10     FORMAT(I2,1X,I2)
        DO 100 I=1,N
          READ(5,20) NUM,(OP(I,J),J=1,NUM)
          NUMOP(I)=NUM
 20       FORMAT(I2,10(1X,A4),/,(3X,10(A4,1X)))
 100    CONTINUE
        NM1=N-1
        DO 200 I=1,NM1
        I1=I+1
        DO 300 J=I1,N
          SIM(I,J)=0
          NTERMI=NUMOP(I)
          NTERMJ=NUMOP(J)
          DO 400 OPI=1,NTERMI
            DO 500 OPJ=1,NTERMJ
              IF (OP(I,OPI).EQ.OP(J,OPJ)) SIM(I,J)=SIM(I,J)+1
 500        CONTINUE
 400      CONTINUE
          DISIM(I,J)=NUMOP(I)+NUMOP(J)-2*SIM(I,J)
          COEFF(I,J)=DISIM(I,J)-LAMDA*SIM(I,J)
 300     CONTINUE
 200     CONTINUE
C CREATING THE SAS FORMAT INPUT DATA FOR PART FAMILY FORMATION PROBLEM
C                                        ******************
        METH=3
        CALL TINIT
        NBLOCK=0
        EXECUTE ALLCLR
        EXECUTE TIMEL
```

```
      EXECUTE OBJROW
      EXECUTE TIMEL
      EXECUTE ALLOC
      EXECUTE TIMEL
      IF (METH.EQ.1) THEN
        EXECUTE CNSTR
        EXECUTE MLIM
        EXECUTE INTGER
        EXECUTE UPPER
      ENDIF
      IF (METH.EQ.2) THEN
        EXECUTE CNSTR2
        EXECUTE MLIM2
        EXECUTE INTGER
        EXECUTE UPPER
      ENDIF
      IF (METH.EQ.3) THEN
        EXECUTE CNSTR2
        EXECUTE TIMEL
        EXECUTE MLIM
        EXECUTE TIMEL
        EXECUTE INTGER
        EXECUTE TIMEL
        EXECUTE UPPER
        EXECUTE TIMEL
      ENDIF
      IF (METH.EQ.4) THEN
        EXECUTE CNSTR
        EXECUTE MLIM2
        EXECUTE INTGER
        EXECUTE UPPER
      ENDIF
      IF(INSOL.EQ.1) EXECUTE BASICS
      GO TO 9999
      REMOTE BLOCK TIMEL
      CALL TUSED(MSEC)
      NBLOCK=NBLOCK+1
      PRINT, 'TIME USED FOR BLOCK',NBLOCK, '  IS ',MSEC
      CALL TINIT
      ENDBLOCK
C  REMOTE BLOCK ALLCLR                        ******************
      REMOTE BLOCK ALLCLR
      DO  3030 II=1,N
      DO  3030 IIFAM=1,K
        XIND= (II-1)*K + IIFAM
        X(XIND)=0
3030  CONTINUE
      NM1=N-1
      DO 3040 II=1,NM1
      II1=II+1
        DO 3050 JJ=II1,N
          DO 3060 IIFAM=1,K
            LI=II
            LJ=JJ
            LIFAM=IIFAM
            EXECUTE MIJKA
            M(MINDEX)=0
3060      CONTINUE
3050    CONTINUE
3040  CONTINUE
```

```
         ENDBLOCK
C   REMOTE BLOCK MIJKA                          ******************
         REMOTE BLOCK MIJKA
         MINDA=0
         LI1=LI+1
         ISIGM =LI-1
         IF (ISIGM.EQ.0) GO TO 3064
         DO 3065 IX = 1,ISIGM
           MINDA=MINDA + (N-IX) * K
3065     CONTINUE
3064     MINDEX=MINDA+ (LJ-LI1)*K +LIFAM
         ENDBLOCK
C   REMOTE BLOCK OBJROW                         ******************
        REMOTE BLOCK OBJROW
        NM1= N-1
        DO 3100 I=1,NM1
          I1=I+1
          DO 3110 J=I1,N
            DO 3120 IFAM=1,K
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX) = COEFF(I,J)
3120        CONTINUE
3110      CONTINUE
3100    CONTINUE
        CTYPE='MIN'
        FLAG='OBJROW'
        EXECUTE PUTROW
        ENDBLOCK
C REMOTE BLOCK PUTROW                           ******************
        REMOTE BLOCK PUTROW
        XVARS=N*K
        MVARS=N*(N-1)/2*K
        IF (FLAG.EQ.'OBJROW') THEN
          WRITE(8,4000) (X(II),II=1,XVARS)
4000      FORMAT(15(' ',I4))
          WRITE(8,4002) (M(JJ),JJ=1,MVARS)
4002      FORMAT(8(' ',F9.2))
        ELSE
          DO 7856 II=1,XVARS
            KX(II) = X(II)
7856      CONTINUE
          DO 7857 JJ=1,MVARS
            KM(JJ)=M(JJ)
7857      CONTINUE
          WRITE(8,4112) (KX(II),II=1,XVARS)
          WRITE(8,4113) (KM(JJ),JJ=1,MVARS)
4112      FORMAT(15(' ',I4))
4113      FORMAT(15(' ',I4))
       ENDIF
        IF (FLAG.EQ.'OBJROW') THEN
          WRITE(8,4010) CTYPE
4010      FORMAT(' ',A8,'  .')
        ELSEIF (FLAG.EQ.'ALLOC') THEN
          WRITE(8,4020) CTYPE,RHS
4020      FORMAT(' ',A8,1X,I5)
        ELSEIF (FLAG.EQ.'CNSTR') THEN
          WRITE(8,4020) CTYPE,RHS
```

```
      ELSEIF (FLAG.EQ.'MLIM') THEN
        WRITE(8,4020) CTYPE,RHS
      ELSEIF (FLAG.EQ.'INTGER') THEN
        WRITE(8,4010) CTYPE
      ELSEIF (FLAG.EQ.'UPPER') THEN
        WRITE(8,4010) CTYPE
      ENDIF
      EXECUTE ALLCLR
      ENDBLOCK
C REMOTE BLOCK ALLOC                                ******************
      REMOTE BLOCK ALLOC
      FLAG='ALLOC'
      CTYPE='EQ'
      RHS=1
      DO 3200 I=1,N
        DO 3210 IFAM=1,K
          XIND= (I-1)*K+ IFAM
          X(XIND)=1
3210    CONTINUE
      EXECUTE PUTROW
3200  CONTINUE
      ENDBLOCK
C REMOTE BLOCK CNSTR                                ******************
      REMOTE BLOCK CNSTR
      FLAG='CNSTR'
      CTYPE='LE'
      RHS=1
      NM1=N-1
      DO 3230 I=1,NM1
        I1=I+1
        DO 3240 J=I1,N
          DO 3250 IFAM=1,K
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            M(MINDEX) = -1
            XINDI= (I-1) * K + IFAM
            XINDJ= (J-1) * K + IFAM
            X(XINDI)=1
            X(XINDJ)=1
            EXECUTE PUTROW
3250      CONTINUE
3240    CONTINUE
3230  CONTINUE
      ENDBLOCK
C REMOTE BLOCK CNSTR2                               ******************
      REMOTE BLOCK CNSTR2
      FLAG='CNSTR'
      CTYPE='LE'
      NM1=N-1
      DO 3400 I=1,NM1
        DO 3410 IFAM=1,K
          XIND=(I-1)*K + IFAM
          RHS=N-I
          X(XIND)=N-I
          I1=I+1
          DO 3420 J=I1,N
            LI=I
            LJ=J
```

```
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX)=-1
              XINDJ=(J-1)*K+IFAM
              X(XINDJ)=1
3420        CONTINUE
            EXECUTE PUTROW
3410      CONTINUE
3400    CONTINUE
        ENDBLOCK
C REMOTE BLOCK MLIM                         ******************
        REMOTE BLOCK MLIM
        FLAG='MLIM'
        CTYPE='LE'
        RHS=0
        NM1=N-1
        DO 3260 I=1,NM1
          I1=I+1
          DO 3270 J=I1,N
            DO 3280 IFAM=1,K
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX)=1
              XINDI= (I-1)*K + IFAM
              X(XINDI)=-1
              EXECUTE PUTROW
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              XINDJ=(J-1)*K + IFAM
              M(MINDEX)=1
              X(XINDJ)=-1
              EXECUTE PUTROW
3280        CONTINUE
3270      CONTINUE
3260    CONTINUE
        ENDBLOCK
C REMOTE BLOCK MLIM2                        ******************
        REMOTE BLOCK MLIM2
        FLAG='MLIM'
        CTYPE='LE'
        RHS=0
        DO 3500 I=1,N
          DO 3510 IFAM=1,K
            XIND=(I-1)*K+IFAM
            X(XIND)=1-N
            DO 3520 J=1,N
              IF (J.EQ.I) GO TO 3520
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX)=1
3520        CONTINUE
            EXECUTE PUTROW
3510      CONTINUE
3500    CONTINUE
```

```
         ENDBLOCK
C REMOTE BLOCK INTGER                              ******************
       REMOTE BLOCK INTGER
       FLAG='INTGER'
       CTYPE='INTEGER'
       DO 3300 I=1,N
         K11=K-1
         DO 3310 IFAM=1,K11
           XIND= (I-1) * K + IFAM
           X(XIND)=1
3310     CONTINUE
3300   CONTINUE
       EXECUTE PUTROW
       ENDBLOCK
C REMOTE BLOCK UPPER
       REMOTE BLOCK UPPER
       FLAG='UPPER'
       CTYPE='UPPERBD'
       DO 3700 I=1,N
         DO 3710 IFAM=1,K
           XIND=(I-1)*K +IFAM
           X(XIND)=1
3710     CONTINUE
3700    CONTINUE
        NM1=N-1
        DO 3730 I=1,NM1
          I1=I+1
          DO 3740 J=I1,N
            DO 3750 IFAM=1,K
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX)=1
3750        CONTINUE
3740      CONTINUE
3730    CONTINUE
        EXECUTE PUTROW
        ENDBLOCK
9999    STOP
        END
$ENTRY
25 03
2.45624
26 M501 M502 M701 M602 M101 M401 M102 M103 M301 M702
   M603 M108 M503 M504 D201 D142 D109 D202 D130 R142
   R130 R148 B108 B109 B101 B115
18 M101 M103 M104 M105 M901 M602 M501 M506 M508 M402
   D201 D125 R128 S104 B101 B109 B105 B104
23 M403 M404 M701 M401 M412 M405 M406 M702 M712 M101
   M102 D202 D128 D203 R130 T130 B108 B109 B101 B102
   B115 B106 B112
23 M501 M518 M502 M701 M712 M211 M101 M102 M212 M503
   M301 M302 M401 M405 D202 D150 D201 D120 D202 B108
   B112 S109 T120
29 M102 M401 M103 M105 M602 M603 M501 M513 M502 M503
   M101 M701 M702 M402 M508 M406 M610 M301 M305 D201
   D108 D150 D203 D170 R108 B106 B109 B108 S101
18 M501 M504 M502 M415 M412 M416 M901 M902 M401 M507
   M509 D201 D118 D130 B108 B112 B109 S102
```

```
28 M102 M401 M103 M105 M404 M511 M402 M512 M407 M408
   M502 M701 M704 M702 M413 M506 M508 M601 D201 D108
   D120 D115 D202 D105 S102 S101 T108 R120
20 M101 M108 M501 M401 M102 M502 M702 M301 M703 D201
   D125 D140 D203 B105 B108 B109 B112 B110 S102 T125
24 M102 M401 M103 M105 M402 M101 M106 M501 M506 M502
   M508 M301 M603 M604 M404 M302 D203 D115 D130 S101
   T115 B107 B109 B112
24 M101 M103 M104 M503 M105 M408 M106 M403 M415 M407
   M501 M509 M703 M502 M401 M402 M901 M902 M701 D201
   D115 D202 S101 T115
15 M112 M401 M104 M412 M405 M402 M410 D201 D130 D202
   D116 B101 B109 B105 T116
21 M112 M401 M104 M105 M101 M106 M402 M505 M710 M107
   M702 M704 M502 D201 D130 D202 D125 B108 B109 R130
   S102
30 M701 M101 M702 M102 M401 M413 M406 M301 M302 M306
   M315 M801 M802 M402 M405 M501 M502 M506 M508 D202
   D140 D201 D120 D130 R140 R120 T140 T120 S102 B108
27 M102 M401 M103 M105 M508 M402 M509 M415 M412 M101
   M701 M702 M403 M405 M406 M601 M602 M603 M604 D202
   D125 D201 R125 B102 B103 S102 T125
21 M105 M708 M102 M702 M108 M302 M401 M412 M402 M408
   M416 M417 M508 D201 D150 B108 B109 D202 D145 B112
   B118
$IBSYS
$STOP
//
```

