1990

Design of cellular manufacturing systems.

Divakar. Rajamani
University of Windsor

DESIGN OF CELLULAR MANUFACTURING SYSTEMS

by

Divakar Rajamani

A Dissertation
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 Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

1990

ISBN 0-315-61895-7

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

Divakar Rajamani

I further authorize the University of Windsor to reproduce this dissertation 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.

Divakar Rajamani

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

ABSTRACT

Cellular manufacturing represents a major technological innovation to most organizations. One of the problems in the design of cellular manufacturing systems is the cell formation, which is essentially the identification of part families and machine groups. Cells are formed using new and often automated machines and material handling systems. A judicious selection of processes and machines are necessary for cell formation. With the introduction of new parts and changed demands, new part families and machine groups have to be identified. The redesign of such systems warrant the consideration of practical issues such as relocation expense on existing machines, investment on new machines etc.. Usually, the manufacturing facility cannot be completely divided into cells. Rather, a portion of it remains a large functional job shop. Thus, there is a need to select parts and machines for cellularization. This has a great impact on the utilization of cells. The creation of exclusive cells with no inter-cell movement is a common goal for cell formation. However, many times it is not economical or practical to achieve exclusive cells. Material handling is an important factor to be considered in this situation. In fact, new technology and faster deterioration rate of certain machines could render the previously allocated parts/machines undesirable. Thus, there is also a need to determine if the old machines must be replaced with new or technologically updated machines. The objective of this research is to develop mathematical models to address these issues for cell formation encountered in cell design and suggest efficient solution methodologies to solve the models developed.

It is assumed that a part can be produced through one or more process plans. Each operation in a process plan can be performed on alternate machines. Thus, for each process plan we have a number of production plans depending on the machines selected for each operation. It is also assumed that the demand for a part could be split and can be produced in more than one cell. The plans identified to produce the same part in different cells could be different.

The cell formation problem, in addition to identifying part families and machine groups is to specify the plans selected for each part, quantity to be produced through the plans selected, machine type to perform each operation in the plans, total number of machines required, machines to be relocated, machines to be replaced and parts and machines to be selected for cellularization considering demand, time, material handling and resource constraints.

Some pertinent objectives to be considered are minimization of investment, operating cost, machine relocation cost, material handling cost and maximization of output. Consideration of physical limitations such as upp bound on cell size, machine capacity, material handling capacity etc., should also be incorporated into the cell design process.

Accordingly, a number of mathematical models are developed to provide a framework for discussing the issues related to design of cellular manufacturing systems. All the models developed are large scale linear and mixed integer programs. For the solution of the linear and relaxed

mixed integer programming models, an efficient column generation scheme is presented. In the problems under consideration, column generation is achieved by solving simple assignment problems. A branch and bound scheme on the integer variables leads to an optimal solution for the mixed integer programs. Each node in the branch and bound tree represents a solution to an augmented continuous problem with additional constraints on the integer variables. These additional constraints are easily incorporated without increasing the size of the prohlem by the bounded variables procedure. A number of illustrative examples are solved tc illustrate the application of the solution methodology. Computational experience is provided for a few test problems and sta'istics on number of nodes, number of plans generated, number of pivot operations, number of assignment problems solved and time for execution are included.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I sincerely wish to thank Dr. N. Singh and Dr. Y.P. Aneja for their valuable guidance and support during the course of this research. I also, thank the committee members Dr. R.S. Lashkari, Dr. M. Zuo and Dr. W.P. North and Dr. G.R. Monforton, Chairperson, for their suggestions. My gratitude also goes to the external examiner Dr. J.A Buzacott for his comments and suggestions.

I would like to express my appreciation to Ms. Jacquie Mummery, Mr. Tom Williams, Ms. Alison Samson and the computer consultants for their assistance. I also, acknowledge the financial support provided by University of Windsor during my research.

Finally, I thank all my friends and roommates for their encouragement and being accommodative.

TABLE OF CONTENTS

LIST OF TABLES

NOMENCLATURE

NOTATION:

INDEXING SETS:

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

$m = 1,2......M$ machine

$p = 1,2......P_k$ process plans for part k

$s = 1,2......S(kp)$ operations for (kp) combination

$c = 1,2......C$ cell

$g = 1,2......G$ group

$g' = 1,2......G$ group

$l = 1,2......L(kp)$ for (kp) combination

$= 1,2......L(kpg)$ for (kpg) combination

$= 1,2......L(kpc)$ for (kpc) combination

DECISION VARIABLES:

d_{mg} = Under utilization of machine type m in group g

$$I_k = \begin{cases} 1 & \text{if part k is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$r_{fc} = \begin{cases} 1 & \text{if part family f is assigned to cell c} \\ 0 & \text{otherwise} \end{cases}$$

$$r_{kc} = \begin{cases} 1 & \text{if part k is assigned to cell c} \\ 0 & \text{otherwise} \end{cases}$$