```
//PB    JOB  (R240,NU7,10,5),'GAJ',CLASS=A,REGION=2048K
//         EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//    DISP=(NEW,KEEP),SPACE=(TRK,(40,10)),
//     DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN   DD  *
$JOB   WATFIV                          REF. SECTION 4.2.2
C -----------------------------------------------------------
C CELL FORMATION IN FMS - PART FAMILY FORMATION PROBLEM
C -----------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C          GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
           UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 2Z2
C -----------------------------------------------------------
C THIS PROGRAM IMPLEMENTS THE ALGORITHM FOR FINDING THE
C LOWER BOUND AND UPPERBOUND FOR FUNCTION P(R) WITH
C DIFFERENT VALUES OF R TO ESTABLISH THE REGION (L,U).
C -----------------------------------------------------------
C   THE  VARIABLE DECLARATION SECTION ******************
        INTEGER NUMOP(40),SIM(40,40),DISIM(40,40)
        INTEGER OPI,OPJ,A
        INTEGER X(500),XIND,XVARS,MVARS,XINDI,XINDJ
        INTEGER RHS
        REAL LAMDA,M(2500),COEFF(40,40),S(435),LBOUND
        CHARACTER CTYPE*8,FLAG*8,OP*4(40,80)
        COMMON OP,NUMOP,SIM,DISIM,OPI,OPJ,COEFF,I,J,IFAM,K,
     *  L,NM,MP,MN,N,K1
C   THE DATA INPUT AND CALCULATION OF SIMILAR AND DISSIMILAR PROCESSES
C   BETWEEN THE PARTS                        ******************
        READ(5,10) N,K
 10     FORMAT(I2,1X,I2)
        DO 100 I=1,N
         PRINT, 'PART NUMBER =',I
          READ(5,20) NUM,(OP(I,J),J=1,NUM)
          NUMOP(I)=NUM
 20       FORMAT(I2,10(1X,A4),/,(3X,10(A4,1X)))
C         WRITE(8,20) NUMOP(I),(OP(I,J),J=1,NUM)
 100    CONTINUE
        NM1=N-1
        DO 200 I=1,NM1
        I1=I+1
        DO 300 J=I1,N
          SIM(I,J)=0
          NTERMI=NUMOP(I)
          NTERMJ=NUMOP(J)
          DO 400 OPI=1,NTERMI
            DO 500 OPJ=1,NTERMJ
              IF (OP(I,OPI).EQ.OP(J,OPJ)) SIM(I,J)=SIM(I,J)+1
 500        CONTINUE
 400      CONTINUE
          DISIM(I,J)=NUMOP(I)+NUMOP(J)-2*SIM(I,J)
 300     CONTINUE
 200     CONTINUE
         WRITE(8,4018)
4018     FORMAT(' BOUNDS ON THE OBJECTIVE FUNCTION FOR DIFFERENT VALUES
     *  OF  R  ')
         WRITE(8,4053)
         WRITE(8,4024)
4024     FORMAT(  '  ')
         WRITE(8,4024)
```

```
        WRITE(8,4026) N
4026    FORMAT('  # OF PARTS= ',I2)
        WRITE(8,4027) K
4027    FORMAT('  # OF FAMILIES=',I2)
        WRITE(8,4024)
        WRITE(8,4029) N*(N-1)/2*K
4029    FORMAT('  # OF NON-ZERO COEFFICIENTS  =',I5)
        WRITE(8,4053)
        WRITE(8,4024)
        WRITE(8,4024)
       WRITE(8,9000)
9000   FORMAT('   R       # OF NEGATIVE     OBJECTIVE FUNCTION ')
       WRITE(8,9001)
9001   FORMAT('           COEFFICIENTS    LOWERBOUND    UPPERBOUND')
       WRITE(8,9002)
9002   FORMAT('  -----    ------------    -----------   -----------')
       EXECUTE ALLCLR
       DO 18 LLAMDA=5,1000,5
       LAMDA=LLAMDA/100.00
       NM1=N-1
       DO 21 I=1,NM1
         I1=I+1
         DO 22 J=I1,N
           COEFF(I,J)=DISIM(I,J)-LAMDA*SIM(I,J)
 22      CONTINUE
 21    CONTINUE
       DO 89 JM=1,435
        S(JM)=0.0
 89    CONTINUE
       EXECUTE OBJROW
 18    CONTINUE
       GO TO 9999
C  REMOTE BLOCK ALLCLR                    ******************
      REMOTE BLOCK ALLCLR
      DO  3030 II=1,N
      DO  3030 IIFAM=1,K
        XIND= (II-1)*K + IIFAM
        X(XIND)=0
3030  CONTINUE
      NM1=N-1
      DO 3040 II=1,NM1
      III=II+1
        DO 3050 JJ=III,N
          DO 3060 IIFAM=1,K
            LI=II
            LJ=JJ
         LIFAM=IIFAM
            EXECUTE MIJKA
            M(MINDEX)=0
3060      CONTINUE
3050    CONTINUE
3040  CONTINUE
      ENDBLOCK
C   REMOTE BLOCK MIJKA                    ******************
      REMOTE BLOCK MIJKA
      MINDA=0
      LI1=LI+1
      ISIGM =LI-1
      IF (ISIGM.EQ.0) GO TO 3064
      DO 3065 IX = 1,ISIGM
```

```
         MINDA=MINDA + (N-IX) * K
3065    CONTINUE
3064    MINDEX=MINDA+ (LJ-LI1)*K +LIFAM
        ENDBLOCK
C   REMOTE BLOCK OBJROW                    ******************
       REMOTE BLOCK OBJROW
       NM1= N-1
        NEG=0
          MS=1
       DO 3100 I=1,NM1
         I1=I+1
         DO 3110 J=I1,N
           DO 3120 IFAM=1,K
             LI=I
             LJ=J
             LIFAM=IIFAM
             EXECUTE MIJKA
             M(MINDEX) = COEFF(I,J)
             IF(COEFF(I,J).LT.0) NEG=NEG+1
             IF(IFAM.EQ.1) THEN
                S(MS)=COEFF(I,J)
                MS=MS+1
              ENDIF
3120        CONTINUE
3110      CONTINUE
3100    CONTINUE
        CTYPE='MIN'
        FLAG='OBJROW'
        IES=N/K
        IPAIRS=IES*(IES-1)/2*K + (N-IES*K)*IES
        MPAIRS=N*(N-1)/2
        LBOUND=0.000
        UBOUND=0.000
        ICOUNT=0
        DO 889 KI=1,435
          INTS=S(KI)*10000.000
          IF (INTS.NE.0) THEN
            ICOUNT=ICOUNT+1
          ELSE
            GO TO 889
          ENDIF
          IF (S(KI).LT.0.00) THEN
            LBOUND=LBOUND+S(KI)
          ENDIF
          IF (S(KI).GT.0.00) THEN
            IF(ICOUNT.LE.IPAIRS) LBOUND=LBOUND+S(KI)
          ENDIF
889     CONTINUE
        ICOUNT=0
        DO 899 KI=1,435
          INTS=S(435-KI+1)*10000.000
          IF (INTS.NE.0) THEN
            ICOUNT=ICOUNT+1
          ELSE
            GO TO 899
          ENDIF
          IF (S(435-KI+1).GT.0.00) THEN
            UBOUND=UBOUND+S(435-KI+1)
          ENDIF
          IF (S(435-KI+1).LT.0.00) THEN
```

```
                IF(ICOUNT.LE.IPAIRS) UBOUND=UBOUND+S(435-KI+1)
              ENDIF
899         CONTINUE
            EXECUTE PUTROW
            ENDBLOCK
C REMOTE BLOCK PUTROW                                    ******************
            REMOTE BLOCK PUTROW
            XVARS=N*K
            MVARS=N*(N-1)/2*K
            EXECUTE ALLCLR
            ENDBLOCK
9999         STOP
             END
$ENTRY

     THE INPUT DATA IS THE SAME AS THE ONE LISTED FOR
     THE PROGRAM 1.

$IBSYS
$STOP
//
```

```
//PC    JOB  (R240,NU7,1,5),'RAO',CLASS=Z,REGION=2048K
//        EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//   DISP=(NEW,KEEP),SPACE=(TRK,(40,10)),
//   DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN   DD  *
$JOB   WATFIV
C ------------------------------ REF. SECTION 4.3.2 ------------
C CELL FORMATION IN FMS - PART FAMILY FORMATION PROBLEM
C -------------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C           GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
            UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 2Z2
C -------------------------------------------------------------
C THIS PROGRAM GENERATES THE INPUT FILE OF THE SUBPROBLEMS
C OF THE TYPE P(R) TO BE SOLVED IN THE ITERATIONS OF THE
C APPROXIMATION PROCEDURE, FOR INPUT TO THE SAS/OR(VERSION 5)
C INTEGER PROGRAMMING ROUTINE.
C -------------------------------------------------------------
C   THE  VARIABLE DECLARATION SECTION ******************
         INTEGER NUMOP(30),SIM(30,30),DISIM(30,30)
         INTEGER OPI,OPJ,A
         INTEGER XIND,XVARS,MVARS,XINDI,XINDJ
         INTEGER RHS
         INTEGER MK(30,3),MEM(30,3),NNN(3),FAMN(3),FAMD(3)
         INTEGER FSIM(30,3),FDISIM(30,3),NSIM(30,30)
         INTEGER NDISIM(30,30)
         REAL LAMDA,OP*4(30,50),COEFF(30,30),FCOEFF(30,3),M(500),X(500)
         REAL FAMC(3),MINC
         CHARACTER CTYPE*8,FLAG*8
         COMMON OP,NUMOP,SIM,DISIM,OPI,OPJ,COEFF,I,J,IFAM,K,
     *   L,NM,MP,MN,N,K1
C   THE DATA INPUT AND CALCULATION OF SIMILAR AND DISSIMILAR PROCESSES
C   BETWEEN THE PARTS                     ******************
         READ(5,10) N,K
 10      FORMAT(I2,1X,I2)
         DO 100 I=1,N
           READ(5,20) NUM,(OP(I,J),J=1,NUM)
           NUMOP(I)=NUM
 20        FORMAT(I2,10(1X,A4),/,(3X,10(A4,1X)))
           WRITE(6,20) NUMOP(I),(OP(I,J),J=1,NUM)
 100     CONTINUE
         NM1=N-1
         DO 200 I=1,NM1
         I1=I+1
         DO 300 J=I1,N
           SIM(I,J)=0
           NTERMI=NUMOP(I)
           NTERMJ=NUMOP(J)
           DO 400 OPI=1,NTERMI
             DO 500 OPJ=1,NTERMJ
               IF (OP(I,OPI).EQ.OP(J,OPJ)) SIM(I,J)=SIM(I,J)+1
 500         CONTINUE
 400       CONTINUE
           PRINT, 'SIMILARITY BETWEEN',I,'AND',J,'IS',SIM(I,J)
           DISIM(I,J)=NUMOP(I)+NUMOP(J)-2*SIM(I,J)
          PRINT,'DISSIMILARITY BETWEEN ',I, ' AND ',J,' IS ',DISIM(I,J)
 300      CONTINUE
 200      CONTINUE
C READING THE INITIAL CONFIGURATION
```

```
      LAMDA=2.4562420
     DO 2010 IFAM =1,K
       DO 2010 I=1,N
         MK(I,IFAM)=0
         MEM(I,IFAM)=0
2010   CONTINUE
       DO 2020 IFAM=1,K
         READ(5,2030) NUMF,(MK(I,IFAM),I=1,NUMF)
        WRITE(6,2036) NUMF,(MK(I,IFAM),I=1,NUMF)
2036    FORMAT('INITIAL CONFIGURATION',I2,10(1X,I2))
         NNN(IFAM)=NUMF
2030     FORMAT(I2,20(1X,I2))
2020   CONTINUE
      DO 2050 IFAM=1,K
        DO 2060 I=1,N
          LU=NNN(IFAM)
          DO 2070 LL=1,LU
            IF (MK(LL,IFAM).EQ.I) MEM(I,IFAM) =1
2070      CONTINUE
2060    CONTINUE
2050  CONTINUE
      TFAMD=0.0
      DO 2200 IFAM=1,K
        N1=N-1
        FAMN(IFAM)=0
        FAMD(IFAM)=0
        DO 2240 I=1,N1
        I1=I+1
          DO  2240 J=I1,N
      FAMN(IFAM)=FAMN(IFAM) +MEM(I,IFAM)*MEM(J,IFAM)*DISIM(I,J)
      FAMD(IFAM)=FAMD(IFAM) +MEM(I,IFAM)*MEM(J,IFAM)*SIM(I,J)
2240      CONTINUE
           TFAMD=TFAMD+FAMD(IFAM)
2200    CONTINUE
        DO 188 IFAM=1,K
          FAMC(IFAM)=FAMN(IFAM)/TFAMD
 188    CONTINUE
        MINC=0
        DO 2260 IFAM=1,K
          IF (FAMC(IFAM).GT.MINC) THEN
            MF=IFAM
            MINC=FAMC(IFAM)
          ENDIF
2260    CONTINUE
C   IMPOSING THE MF **********
             MF=2
        NEWN=NNN(MF)
        NEWN1=NEWN-1
        DO 2300 NI=1,NEWN1
          NI1=NI+1
          DO 2400 NJ=NI1,NEWN
            I=MK(NI,MF)
            J=MK(NJ,MF)
            IF (I.LT.J) THEN
              NSIM(NI,NJ)=SIM(I,J)
              NDISIM(NI,NJ)=DISIM(I,J)
            ENDIF
            IF (I.GT.J) THEN
              NSIM(NI,NJ)=SIM(J,I)
              NDISIM(NI,NJ)=DISIM(J,I)
```

```
               ENDIF
2400          CONTINUE
2300         CONTINUE
             DO 2401 I=1,N
               DO 2410 IFAM=1,K
                 FSIM(I,IFAM)=0
                 FDISIM(I,IFAM)=0
2410           CONTINUE
2401         CONTINUE
             DO 2430 NI=1,NEWN
               NPART=MK(NI,MF)
               DO 2440 IFAM=1,K
                 IF (IFAM.EQ.MF) GO TO 2440
                 DO 2460 J=1,N
                   IF (J.EQ.NPART) GO TO 2460
                   IF (J.GT.NPART) THEN
               FSIM(NI,IFAM) =FSIM(NI,IFAM) + SIM(NPART,J)*MEM(J,IFAM)
             FDISIM(NI,IFAM) =FDISIM(NI,IFAM) +DISIM(NPART,J)*MEM(J,IFAM)
                   ELSE
                 FSIM(NI,IFAM)=FSIM(NI,IFAM) +SIM(J,NPART)*MEM(J,IFAM)
               FDISIM(NI,IFAM)=FDISIM(NI,IFAM)+DISIM(J,NPART)*MEM(J,IFAM)
                   ENDIF
2460             CONTINUE
2440           CONTINUE
2430         CONTINUE
             N1=N-1
             DO 2462 I=1,N
               I1=I+1
               DO 2470 J=I1,N
                 SIM(I,J)=0
                 DISIM(I,J)=0
2470           CONTINUE
2462         CONTINUE
             NEWN1=NEWN-1
             DO 2480 NI=1,NEWN1
               NI1=NI+1
               DO 2490 NJ=NI1,NEWN
                 SIM(NI,NJ)=NSIM(NI,NJ)
                 DISIM(NI,NJ)=NDISIM(NI,NJ)
                 COEFF(NI,NJ)=DISIM(NI,NJ)-LAMDA*SIM(NI,NJ)
2490           CONTINUE
2480         CONTINUE
             DO 2550 I=1,NEWN
               DO 2560 IFAM=1,K
                 FCOEFF(I,IFAM)=FDISIM(I,IFAM)-LAMDA*FSIM(I,IFAM)
2560           CONTINUE
2550         CONTINUE
             N=NEWN
             C=0.0
             DO 58 IFAM=1,K
               IF (IFAM.EQ.MF) GO TO 58
               C=C+FAMN(IFAM)-LAMDA*FAMD(IFAM)
         PRINT ,'CONSTANT ',IFAM,' IS ',FAMN(IFAM)-LAMDA*FAMD(IFAM)
58           CONTINUE
          PRINT,' THE CONSTANT FACTOR = ',C
         DO 45 I=1,N
          DO 46 IFAM=1,K
           IF (IFAM.EQ.MF) GO TO 46
           PRINT,'FSIM(',I,IFAM,') =',FSIM(I,IFAM)
           PRINT,'FDISIM(',I,IFAM,') =',FDISIM(I,IFAM)
```

```
46      . CONTINUE
45     CONTINUE
C CREATING THE SAS FORMAT INPUT DATA FOR PART FAMILY FORMATION PROBLEM
C                                                 ******************
          METH=3
          EXECUTE ALLCLR
          EXECUTE OBJROW
          EXECUTE ALLOC
          IF (METH.EQ.1) THEN
            EXECUTE CNSTR
            EXECUTE MLIM
            EXECUTE INTGER
            EXECUTE UPPER
          ENDIF
          IF (METH.EQ.2) THEN
            EXECUTE CNSTR2
            EXECUTE MLIM2
            EXECUTE INTGER
            EXECUTE UPPER
          ENDIF
          IF (METH.EQ.3) THEN
            EXECUTE CNSTR2
            EXECUTE MLIM
            EXECUTE INTGER
            EXECUTE UPPER
          ENDIF
          IF (METH.EQ.4) THEN
            EXECUTE CNSTR
            EXECUTE MLIM2
            EXECUTE INTGER
            EXECUTE UPPER
          ENDIF
          GO TO 9999
C  REMOTE BLOCK ALLCLR                            ******************
         REMOTE BLOCK ALLCLR
         DO  3030 II=1,N
         DO  3030 IIFAM=1,K
           XIND= (II-1)*K + IIFAM
           X(XIND)=0
3030     CONTINUE
         NM1=N-1
         DO 3040 II=1,NM1
         II1=II+1
           DO 3050 JJ=II1,N
             DO 3060 IIFAM=1,K
               LI=II
               LJ=JJ
               LIFAM=IIFAM
               EXECUTE MIJKA
               M(MINDEX)=0
3060         CONTINUE
3050       CONTINUE
3040     CONTINUE
         ENDBLOCK
C   REMOTE BLOCK MIJKA                            ******************
         REMOTE BLOCK MIJKA
         MINDA=0
         LI1=LI+1
         ISIGM =LI-1
         IF (ISIGM.EQ.0) GO TO 3064
```

```
        DO 3065 IX = 1,ISIGM
          MINDA=MINDA + (N-IX) * K
3065    CONTINUE
3064    MINDEX=MINDA+ (LJ-LI1)*K +LIFAM
        ENDBLOCK
C   REMOTE BLOCK OBJROW                          ******************
       REMOTE BLOCK OBJROW
       NM1= N-1
       DO 3100 I=1,NM1
         I1=I+1
         DO 3110 J=I1,N
           DO 3120 IFAM=1,K
               LI=I
               LJ=J
               LIFAM=IFAM
               EXECUTE MIJKA
               M(MINDEX) = COEFF(I,J)
3120       CONTINUE
3110     CONTINUE
3100   CONTINUE
       DO 3121 I=1,N
       DO 3121 IFAM=1,K
         XIND=(I-1)*K+IFAM
         X(XIND) =FCOEFF(I,IFAM)
        IF (I.EQ.1) X(XIND)=X(XIND)+C
3121   CONTINUE
       CTYPE='MIN'
       FLAG='OBJROW'
       EXECUTE PUTROW
       ENDBLOCK
C REMOTE BLOCK PUTROW                            ******************
       REMOTE BLOCK PUTROW
       XVARS=N*K
       MVARS=N*(N-1)/2*K
       WRITE(8,4000) (X(II),II=1,XVARS)
       WRITE(8,4002) (M(JJ),JJ=1,MVARS)
4002   FORMAT(9(' ',F7.2))
4000   FORMAT(9(' ',F7.2))
       IF (FLAG.EQ.'OBJROW') THEN
         WRITE(8,4010) CTYPE
4010     FORMAT(' ',A8,'  .')
       ELSEIF (FLAG.EQ.'ALLOC') THEN
         WRITE(8,4020) CTYPE,RHS
4020     FORMAT(' ',A8,I2)
       ELSEIF (FLAG.EQ.'CNSTR') THEN
         WRITE(8,4020) CTYPE,RHS
       ELSEIF (FLAG.EQ.'MLIM') THEN
         WRITE(8,4020) CTYPE,RHS
       ELSEIF (FLAG.EQ.'INTGER') THEN
         WRITE(8,4010) CTYPE
       ELSEIF (FLAG.EQ.'UPPER') THEN
         WRITE(8,4010) CTYPE
       ENDIF
       EXECUTE ALLCLR
       ENDBLOCK
C REMOTE BLOCK ALLOC                             ******************
       REMOTE BLOCK ALLOC
       FLAG='ALLOC'
       CTYPE='EQ'
       RHS=1
```

```
      DO 3200 I=1,N
        DO 3210 IFAM=1,K
          XIND= (I-1)*K+ IFAM
          X(XIND)=1
3210    CONTINUE
      EXECUTE PUTROW
3200  CONTINUE
      ENDBLOCK
C REMOTE BLOCK CNSTR                          ******************
      REMOTE BLOCK CNSTR
      FLAG='CNSTR'
      CTYPE='LE'
      RHS=1
      NM1=N-1
      DO 3230 I=1,NM1
        I1=I+1
        DO 3240 J=I1,N
          DO 3250 IFAM=1,K
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            M(MINDEX) = -1
            XINDI= (I-1) * K + IFAM
            XINDJ= (J-1) * K + IFAM
            X(XINDI)=1
            X(XINDJ)=1
            EXECUTE PUTROW
3250      CONTINUE
3240    CONTINUE
3230  CONTINUE
      ENDBLOCK
C REMOTE BLOCK CNSTR2                         ******************
      REMOTE BLOCK CNSTR2
      FLAG='CNSTR'
      CTYPE='LE'
      NM1=N-1
      DO 3400 I=1,NM1
        DO 3410 IFAM=1,K
          XIND=(I-1)*K + IFAM
          RHS=N-I
          X(XIND)=N-I
          I1=I+1
          DO 3420 J=I1,N
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            M(MINDEX)=-1
            XINDJ=(J-1)*K+IFAM
            X(XINDJ)=1
3420      CONTINUE
          EXECUTE PUTROW
3410    CONTINUE
3400  CONTINUE
      ENDBLOCK
C REMOTE BLOCK MLIM                           ******************
      REMOTE BLOCK MLIM
      FLAG='MLIM'
      CTYPE='LE'
```

```
      RHS=0
      NM1=N-1
      DO 3260 I=1,NM1
        I1=I+1
        DO 3270 J=I1,N
          DO 3280 IFAM=1,K
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            M(MINDEX)=1
            XINDI= (I-1)*K + IFAM
            X(XINDI)=-1
            EXECUTE PUTROW
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            XINDJ=(J-1)*K + IFAM
            M(MINDEX)=1
            X(XINDJ)=-1
            EXECUTE PUTROW
3280      CONTINUE
3270    CONTINUE
3260  CONTINUE
      ENDBLOCK
C REMOTE BLOCK MLIM2                                     ******************
      REMOTE BLOCK MLIM2
      FLAG='MLIM'
      CTYPE='LE'
      RHS=0
      DO 3500 I=1,N
        DO 3510 IFAM=1,K
          XIND=(I-1)*K+IFAM
          X(XIND)=1-N
          DO 3520 J=1,N
            IF (J.EQ.I) GO TO 3520
            LI=I
            LJ=J
            LIFAM=IFAM
            EXECUTE MIJKA
            M(MINDEX)=1
3520      CONTINUE
          EXECUTE PUTROW
3510    CONTINUE
3500   CONTINUE
       ENDBLOCK
C REMOTE BLOCK INTGER                                    ******************
      REMOTE BLOCK INTGER
      FLAG='INTGER'
      CTYPE='INTEGER'
      DO 3300 I=1,N
        K11=K-1
        DO 3310 IFAM=1,K11
          XIND= (I-1) * K + IFAM
          X(XIND)=1
3310    CONTINUE
3300  CONTINUE
      EXECUTE PUTROW
      ENDBLOCK
```

```
C REMOTE BLOCK UPPER                         *************************
       REMOTE BLOCK UPPER
       FLAG='UPPER'
       CTYPE='UPPERBD'
       DO 3700 I=1,N
         DO 3710 IFAM=1,K
           XIND=(I-1)*K +IFAM
           X(XIND)=1
3710     CONTINUE
3700    CONTINUE
        NM1=N-1
        DO 3730 I=1,NM1
          I1=I+1
          DO 3740 J=I1,N
            DO 3750 IFAM=1,K
              LI=I
              LJ=J
              LIFAM=IFAM
              EXECUTE MIJKA
              M(MINDEX)=1
3750        CONTINUE
3740      CONTINUE
3730    CONTINUE
        EXECUTE PUTROW
        ENDBLOCK
9999    STOP
        END
$ENTRY
$IBSYS
$STOP
//
```

```
//PD    JOB  (R240,NU7,3,5),'GAJ',CLASS=A,REGION=2048K
//       EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//    DISP=(NEW,KEEP),SPACE=(TRK,(40,10)),
//     DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN   DD  *
$JOB   WATFIV                     REF. SECTION 4.3.2
C -----------------------------------------------------------
C CELL FORMATION IN FMS - PART FAMILY FORMATION PROBLEM
C -----------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C          GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
           UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 2Z2
C -----------------------------------------------------------
C THIS PROGRAM IMPLEMENTS THE ALGORITHM FOR FINDING THE
C UPPER BOUND AND LOWER BOUND FOR SUBPROBLEMS OF THE TYPE
C SOLVED IN THE ITERATIONS OF THE APPROXIMATION PROCEDURE
C CALCULATING THESE BOUNDS WITH DIFFERENT VALUES OF R
C ESTABLISHES THE REGION (L,U)' FOR THESE SUBPROBLEMS.
C -----------------------------------------------------------
C   THE  VARIABLE DECLARATION SECTION ******************
        INTEGER NUMOP(30),SIM(30,30),DISIM(30,30)
        INTEGER OPI,OPJ,A
        INTEGER XIND,XVARS,MVARS,XINDI,XINDJ
        INTEGER RHS
        INTEGER MK(30,3),MEM(30,3),NNN(3),FAMN(3),FAMD(3)
        INTEGER FSIM(30,3),FDISIM(30,3),NSIM(30,30)
        INTEGER NDISIM(30,30)
        REAL LAMDA,OP*4(30,50),COEFF(30,30),FCOEFF(30,3),M(500),X(500)
        REAL S(315),LBOUND,FAMC(3),MINC
        CHARACTER CTYPE*8,FLAG*8
        COMMON OP,NUMOP,SIM,DISIM,OPI,OPJ,COEFF,I,J,IFAM,K,
     *  L,NM,MP,MN,N,K1
C   THE DATA INPUT AND CALCULATION OF SIMILAR AND DISSIMILAR PROCESSES
C   BETWEEN THE PARTS                            ******************
        READ(5,10) N,K
 10     FORMAT(I2,1X,I2)
        DO 100 I=1,N
          READ(5,20) NUM,(OP(I,J),J=1,NUM)
          NUMOP(I)=NUM
 20       FORMAT(I2,10(1X,A4),/,(3X,10(A4,1X)))
          WRITE(6,20) NUMOP(I),(OP(I,J),J=1,NUM)
 100    CONTINUE
        NM1=N-1
        DO 200 I=1,NM1
        I1=I+1
        DO 300 J=I1,N
          SIM(I,J)=0
          NTERMI=NUMOP(I)
          NTERMJ=NUMOP(J)
          DO 400 OPI=1,NTERMI
            DO 500 OPJ=1,NTERMJ
              IF (OP(I,OPI).EQ.OP(J,OPJ)) SIM(I,J)=SIM(I,J)+1
 500        CONTINUE
 400      CONTINUE
C         PRINT, 'SIMILARITY BETWEEN',I,'AND',J,'IS',SIM(I,J)
          DISIM(I,J)=NUMOP(I)+NUMOP(J)-2*SIM(I,J)
C        PRINT,'DISSIMILARITY BETWEEN ',I, ' AND ',J,' IS ',DISIM(I,J)
 300     CONTINUE
 200     CONTINUE
```

```
C READING THE INITIAL CONFIGURATION
          LAMDA=1.00
          DO 2010 IFAM =1,K
            DO 2010 I=1,N
              MK(I,IFAM)=0
              MEM(I,IFAM)=0
2010        CONTINUE
            DO 2020 IFAM=1,K
              READ(5,2030) NUMF,(MK(I,IFAM),I=1,NUMF)
             WRITE(6,2036) NUMF,(MK(I,IFAM),I=1,NUMF)
2036         FORMAT('INITIAL CONFIGURATION',I2,10(1X,I2))
              NNN(IFAM)=NUMF
2030          FORMAT(I2,20(1X,I2))
2020        CONTINUE
           DO 2050 IFAM=1,K
             DO 2060 I=1,N
               LU=NNN(IFAM)
               DO 2070 LL=1,LU
                 IF (MK(LL,IFAM).EQ.I) MEM(I,IFAM) =1
2070           CONTINUE
2060         CONTINUE
2050       CONTINUE
            TFAMD=0.0
           DO 2200 IFAM=1,K
             N1=N-1
             FAMN(IFAM)=0
             FAMD(IFAM)=0
             DO 2240 I=1,N1
             I1=I+1
               DO  2240 J=I1,N
           FAMN(IFAM)=FAMN(IFAM) +MEM(I,IFAM)*MEM(J,IFAM)*DISIM(I,J)
                 FAMD(IFAM)=FAMD(IFAM) +MEM(I,IFAM)*MEM(J,IFAM)*SIM(I,J)
2240           CONTINUE
             TFAMD=TFAMD+FAMD(IFAM)
2200         CONTINUE
             DO 188 IFAM=1,K
               FAMC(IFAM)=FAMN(IFAM)/TFAMD
 188         CONTINUE
             MINC=0
             MF=1
             DO 2260 IFAM=1,K
               IF (FAMC(IFAM).GT.MINC) THEN
                 MF=IFAM
                 MINC=FAMC(IFAM)
               ENDIF
2260         CONTINUE
C              MF=3
             NEWN=NNN(MF)
             NEWN1=NEWN-1
             DO 2300 NI=1,NEWN1
               NI1=NI+1
               DO 2400 NJ=NI1,NEWN
                 I=MK(NI,MF)
                 J=MK(NJ,MF)
                 IF (I.LT.J) THEN
                   NSIM(NI,NJ)=SIM(I,J)
                   NDISIM(NI,NJ)=DISIM(I,J)
                 ENDIF
                 IF (I.GT.J) THEN
                   NSIM(NI,NJ)=SIM(J,I)
```

```
                NDISIM(NI,NJ)=DISIM(J,I)
              ENDIF
2400        CONTINUE
2300      CONTINUE
          DO 2401 I=1,N
            DO 2410 IFAM=1,K
              FSIM(I,IFAM)=0
              FDISIM(I,IFAM)=0
2410        CONTINUE
2401      CONTINUE
          DO 2430 NI=1,NEWN
            NPART=MK(NI,MF)
            DO 2440 IFAM=1,K
              IF (IFAM.EQ.MF) GO TO 2440
              DO 2460 J=1,N
               IF (J.EQ.NPART) GO TO 2460
               IF (J.GT.NPART) THEN
           FSIM(NI,IFAM) =FSIM(NI,IFAM) + SIM(NPART,J)*MEM(J,IFAM)
         FDISIM(NI,IFAM) =FDISIM(NI,IFAM) +DISIM(NPART,J)*MEM(J,IFAM)
               ELSE
             FSIM(NI,IFAM)=FSIM(NI,IFAM) +SIM(J,NPART)*MEM(J,IFAM)
           FDISIM(NI,IFAM)=FDISIM(NI,IFAM)+DISIM(J,NPART)*MEM(J,IFAM)
               ENDIF
2460          CONTINUE
2440        CONTINUE
2430      CONTINUE
          N1=N-1
          DO 2462 I=1,N
            I1=I+1
            DO 2470 J=I1,N
              SIM(I,J)=0
              DISIM(I,J)=0
2470        CONTINUE
2462      CONTINUE
      N=NEWN
      WRITE(8,4018)
4018  FORMAT(' BOUNDS ON THE OBJECTIVE FUNCTION FOR DIFFERENT VALUES
     * OF R')
      WRITE(8,4053)
      WRITE(8,4024)
4024  FORMAT( '  ')
      WRITE(8,4024)
      WRITE(8,4026) N
4026  FORMAT('  # OF PARTS= ',I2)
      WRITE(8,4027) K
4027  FORMAT('  # OF FAMILIES=',I2)
      WRITE(8,4024)
      WRITE(8,4029) N*(N-1)/2*K+N*K
4029  FORMAT('  # OF NON-ZERO COEFFICIENTS  =',I4)
      WRITE(8,4053)
      WRITE(8,4024)
      WRITE(8,4024)
       WRITE(8,9000)
9000   FORMAT('  R         # OF NEGATIVE   OBJECTIVE FUNCTION BOUNDS')
       WRITE(8,9001)
9001   FORMAT('               COEFFICIENTS    LOWERBOUND    UPPERBOUND')
       WRITE(8,9002)
9002   FORMAT('  -----     ------------    -----------   -----------')
       EXECUTE ALLCLR
       DO 18 LLAMDA=200,500,10
```

```
          LAMDA=LLAMDA/100.00
             EXECUTE FAMIL
          NM1=N-1
          DO 21 I=1,NM1
            I1=I+1
            DO 22 J=I1,N
              COEFF(I,J)=DISIM(I,J)-LAMDA*SIM(I,J)
 22         CONTINUE
 21       CONTINUE
          DO 89 JM=1,315
            S(JM)=0.0
89        CONTINUE
          EXECUTE OBJROW
 18       CONTINUE
          GO TO 9999
C REMOTE BLOCK FAMIL
            REMOTE BLOCK FAMIL
              NEWN1=NEWN-1
              DO 2480 NI=1,NEWN1
                NI1=NI+1
                DO 2490 NJ=NI1,NEWN
                  SIM(NI,NJ)=NSIM(NI,NJ)
                  DISIM(NI,NJ)=NDISIM(NI,NJ)
                  COEFF(NI,NJ)=DISIM(NI,NJ)-LAMDA*SIM(NI,NJ)
2490            CONTINUE
2480          CONTINUE
              DO 2550 I=1,NEWN
                DO 2560 IFAM=1,K
                  FCOEFF(I,IFAM)=FDISIM(I,IFAM)-LAMDA*FSIM(I,IFAM)
2560            CONTINUE
2550          CONTINUE
             C=0.0
             DO 58 IFAM=1,K
               IF (IFAM.EQ.MF) GO TO 58
               C=C+FAMN(IFAM)-LAMDA*FAMD(IFAM)
                IF(LAMDA.GT.0.15) GO TO 58
C       PRINT ,'CONSTANT ',IFAM,' IS ',FAMN(IFAM)-LAMDA*FAMD(IFAM)
 58          CONTINUE
           IF (LAMDA.GT.0.15) GO TO 76
G          PRINT,' THE CONSTANT FACTOR = ',C
          DO 45 I=1,N
           DO 46 IFAM=1,K
            IF (IFAM.EQ.MF) GO TO 46
C           PRINT,'FSIM(',I,IFAM,') =',FSIM(I,IFAM)
C           PRINT,'FDISIM(',I,IFAM,') =',FDISIM(I,IFAM)
46         CONTINUE
45      CONTINUE
 76        DUMM=0.00
           ENDBLOCK
C  REMOTE BLOCK ALLCLR                          ******************
          REMOTE BLOCK ALLCLR
          DO  3030 II=1,N
          DO  3030 IIFAM=1,K
            XIND= (II-1)*K + IIFAM
            X(XIND)=0
3030      CONTINUE
          NM1=N-1
          DO 3040 II=1,NM1
          II1=II+1
            DO 3050 JJ=II1,N
```

```
              DO 3060 IIFAM=1,K
                LI=II
                LJ=JJ
                LIFAM=IIFAM
                EXECUTE MIJKA
                M(MINDEX)=0
3060          CONTINUE
3050        CONTINUE
3040      CONTINUE
          ENDBLOCK
C     REMOTE BLOCK MIJKA                              ******************
          REMOTE BLOCK MIJKA
          MINDA=0
          LI1=LI+1
          ISIGM =LI-1
          IF (ISIGM.EQ.0) GO TO 3064
          DO 3065 IX = 1,ISIGM
            MINDA=MINDA + (N-IX) *K
3065      CONTINUE
3064      MINDEX=MINDA+ (LJ-LI1)*K +LIFAM
          ENDBLOCK
C     REMOTE BLOCK OBJROW                             ******************
         REMOTE BLOCK OBJROW
         NM1= N-1
          NEG=0
            MS=1
         DO 3100 I=1,NM1
           I1=I+1
           DO 3110 J=I1,N
             DO 3120 IFAM=1,K
                 LI=I
                 LJ=J
                 LIFAM=IFAM
                 EXECUTE MIJKA
                 M(MINDEX) = COEFF(I,J)
                 IF(COEFF(I,J).LT.0) NEG=NEG+1
                 IF(IFAM.EQ.1) THEN
                    S(MS)=COEFF(I,J)
                    MS=MS+1
                  ENDIF
3120         CONTINUE
3110       CONTINUE
3100     CONTINUE
         DO 312 I=1,N
           DO 312 IFAM=1,K
             XIND=(I-1)*K + IFAM
             X(XIND)=FCOEFF(I,IFAM)
             IF (I.EQ.1) X(XIND)=X(XIND)+C
312      CONTINUE
         CTYPE='MIN'
         FLAG='OBJROW'
         LPOS=N*(N-1)/2*K-NEG
         CALL SORT(315,S)
         IES=N/K
         IPAIRS=IES*(IES-1)/2*K+(N-IES*K)*IES
         MPAIRS=N*(N-1)/2
         LBOUND=0.000
         UBOUND=0.000
         ICOUNT=0
         DO 889 KI=1,315
```

```
           INTS=S(KI)*10000.000
          IF (INTS.NE.0) THEN
            ICOUNT=ICOUNT+1
          ELSE
            GO TO 889
          ENDIF
          IF (S(KI).LT.0.00) THEN
            LBOUND=LBOUND+S(KI)
          ENDIF
          IF (S(KI).GT.0.00) THEN
            IF(ICOUNT.LE.IPAIRS) LBOUND=LBOUND+S(KI)
          ENDIF
889     CONTINUE
        ICOUNT=0
        DO 899 KI=1,315
            INTS=S(315-KI+1)*10000.000
          IF (INTS.NE.0) THEN
            ICOUNT=ICOUNT+1
          ELSE
            GO TO 899
          ENDIF
          IF (S(315-KI+1).GT.0.00) THEN
            UBOUND=UBOUND+S(315-KI+1)
          ENDIF
          IF (S(315-KI+1).LT.0.00) THEN
            IF(ICOUNT.LE.IPAIRS) UBOUND=UBOUND+S(315-KI+1)
          ENDIF
899     CONTINUE
         DO 891 I=1,N
           TM=-100000000.00
           TL=+100000000.00
           DO 892 IFAM=1,K
             XIND=(I-1)*K +IFAM
             IF(X(XIND).GT.TM) TM= X(XIND)
             IF(X(XIND).LT.TL) TL=X(XIND)
892        CONTINUE
        ENDBLOCK
C REMOTE BLOCK PUTROW                               ******************
        REMOTE BLOCK PUTROW
        EXECUTE ALLCLR
        ENDBLOCK
9999     STOP
         END
$ENTRY
$IBSYS
$STOP
//
```

```
//PE   JOB  (R240,NU7,5,5),'GAJ',CLASS=A,REGION=2048K
//       EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//   DISP=(NEW,KEEP),SPACE=(TRK,(40,10)),
//    DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN  DD  *
$JOB   WATFIV                     REF. SECTION 5.2
C -------------------------------------------------------------
C CELL FORMATION IN FMS - MACHINE GROUP ALLOCATION
C -------------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C           GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
            UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 2Z2
C -------------------------------------------------------------
C THIS PROGRAM GENERATES THE INPUT FILE OF THE PROBLEM
C 'P1' FOR THE SAS/OR INTEGER PROGRAMMING ROUTINE.
C ALLOCATION OF MACHINE GROUPS TO THE PART FAMILIES FORMED.
C -------------------------------------------------------------
C  VARIABLE DECLARATION SECTION        ******************
        INTEGER N(50),P,OPN,A(30,3,30,30),NUMOP(3,30),ROUTES(30,3)
        INTEGER X(500),XIND,M(5000),IRR(30)
        INTEGER TABL(3,1300,50),IENTRY(3),JCOUNT(30),KCOUNT(30)
        INTEGER RPTF(3,1300),LID(99),RHS,NLID(15),CARQ,OPN1
        INTEGER ONEOBJ(15,3),ONEAVL(15,3,30)
        CHARACTER CTYPE*8,FLAG*8
C  READING THE OPERATION DATA FOR THE PARTS   ******************
        MAXOP=15
        MAXMLT=1300
        MAXWID=MAXOP+2+20
        MQ=MAXOP+1
        MSLN=MAXOP+2
        READ(5,10) NMACH,K,(N(IFAM),IFAM=1,K)
10      FORMAT(I2,2X,I2,10(2X,I2))
        WRITE(6,10) NMACH,K,(N(IFAM),IFAM=1,K)
        DO 100 IFAM=1,K
          NUM=N(IFAM)
          DO 110 J=1,NUM
            READ(5,20) OPN
            WRITE(6,20) OPN
20          FORMAT(I2)
            NUMOP(IFAM,J)=OPN
            DO 120 P=1,OPN
              READ(5,30) (A(J,IFAM,P,IM),IM=1,NMACH)
              WRITE(6,30) (A(J,IFAM,P,IM),IM=1,NMACH)
30            FORMAT(15(I1,1X))
120         CONTINUE
110       CONTINUE
100     CONTINUE
        DO 200 IFAM=1,K
          DO 210 ISL=1,MAXMLT
            RPTF(IFAM,ISL)=0
            DO 220 IWID=1,MAXWID
              TABL(IFAM,ISL,IWID)=0
220         CONTINUE
210       CONTINUE
          IENTRY(IFAM)=0
200     CONTINUE
        DO 4100 IFAM=1,K
          DO 4110 MMM=1,NMACH
            ONEOBJ(MMM,IFAM)=0
```

```
              NPARTS=N(IFAM)
              DO 4120 JJJ=1,NPARTS
                ONEAVL(MMM,IFAM,JJJ)=0
4120          CONTINUE
4110        CONTINUE
4100      CONTINUE
C  MAIN SEGMENT OF THE PROGRAM
          EXECUTE MTERMS
          EXECUTE CLRALL
          EXECUTE OBJROW
          EXECUTE AVAIL
          EXECUTE ALLOC
          EXECUTE TYP101
          EXECUTE TYP201
          EXECUTE INTGER
          EXECUTE UPPER
          EXECUTE OUT
          GO TO 9999
C REOTE BLOCK OBJROW
          REMOTE BLOCK OBJROW
          EXECUTE MCOBJ
          EXECUTE MLTOBJ
          FLAG='SPL'
          CTYPE='MAX'
          EXECUTE PUTROW
          ENDBLOCK
C REMOTE BLOCK MCOBJ
          REMOTE BLOCK MCOBJ
          DO 1000 MM=1,NMACH
            DO 1010 IIFAM=1,K
              XIND=(MM-1) *K  +IIFAM
              X(XIND) =ONEOBJ(MM,IIFAM)
1010        CONTINUE
1000      CONTINUE
          ENDBLOCK
C REMOTE BLOCK UPPER
          REMOTE BLOCK UPPER
          DO 700 MM=1,NMACH
            DO 710 IIFAM=1,K
              XIND= (MM-1)*K + IIFAM
              X(XIND)=1
710         CONTINUE
700       CONTINUE
          MIND=0
          DO 711 IFAM=1,K
            NENTRY=IENTRY(IFAM)
            DO 712 IE=1,NENTRY
              MIND=MIND+1
              M(MIND)=1
712         CONTINUE
711       CONTINUE
          FLAG='SPL'
          CTYPE='UPPERBD'
          EXECUTE PUTROW
          ENDBLOCK
C REMOTE BLOCK MTERMS
          REMOTE BLOCK MTERMS
          DO 2000 IFAM=1,K
            NUM=N(IFAM)
            DO 2010 J=1,NUM
```

```
              OPN=NUMOP(IFAM,J)
              EXECUTE GEN
2010        CONTINUE
2000      CONTINUE
          END BLOCK
C REMOTE BLOCK GEN
         REMOTE BLOCK GEN
         ROUTES(J,IFAM)=0
         IA=1
         P=1
         ID1=0
         ID2=0
         ID3=0
         ID4=0
         ID5=0
         ID6=0
         ID7=0
         ID8=0
         ID9=0
         ID10=0
         ID11=0
         ID12=0
         ID13=0
         ID14=0
         ID15=0
         DO 3010 ID1=1,NMACH
           IA=1
           IA=IA*(A(J,IFAM,1,ID1))
           IF (IA.EQ.0) THEN
             GO TO 3010
           ENDIF
           IF (OPN.EQ.1) THEN
             ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
             EXECUTE RTFILE
             GO TO 3010
           ENDIF
         DO 3020 ID2=1,NMACH
           IA= 1
           IA=IA*(A(J,IFAM,2,ID2))
           IF (IA.EQ.0) THEN
             GO TO 3020
           ENDIF
           IF (OPN.EQ.2) THEN
             ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
             EXECUTE RTFILE
             GO TO 3020
           ENDIF
         DO 3030 ID3=1,NMACH
           IA=1
           IA=IA*(A(J,IFAM,3,ID3))
           IF (IA.EQ.0) THEN
             GO TO 3030
           ENDIF
           IF (OPN.EQ.3) THEN
             ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
             EXECUTE RTFILE
             GO TO 3030
           ENDIF
         DO 3040 ID4=1,NMACH
           IA=1
```

```
      IA=IA*(A(J,IFAM,4,ID4))
      IF (IA.EQ.0) THEN
        GO TO 3040
      ENDIF
      IF (OPN.EQ.4) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE RTFILE
        GO TO 3040
      ENDIF
    DO 3050 ID5=1,NMACH
       IA=1
      IA=IA*(A(J,IFAM,5,ID5))
      IF (IA.EQ.0) THEN
        GO TO 3050
      ENDIF
      IF (OPN.EQ.5) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE RTFILE
        GO TO 3050
      ENDIF
    DO 3060 ID6=1,NMACH
       IA=1
      IA=IA*(A(J,IFAM,6,ID6))
      IF (IA.EQ.0) THEN
        GO TO 3060
      ENDIF
      IF (OPN.EQ.6) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE RTFILE
        GO TO 3060
      ENDIF
    DO 3070 ID7=1,NMACH
      IA=1
      IA=IA*(A(J,IFAM,7,ID7))
      IF (IA.EQ.0) THEN
        GO TO 3070
      ENDIF
      IF (OPN.EQ.7) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE RTFILE
        GO TO 3070
      ENDIF
    DO 3080 ID8=1,NMACH
      IA=1
      IA=IA*(A(J,IFAM,8,ID8))
      IF (IA.EQ.0) THEN
        GO TO 3080
      ENDIF
      IF (OPN.EQ.8) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE RTFILE
        GO TO 3080
      ENDIF
    DO 3090 ID9=1,NMACH
      IA=1
      IA=IA*(A(J,IFAM,9,ID9))
      IF (IA.EQ.0) THEN
        GO TO 3090
      ENDIF
      IF (OPN.EQ.9) THEN
```

```
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3090
   ENDIF
 DO 3100 ID10=1,NMACH
   IA=1
   IA=IA*(A(J,IFAM,10,ID10))
   IF (IA.EQ.0) THEN
     GO TO 3100
   ENDIF
   IF (OPN.EQ.10) THEN
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3100
   ENDIF
 DO 3110 ID11=1,NMACH
   IA=1
   IA=IA*(A(J,IFAM,11,ID11))
   IF (IA.EQ.0) THEN
     GO TO 3110
   ENDIF
   IF (OPN.EQ.11) THEN
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3110
   ENDIF
 DO 3120 ID12=1,NMACH
   IA=1
   IA=IA*(A(J,IFAM,12,ID12))
   IF (IA.EQ.0) THEN
     GO TO 3120
   ENDIF
   IF (OPN.EQ.P12) THEN
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3120
   ENDIF
 DO 3130 ID13=1,NMACH
   IA=1
   IA=IA*(A(J,IFAM,13,ID13))
   IF (IA.EQ.0) THEN
     GO TO 3130
   ENDIF
   IF (OPN.EQ.13) THEN
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3130
   ENDIF
 DO 3140 ID14=1,NMACH
   IA=1
   IA=IA*(A(J,IFAM,14,ID14))
   IF (IA.EQ.0) THEN
     GO TO 3140
   ENDIF
   IF (OPN.EQ.14) THEN
     ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
     EXECUTE RTFILE
     GO TO 3140
   ENDIF
 DO 3150 ID15=1,NMACH
```

```
      IA=1
      IA=IA*(A(J,IFAM,15,ID15))
      IF (IA.EQ.0) THEN
        GO TO 3150
      ENDIF
      IF (OPN.EQ.15) THEN
        ROUTES(J,IFAM)=ROUTES(J,IFAM)+1
        EXECUTE.RTFILE
        GO TO 3150
      ENDIF
3150  CONTINUE
3140  CONTINUE
3130  CONTINUE
3120  CONTINUE
3110  CONTINUE
3100  CONTINUE
3090  CONTINUE
3080  CONTINUE
3070  CONTINUE
3060  CONTINUE
3050  CONTINUE
3040  CONTINUE
3030  CONTINUE
3020  CONTINUE
3010  CONTINUE
      ENDBLOCK
C  REMOTE BLOCK RTFILE
      .REMOTE BLOCK RTFILE
      IDUP=0
      LID(1)=ID1
      LID(2)=ID2
      LID(3)=ID3
      LID(4)=ID4
      LID(5)=ID5
      LID(6)=ID6
      LID(7)=ID7
      LID(8)=ID8
      LID(9)=ID9
      LID(10)=ID10
      LID(11)=ID11
      LID(12)=ID12
      LID(13)=ID13
      LID(14)=ID14
      LID(15)=ID15
       DO 1 IXX=1,15
          NLID(IXX)=0
1      CONTINUE
        CARQ=OPN
        OPN1=OPN-1
       DO 2 IX=1,OPN1
          IF (LID(IX).EQ.999) GO TO 2
          IX1=IX+1
          DO 3 IY=IX1,OPN
              IF (LID(IY).EQ.LID(IX)) THEN
                 LID(IY)=999
                 CARQ=CARQ-1
              ENDIF
3          CONTINUE
2      CONTINUE
       NC=0
```

```
          ENDBLOCK
C REMOTE BLOCK MLTOBJ
          REMOTE BLOCK MLTOBJ
          MIND=0
          DO 2100 IFAM=1,K
            NENTRY=IENTRY(IFAM)
            DO 2110 IE=1,NENTRY
              MIND=MIND+1
              M(MIND)=RPTF(IFAM,IE)+1
2110        CONTINUE
2100      CONTINUE
          ENDBLOCK
C REMOTE BLOCK AVAIL
           REMOTE BLOCK AVAIL
          MINDF=0
          DO 2140 IFAM=1,K
            IF (IFAM.NE.1) MINDF=MINDF+IENTRY(IFAM-1)
            MIND=MINDF
            NENTRY=IENTRY(IFAM)
            NPARTS=N(IFAM)
            DO 2150 JJ=1,NPARTS
              DO 2160 IE=1,NENTRY
                MIND=MIND+1
                IREP=MSLN+RPTF(IFAM,IE)+1
C              IF (IFAM.EQ.1) THEN
C                IF(IE.EQ.3) PRINT,'RPTF+1',RPTF(IFAM,IE)+1
C              ENDIF
                IB = MSLN+1
                MCOEF=0
C               DO 2170 IX=IB,IREP
C                 IF (TABL(IFAM,IE,IX).EQ.JJ)  MCOEF=MCOEF+1
C2170            CONTINUE
                 JPOS=MSLN+JJ
                 MCOEF=TABL(IFAM,IE,JPOS)
                M(MIND)=MCOEF
2160          CONTINUE
              DO 4600 MMM=1,NMACH
                XIND=(MMM-1)*K+IFAM
                X(XIND)=ONEAVL(MMM,IFAM,JJ)
4600          CONTINUE
              CTYPE='GE'
              RHS=1
              EXECUTE PUTROW
              MIND=MINDF
2150        CONTINUE
2140      CONTINUE
          ENDBLOCK
C REMOTE BLOCK TYP101
          REMOTE BLOCK TYP101
          MIND=0
          DO 2300 IFAM=1,K
            NENTRY=IENTRY(IFAM)
            DO 2310 IE=1,NENTRY
              MIND=MIND+1
              M(MIND)=-1
              OPN=TABL(IFAM,IE,MQ)
              DO 2320 IJ=1,OPN
                IX=TABL(IFAM,IE,IJ)
                XIND=(IX-1)*K +IFAM
                X(XIND)=X(XIND)+1
```

```
            DO 4 IX=1,OPN
               IF (LID(IX).NE.999) THEN
                  NC=NC+1
                  NLID(NC)=LID(IX)
               ENDIF
4           CONTINUE
            IF (NC.NE.CARQ) PRINT,'********ERROR**********'
C           PRINT,'J,IFAM,NC',J,IFAM,NC
             IF (CARQ.EQ.1) THEN
               DO 4500 IXY=1,OPN
                 IF (LID(IXY).NE.999) THEN
                   MMM=LID(IXY)
                   ONEOBJ(MMM,IFAM)=ONEOBJ(MMM,IFAM)+1
                   ONEAVL(MMM,IFAM,J)=ONEAVL(MMM,IFAM,J)+1
                  ENDIF
 4500            CONTINUE
                 GO TO 8
              ENDIF
         IF (IENTRY(IFAM).EQ.0) THEN
           GO TO 6500
         ELSE
           LIENT=IENTRY(IFAM)
           DO 6000 ICHK=1,LIENT
             IF (IDUP.EQ.1) GO TO 6500
             IF (TABL(IFAM,ICHK,MQ).NE.CARQ) GO TO 6000
             DO 6010 JM=1,NMACH
               JCOUNT(JM)=0
               DO 6020 IX=1,CARQ
                 IF (NLID(IX).EQ.JM) JCOUNT(JM)=JCOUNT(JM)+1
6020           CONTINUE
               KCOUNT(JM)=0
               DO 6030 IY=1,CARQ
                 IF (TABL(IFAM,ICHK,IY).EQ.JM) KCOUNT(JM)=KCOUNT(JM)+1
6030           CONTINUE
               IF (JCOUNT(JM).NE.KCOUNT(JM)) GO TO 6000
6010         CONTINUE
             IDUP=1
             IDUPSL=ICHK
6000       CONTINUE
          ENDIF
6500      IF (IDUP.EQ.1) THEN
            RPTF(IFAM,IDUPSL)=RPTF(IFAM,IDUPSL)+1
C          JPOS=MSLN+1+RPTF(IFAM,IDUPSL)
C           TABL(IFAM,IDUPSL,JPOS)=J
            JPOS=MSLN+J
            TABL(IFAM,IDUPSL,JPOS)=TABL(IFAM,IDUPSL,JPOS)+1
          ELSE
            IENTRY(IFAM)=IENTRY(IFAM)+1
            NENTRY=IENTRY(IFAM)
            DO 6550 IX=1,CARQ
              TABL(IFAM,NENTRY,IX)=NLID(IX)
6550        CONTINUE
            TABL(IFAM,NENTRY,MQ)=CARQ
            TABL(IFAM,NENTRY,MSLN)=NENTRY
            JFIRST=MSLN+1
C           TABL(IFAM,NENTRY,JFIRST)=J
            JPOS=MSLN+J
            TABL(IFAM,NENTRY,JPOS)=TABL(IFAM,NENTRY,JPOS)+1
          ENDIF
8            DUZM=0.00
```

```
2320          CONTINUE
              RHS=OPN-1
              CTYPE='LE'
              EXECUTE PUTROW
2310        CONTINUE
2300      CONTINUE
          ENDBLOCK
C REMOTE BLOCK TYP201
          REMOTE BLOCK TYP201
          MIND=0
          DO 2400 IFAM=1,K
            NENTRY=IENTRY(IFAM)
            DO 2410 IE=1,NENTRY
              MIND=MIND+1
              OPN=TABL(IFAM,IE,MQ)
               DO 954 MREP=1,NMACH
                 IRR(MREP)=0
 954           CONTINUE
              DO 2420 IJ=1,OPN
                M(MIND)=1
                IX=TABL(IFAM,IE,IJ)
                IF (IRR(IX).EQ.0) THEN
                  XIND=(IX-1)*K +IFAM
                  X(XIND)=-1
                  CTYPE='LE'
                  RHS=0
                  EXECUTE PUTROW
                ENDIF
                IRR(IX)=1
2420          CONTINUE
2410        CONTINUE
2400      CONTINUE
          ENDBLOCK
C REMOTE BLOCK INTGER
          REMOTE BLOCK INTGER
          DO 2600 INTM=1,NMACH
            INTK=K-1
            DO  2610 INTFAM=1,INTK
              IXIND=(INTM-1)*K +INTFAM
              X(IXIND)=1
2610        CONTINUE
2600      CONTINUE
          CTYPE = 'INTEGER'
          FLAG='SPL'
          EXECUTE PUTROW
          ENDBLOCK
C REMOTE BLOCK PUTROW
          REMOTE BLOCK PUTROW
          IXTRMS = NMACH*K
          IMTRMS=0
          DO 2500 IIFAM=1,K
            IMTRMS=IMTRMS+IENTRY(IIFAM)
2500      CONTINUE
          WRITE(8,7000) (X(IPR),IPR=1,IXTRMS),(M(IPR),IPR=1,IMTRMS)
7000      FORMAT(20(1X,I3))
          IF (FLAG.EQ.'SPL') THEN
            WRITE(8,7010) CTYPE
7010        FORMAT(1X,A8,'   .')
          ELSE
            WRITE(8,7020) CTYPE,RHS
```

```
7020       FORMAT(1X,A8,1X,I2)
          ENDIF
          FLAG='REG'
          EXECUTE CLRALL
          ENDBLOCK
C REMOTE   BLOCK CLRALL
          REMOTE BLOCK CLRALL
          DO 2700 MCLR=1,NMACH
            DO 2710 ICLRF=1,K
              XIND=(MCLR-1)*K +ICLRF
              X(XIND)=0
2710        CONTINUE
2700      CONTINUE
          IMTRMS=0
          DO 2730 ICLRF=1,K
            IMTRMS=IMTRMS+IENTRY(ICLRF)
2730      CONTINUE
          DO 2750 LMIND=1,IMTRMS
            M(LMIND)=0
2750      CONTINUE
          FLAG='REG'
          ENDBLOCK
          REMOTE BLOCK ALLOC
          DO 5000 IIM=1,NMACH
            DO 5010 IFAM=1,K
              XIND    =(IIM-1)*K + IFAM
              X(XIND) = 1
5010        CONTINUE
          RHS=1
          CTYPE='EQ'
          IF (IIM.EQ.2) THEN
              RHS = 2
          ENDIF
          IF (IIM.EQ.6) RHS=2
          EXECUTE PUTROW
5000      CONTINUE
          ENDBLOCK
9999      STOP
          END
$ENTRY
12  03  06  04  05
03       PART 1
0 0 0 0 1 0 1 0 0 0 1 0
1 0 0 0 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 0 0 1 0
02       PART 3
0 0 0 0 1 1 0 1 0 1 0 0
0 0 0 0 0 1 0 0 0 1 0 1
03       PART 4
0 0 0 0 0 0 1 0 1 0 0 0
0 1 0 0 1 0 0 0 0 0 0 1
0 0 0 1 0 1 0 0 0 0 0 0
03      PART 5
0 1 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
03    PART 8
1 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0
```

```
03    PART 13
0 1 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 0 0 1 0 0 0
02    PART 6
1 0 0 0 0 0 0 0 0 0 1 0
0 0 0 1 1 1 0 0 1 0 0 0
03    PART 11
1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0 1 0 0 0
03    PART 12
1 0 0 1 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 1 0 0 0
03     PART 15
1 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 1 0
02   PART 2
1 0 0 1 0 1 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 1 0 0
04    PART 7
0 1 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 0 0 0 0 1
0 0 0 1 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0
03    PART 9
0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 0
02   PART 10
0 1 0 0 0 1 0 0 0 1 0 1
0 0 0 1 0 0 1 0 0 0 0 0
04   PART 14
0 0 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0 1 0 0
0 1 0 1 0 0 0 0 0 0 0 0
$IBSYS
$STOP
//
```

```
//PF   JOB  (R240,NU7,5,5),'GAJ',CLASS=A,REGION=2048K
//        EXEC  WATFIV
//FT08F001 DD DSN=WYL.R240NU7.FAMOUT,UNIT=DASD,VOL=SER=WORKPK,
//     DISP=(NEW,KEEP),SPACE=(TRK,(40,10)),
//      DCB=(LRECL=80,BLKSIZE=15440,RECFM=FB)
//GO.SYSIN   DD  *
$JOB   WATFIV                    REF. SECTION 5.3.2
C -----------------------------------------------------------
C CELL FORMATION IN FMS - MACHINE GROUP ALLOCATION
C -----------------------------------------------------------
C  AUTHOR - GAJANANA NADOLI
C          GRADUATE STUDENT, DEPT. OF INDUSTRIAL ENGINEERING
C          UNIVERSITY OF WINDSOR, WINDSOR, ONTARIO N9B 3Z2
C -----------------------------------------------------------
C THIS PROGRAM GENERATES THE INPUT FILE OF THE PROBLEM
C 'INF' FOR THE SAS/OR INTEGER PROGRAMMING ROUTINE.
C IDENTIFICATION OF THE MACHINES CAUSING INFEASIBILITY IN
C MACHINE GROUP ALLOCATION.
C -----------------------------------------------------------
C  VARIABLE DECLARATION SECTION       ******************
         INTEGER N(50),P,OPN,A(30,3,30,30),NUMOP(3,30),ROUTES(30,3)
         INTEGER X(500),XIND,M(5000),IRR(30),XINDJ,XINDK
         INTEGER TABL(3,1300,50),IENTRY(3),JCOUNT(30),KCOUNT(30)
         INTEGER RPTF(3,1300),LID(99),RHS,NLID(15),CARQ,OPN1
         INTEGER ONEOBJ(15,3),ONEAVL(15,3,30)
         CHARACTER CTYPE*8,FLAG*8
C  READING THE OPERATION DATA FOR THE PARTS  ******************
         IMPOS=2
         MAXOP=15
         MAXMLT=1300
         MAXWID=MAXOP+2+20
         MQ=MAXOP+1
         MSLN=MAXOP+2
         READ(5,10) NMACH,K,(N(IFAM),IFAM=1,K)
10       FORMAT(I2,2X,I2,10(2X,I2))
         WRITE(6,10) NMACH,K,(N(IFAM),IFAM=1,K)
         DO 100 IFAM=1,K
           NUM=N(IFAM)
           DO 110 J=1,NUM
             READ(5,20) OPN
             WRITE(6,20) OPN
20           FORMAT(I2)
             NUMOP(IFAM,J)=OPN
             DO 120 P=1,OPN
               READ(5,30) (A(J,IFAM,P,IM),IM=1,NMACH)
               WRITE(6,30) (A(J,IFAM,P,IM),IM=1,NMACH)
30             FORMAT(15(I1,1X))
120          CONTINUE
110        CONTINUE
100      CONTINUE
         DO 200 IFAM=1,K
           DO 210 ISL=1,MAXMLT
             RPTF(IFAM,ISL)=0
             DO 220 IWID=1,MAXWID
               TABL(IFAM,ISL,IWID)=0
220          CONTINUE
210        CONTINUE
           IENTRY(IFAM)=0
200      CONTINUE
         DO 4100 IFAM=1,K
```

```
220          CONTINUE
210        CONTINUE
           IENTRY(IFAM)=0
200      CONTINUE
         DO 4100 IFAM=1,K
           DO 4110 MMM=1,NMACH
             ONEOBJ(MMM,IFAM)=0
             NPARTS=N(IFAM)
             DO 4120 JJJ=1,NPARTS
               ONEAVL(MMM,IFAM,JJJ)=0
4120         CONTINUE
4110       CONTINUE
4100     CONTINUE
         MTRMS=NMACH*K*(K-1)/2 +NMACH
C  MAIN SEGMENT OF THE PROGRAM
          EXECUTE CLRALL
          EXECUTE NEWOBJ
          EXECUTE MIN1
          EXECUTE ALLOC
          EXECUTE LIN1
          EXECUTE INTGER
          EXECUTE UPPER
          GO TO 9999
C REMOTE BLOCK NEWOBJ
          REMOTE BLOCK NEWOBJ
          DO 1001 MM=1,NMACH
            DO 1011 IIFAM=1,K
              XIND=(MM-1) * K + IIFAM
              X(XIND)=1
1011        CONTINUE
1001      CONTINUE
          MIND=1
          DO 8000 MMS=1,NMACH
            K1=K-1
            DO 8010 JFAM=1,K1
              K2=JFAM+1
              DO 8020 KFAM=K2,K
                M(MIND)=37
                MIND=MIND+1
8020          CONTINUE
8010        CONTINUE
8000      CONTINUE
          DO 8030 MMS=1,NMACH
            M(MIND)=1369
            MIND=MIND+1
8030      CONTINUE
8500      DUMMM=0.00
          FLAG='SPL'
          CTYPE='MIN'
          EXECUTE PUTROW
          ENDBLOCK
C REMOTE BLOCK MIN1
          REMOTE BLOCK MIN1
          DO 9000 IFAM=1,K
            NUM =N(IFAM)
```

```
              DO 9010 J=1,NUM
                OPN=NUMOP(IFAM,J)
                DO 9020 P=1,OPN
                  DO 9030 MMS=1,NMACH
                    XIND=(MMS-1) * K + IFAM
                    X(XIND)=A(J,IFAM,P,MMS)
9030              CONTINUE
                  CTYPE='GE'
                  RHS=1
                  EXECUTE PUTROW
9020            CONTINUE
9010          CONTINUE
9000          CONTINUE
             ENDBLOCK
C REMOTE BLOCK UPPER
             REMOTE BLOCK UPPER
             DO 700 MM=1,NMACH
               DO 710 IIFAM=1,K
                 XIND= (MM-1)*K + IIFAM
                 X(XIND)=1
710            CONTINUE
700          CONTINUE
             MIND=1
             DO 8005 MMS=1,NMACH
               K1=K-1
               DO 8015 JFAM=1,K1
                 K2=JFAM+1
                 DO 8025 KFAM=K2,K
                   M(MIND)=1
                   MIND=MIND+1
8025             CONTINUE
8015           CONTINUE
8005         CONTINUE
             DO 8035 MMS=1,NMACH
               M(MIND)=1
               MIND=MIND+1
8035         CONTINUE
8505         DUMMM=0.00
C            MIND=0
C            DO 711 IFAM=1,K
C              NENTRY=IENTRY(IFAM)
C              DO 712 IE=1,NENTRY
C                MIND=MIND+1
C                M(MIND)=1
C712            CONTINUE
C711          CONTINUE
             FLAG='SPL'
             CTYPE='UPPERBD'
             EXECUTE PUTROW
             ENDBLOCK
            REMOTE BLOCK LIN1
            KMIND=1
            DO 8002 LMMS=1,NMACH
              K1=K-1
              DO 8012 LJFAM=1,K1
```

```
            K2=LJFAM+1
            DO 8022 LKFAM=K2,K
              XINDJ=(LMMS-1)*K + LJFAM
              XINDK=(LMMS-1)*K + LKFAM
              X(XINDJ)=1
              X(XINDK)=1
              M(KMIND)=-1
             CTYPE='LE'
             RHS= 1
             EXECUTE PUTROW
             X(XINDJ)=-1
             M(KMIND)=1
             CTYPE='LE'
             RHS=0
             EXECUTE PUTROW
             X(XINDK)=-1
             M(KMIND)=1
              CTYPE='LE'
              RHS=0
             EXECUTE PUTROW
              KMIND=KMIND+1
8022        CONTINUE
8012       CONTINUE
8002      CONTINUE
          DO 8815 KMM=1,NMACH
            M(KMIND)=-1
            DO 8018 IFN=1,K
              IXIND=(KMM-1)*K + IFN
              X(IXIND)=1
8018        CONTINUE
            RHS=2
            CTYPE='LE'
            EXECUTE PUTROW
           DO 8609 IFN=1,K
            M(KMIND)=1
             IXIND=(KMM-1)*K +IFN
             X(IXIND)=-1
             RHS=0
             CTYPE='LE'
             EXECUTE PUTROW
8609       CONTINUE
           KMIND=KMIND+1
8815      CONTINUE
           ENDBLOCK
C REMOTE BLOCK INTGER
           REMOTE BLOCK INTGER
           DO 2600 INTM=1,NMACH
             INTK=K-1
             DO  2610 INTFAM=1,INTK
               IXIND=(INTM-1)*K +INTFAM
               X(IXIND)=1
2610         CONTINUE
2600       CONTINUE
           CTYPE = 'INTEGER'
           FLAG='SPL'
           EXECUTE PUTROW
           ENDBLOCK
```

```
              EXECUTE PUTROW
              ENDBLOCK
C REMOTE     BLOCK PUTROW
             REMOTE BLOCK PUTROW
             IXTRMS = NMACH*K
C            IMTRMS=0
C            DO 2500 IIFAM=1,K
C              IMTRMS=IMTRMS+IENTRY(IIFAM)
C2500         CONTINUE
             WRITE(8,7000) (X(IPR),IPR=1,IXTRMS),(M(JPR),JPR=1,MTRMS)
7000         FORMAT(15(1X,I4))
             IF (FLAG.EQ.'SPL') THEN
               WRITE(8,7010) CTYPE
7010           FORMAT(1X,A8,'  .')
             ELSE
               WRITE(8,7020) CTYPE,RHS
7020           FORMAT(1X,A8,1X,I2)
             ENDIF
             FLAG='REG'
             EXECUTE CLRALL
             ENDBLOCK
C REMOTE     BLOCK CLRALL
             REMOTE BLOCK CLRALL
             DO 2700 MCLR=1,NMACH
               DO 2710 ICLRF=1,K
                 XIND=(MCLR-1)*K +ICLRF
                 X(XIND)=0
2710           CONTINUE
2700         CONTINUE
             MIND=1
             DO 8001 MMS=1,NMACH
               K1=K-1
               DO 8011 JFAM=1,K1
                 K2=JFAM+1
                 DO 8021 KFAM=K2,K
                   M(MIND)=0
                   MIND=MIND+1
8021             CONTINUE
8011           CONTINUE
8001          CONTINUE
              DO 8031 MMS=1,NMACH
                M(MIND)=0
                MIND=MIND+1
8031          CONTINUE
8501          DUMMM=0.00
             REMOTE BLOCK ALLOC
             DO 5000 IIM=1,NMACH
               DO 5010 IFAM=1,K
                 XIND    =(IIM-1)*K + IFAM
                 X(XIND) = 1
5010           CONTINUE
             RHS=3
             CTYPE='LE'
             EXECUTE PUTROW
5000         CONTINUE
```

```
            ENDBLOCK
9999        STOP
            END
$ENTRY
$IBSYS
$STOP
//
```

APPENDIX D

ITERATION LOGS FOR DIFFERENT TRIALS OF APPROXIMATON PROCEDURE

TABLE 9 b

Iteration Log for the Approximation Procedure

Number of Parts = 15

Random Starting Partition = 2

ITR No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
0	[1,2,11,12,13,14] [3,4,9,10,15] [5,6,7,8] MF=1 CDC=3.697	* Based on range analysis choose R=3.50				
		3.55	[2,11] [3,4,9,10,15] [5,6,7,8,1,12,13,14]	-77.5	3.283	High, Choose 3.283
		3.283	Same as above	0.01	3.283	OK
1	[2,11] [3,4,9,10,15] [5,6,7,8,1,12,13,14] MF=3 CDC=3.283	* Based on range analysis choose R=3.05				
		3.05	[2,11] [3,4,9,10,15,1,5,7,13,14] [6,8,12]	1.8	3.053	Low, Choose 3.053
		3.053	Same as above	0.01	3.053	OK
2	[2,11] [3,4,9,10,15,1,5,7,13,14] [6,8,12] MF=2 CDC=3.053	* Based on the range analysis choose R = 2.70				
		2.70	[2,11,15] [5,7,9,10,13,14] [6,8,12,1,3,4]	73.7	2.931	R low Choose 2.93
		2.93	[2,11] [1,4,5,7,9,10,13,14] [6,8,12,3,15]	-0.10	2.93	OK

(Contd.)

TABLE 9 b (Continued)

Itr No.	INITIAL ALLOCATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
3	[2,11] [1,4,5,7,9,10, 13,14] [3,6,8,12,15] MF=3 CDC=2.93	* Based on range analysis choose R=2.775				
		2.775	[2,11] [5,7,9,10,13, 14] [3,6,8,12,15, 1,4]	55.1	2.927	Low, Choose 2.928
		2.928	Same as above	0	2.928	OK
4	[2,11] [5,7,9,10,13, 14] [3,6,8,12,15, 1,4] MF=3 CDC=2.93	* Based on range analysis choose R = 2.93				
		2.93	[2,11,12] [1,4,5,7,9, 10,13,14] [3,6,8,15]	-12.2	2.895	Low, Choose 2.895
		2.895	Same as above	-0.02	2.895	OK
5	[2,11,12] [1,4,5,7,9, 10,13,14] [3,6,8,15] MF=2 CDC=2.895	* Based on range analysis choose R=2.80				
		2.80	[2,11,12] [5,7,9,10,13, 14] [3,6,8,15,1,4]	22.6	2.869	Low, Choose 2.869
		2.869	Same as above	-0.01	2.869	OK
		Refer iteration 4, step 2 onwards in RSP #1				

TABLE 9 c

Iteration Log for the Approximation Procedure

Number of Parts = 15

Random Starting Partition = 3

ITR No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
0	[1,5,7,9,13,14] [3,6,11,15] [2,4,8,10,12]	* Based on range analysis choose R=3.10				
	MF=1 CDC=3.157	3.10	[7,13,14] [3,6,11,15] [2,4,8,10,12, 1,5,9]	7.90	3.124	Low, Choose 3.124
		3.124	Same as above	0	3.124	OK
1	[7,13,14] [3,6,11,15] [2,4,8,10,12, 1,5,9]	* Based on range analysis choose R=2.90				
	MF=3 CDC=3.124	2.90	[7,13,14] [3,6,11,15] [2,4,8,10,12, 1,5,9]	22.4	2.961	Low, Choose 2.961
		2.961	Same as above	0	2.961	OK
2	[1,4,5,7,8,9, 13,14] [3,6,11,15] [2,10,12]	* Based on the range analysis choose R = 2.825				
	MF=1 CDC=2.961	2.825	[1,4,5,8,9, 13] [3,6,11,15] [2,10,12,7,14]	15.56	2.876	Low, Choose 2.876
		2.876	Same as above	0.01	2.876	OK

(Contd.)

TABLE 9 c (continued)

Itr No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
3	[1,4,5,8,9,13] [3,6,11,15] [2,7,10,12,14] MF=1 CDC=2.876	* Based on range analysis choose R=2.876				
		2.876	Same as Initial	0.02	2.876	OK
		Family configuration not changed				
		Choose the family with next highest D_k i.e., MF=3. Based on the range analysis R=2.80				
		2.80	[1,4,5,8,9,13] [3,6,11,12,15] [2,7,10,14]	22.90	2.875	Low Choose 2.875
		2.875	Same as above	-0.01	2.875	OK
4	[1,4,5,8,9,13] [3,6,11,12,15] [2,7,10,14] MF=2 CDC=2.875	* Based on range analysis choose R=2.75				
		2.75	[1,4,8,5,9,13,3] [6,11,12,15] [2,7,10,14]	22.2	2.819	Low, Choose 2.818
		2.818	[1,3,4,5,8,9,13 [6,11,12,15] [2,7,10,14]	0	2.818	OK

(Contd.)

TABLE 9 c (Continued)

Itr No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
5	[1,3,4,5,8, 9,13] [6,11,12,15] [2,7,10,14] MF=1 CDC=2.818	* Based on range analysis choose R=2.725				
		2.725	[1,3,4,5,8,13] [6,11,12,15] [2,7,9,14]	23	2.799	Low, Choose 2.799
		2.799	Same as above	0	2.799	OK
6	[1,3,4,5,8,13] [6,11,12,15] [2,7,9,10,14] MF=1 CDC=2.799	* Based on range analysis choose R=2.799				
		2.799	Same as initial	0	2.799	OK
		Family configueration,not changed.				
		Choose the family with next highest D_k i.e., MF=3. Based on range analysis choose R=2.799				
		2.799	Same as initial	0	2.799	OK
		Family configuration not changed				
		Choose MF=2 and based on range analysis R=2.799				
		2.799	Same as initial	0	2.799	OK
		Family configuration not changed.				
		All families considered for reallocation STOP				

TABLE 9 d

Iteration Log for the Approximation Procedure

Number of Parts = 15

Random Starting Partition = 4

Itr No.	INITIAL ALLOCATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
0	[1,5,8,10,15] [3,9,7,13,14] [2,4,6,11,12] MF=2 CDC=3.44	* Based on range analysis choose R=3.30				
		3.30	[1,5,8,10,15, 7,9,13,14] [3] [2,4,6,11,12]	-40.5	3.20	R high Choose 3.20
		3.20	Same as above	0	3.20	OK
1	[1,5,7,8,9,10, 13,14,15] [3] [2,4,6,11,12] MF=1 CDC=3.204	* Based on range analysis choose R = 3.05				
		3.05	[5,7,9,10,13, 14] [1,3,8,15] [2,4,6,11,12]	15.7	3.105	Low, Choose 3.105
		3.105	Same as above	0.02	3.105	OK
2	[5,7,9,10,13, 14] [1,3,8,15] [2,4,6,11,12] MF=1 CDC=3.104	* Based on range analysis choose 3.105				
		3.104	Same as initial Family configuration not changed.	0.02	3.105	OK
		Choose the family with next highest D_k i.e., MF=3. Based on the range analysis R = 2.95				

(Contd.)

TABLE 9 d (Continued)

Itr No.	INITIAL ALLOC-ATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
		2.95	[5,7,9,10,13, 14] [1,3,8,15,4] [2,6,11,12]	-31.7	2.846	High, Choose 2.846
		2.846	Same as above	-0.01	2.846	OK
		Refer to Iteration 5 onwards in RSP #1				

TABLE 9 e

Iteration Log for the Approximation Procedure

Number of Parts = 15

Random Starting Partition = 5

ITR No.	INITIAL ALLOCATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
0	[1,3,9,12,13] [4,10,11,5,7] [2,6,8,14,15] MF=2 CDC=3.665	* Based on range analysis choose R=3.35				
		3.35	[1,3,9,12,13, 4,5,7,10] [11] [2,6,8,14,15]	-60	3.209	High, Choose 3.209
		3.209	[1,3,9,12,13, 4,5] [7,10,11] [2,6,8,14,15]	-3.7	3.197	High, Choose 3.197
		3.197	Same as above	0	3.197	OK
1	[1,3,4,5,9, 12,13] [7,10,11] [2,6,8,14,15] MF=1 CDC=3.197	* Based on range analysis choose R=3.10				
		3.10	[1,3,4,5,9,13] [7,10,11,12] [2,6,8,14,15]	6.6	3.123	Low, Choose 3.123
		3.123	Same as above	-0.01	3.123	OK
2	[1,3,4,5,9,13] [7,10,11,12] [2,6,8,14,15] MF=3 CDC=3.665	* Based on range analysis choose R=2.95				
		2.95	[1,3,4,5,9,13, 8] [7,10,11,12,14] [2,6,15]	0.4	2.95	OK ,

(Contd.)

TABLE 9 e (Continued)

ITR No.	INITIAL ALLOCATION	R	INTEMEDIATE AND FINAL ALLOCATIONS	OBJ FN. Z(R,X)	NEW CDC	Comments
3	[1,3,4,5,8,9,13] [7,10,11,12,14] [2,6,15] MF=1 CDC=2.95	* Based on range analysis choose R=85				
		2.85	Same as initial	33	2.95	Low, Choose 2.95
		2.95	Same as above	0.03	2.95	OK
		Family configuration not changed.				
		Choose family with the next highest D_k, i.e., MF=2. Based on range analysis Take R=2.85				
		2.85	[1,3,4,5,8,9,13] [7,10,14] [11,12,2,6,15]	52	2.86	Low, Choose 2.86
		2.86	Same as above	0	2.86	OK
4	[1,3,4,5,8,9,13] [7,10,14] [11,12,2,6,15] MF=1 CDC=2.861	* Based on range analysis choose R=2.80				
		2.80	[1,3,4,8,13] [7,10,14,5,9] [11,12,2,6,15]	10.4	2.835	Low, Choose
		2.835	[1,3,4,5,8,13] [7,10,14,9] [11,12,2,6,15]	-0.35	2.835	OK
5	[1,3,4,5,8,13] [7,10,14,9] [11,12,2,6,15] MF=3 CDC=2.835	* Based on the range analysis choose R = 2.775				
		2.775	[1,3,4,5,8,13] [7,9,10,14,2] [6,11,12,15]	7.6	2.799	
		Refer to iteration 6 of RSP #3				

VITA AUCTORIS

1958 Born in Hosmat, India, on 14th of December.

1974 Completed higher secondary education from Government High School, Kadaba, India.

1976 Completed Pre-University Course from Sri. Bhuvanendra College, Karkala, India.

1981 Graduated from The National Institute of Engineering, Mysore, affiliated to the University Of Mysore, India with a Bachelor's degree in Mechanical Engineering.

1981-84 Worked as an Industrial Engineer in the departments of Management Services and Industrial Engineering at Bharat Electronics Ltd., Bangalore, India.

1986 Curerently a candidate for M.A.Sc. degree in Industrial Engineering at the University of Windsor, Windsor, Ontario, Canada.