R_{mg} = Number of machines of type m removed from group g

$$X_{ms}(kp) = \begin{cases} 1 \text{ if machine m is assigned to operation s} \\ \quad \text{for (kp) combination.} \\ 0 \text{ otherwise} \end{cases}$$

$X_{(lkp)}$ = Number of units of part k produced using process plan p and production plan l

$X_{(lkpg)}$ = Number of units of part k produced using process plan p and production plan l in machine group g

$X_{(lkpc)}$ = Number of units of part k produced using process plan p and production plan l in cell c

$X_{(lkpcf)}$ = Number of units of part k in family f produced using process plan p and production plan l in cell c

$$Y_{kp} = \begin{cases} 1 & \text{if part k is manufactured using process plan p} \\ 0 & \text{otherwise} \end{cases}$$

Z_m = Number of machines of type m

Z_{mc} = Number of machines of type m in cell c

Z_{mf} = Number of machines of type m required to process parts in family f

Z_{mg} = Number of machines of type m in group g

Z_{mig} = Number of machines of type m moved from group i to group g

COEFFICIENTS:

$$\alpha_{ms} = \begin{cases} 1 & \text{if machine m can perform operation s} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{kf} = \begin{cases} 1 & \text{if part k is a member of part family f} \\ 0 & \text{otherwise} \end{cases}$$

$$a_s(kp) = \begin{cases} 1 & \text{if operation s has to be performed for the (kp)} \\ & \text{combination} \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ms}(lkp) = \begin{cases} 1 \text{ if in plan } l \text{ machine m is assigned to operation s} \\ \quad \text{for (kp) combination.} \\ 0 \text{ otherwise} \end{cases}$$

$$a_{ms}(lkpg) = \begin{cases} 1 \text{ if in plan } l \text{ machine m is assigned to operation s} \\ \quad \text{for (kpg) combination.} \\ 0 \text{ otherwise} \end{cases}$$

$$a_{ms}(lkpc) = \begin{cases} 1 \text{ if in plan } l \text{ machine m is assigned to operation s} \\ \quad \text{for (kpc) combination.} \\ 0 \text{ otherwise} \end{cases}$$

$$a_{ms}(lkpcf) = \begin{cases} 1 \text{ if in plan } l \text{ machine m is assigned to operation s} \\ \quad \text{for (kpcf) combination.} \\ 0 \text{ otherwise} \end{cases}$$

$$a_{mg's}(lkpg) = \begin{cases} 1 \text{ if in plan } l \text{ machine m in group g' is assigned} \\ \quad \text{to operation s for (kpg) combination.} \\ 0 \text{ otherwise} \end{cases}$$

b_m = Time available on each machine of type m

b_{mg} = Total available time on all machines of type m in group g

B = Operating budget

C_m = Cost of purchasing one machine of type m

C_{mg} = Cost of assigning one machine of type m to group g

C_{mig} = Cost of relocating one machine of type m from group i to group g

$$c_{ms}(kp) = \begin{cases} \text{cost for machine m to perform operation s for (kp)} \\ \text{combination} \\ \text{infinity if machine m cannot perform this operation} \end{cases}$$

$cc_{gg'}(k)$ = Cost of moving part k from group g to g'

$d_{gg'}$ = Distance between group g and g'

d_k = Demand for part k

D = Capacity of material handling equipment

D_m = Depreciation value of each machine of type m during the time period

$L(kp)$ = Number of different production plans for all (kp)

$L(kpc)$ = Number of different production plans for all (kpc)

$L(kpg)$ = Number of different production plans for all (kpg)

Max_c = Maximum number of machines allowed in cell c

Max_g = Maximum number of machines allowed in group g

N_m = Number of machines of type m

N_{mg} = Number of machines of type m in group g

r = Cost of capital

$$t_{ms}(kp) = \begin{cases} \text{time for machine m to perform operation s for (kp)} \\ \text{combination} \\ \text{infinity if machine m cannot perform this operation} \end{cases}$$

W_m = Book value of each machine type m at beginning of time period

CHAPTER 1

INTRODUCTION

Manufacturing systems for discrete-parts production can be classified as job shop, batch and mass production. Job shops involve low-volume production of a large variety of parts. A typical job shop has several departments, where each department provides processing facilities for specific operations (eg. drilling, milling etc.). The parts have to move from one section to another for various operations. With such a layout, the part spends a substantial amount of time (about 95 %) waiting in front of machines and on set up. The large amount of waiting and set up time increases the manufacturing lead time resulting in low productivity. Mass production (Transfer lines) applies when high-volume production of a very few products exists. Changing a transfer line to the production of a different item is costly in terms of lost production. Batch production describes medium-volume production of a moderate number of products. They are characterized for high lead times and low productivity.

The concept of cellular design emerged to reduce set ups and batch sizes. The remedy lay in sorting out parts into groups that have common characteristics - shape, tolerance, process of manufacture etc.. Small groups of machines were formed and then groups of parts were allocated to machine groups. All the operations needed for a part-group were to be performed within the machine group. The need for flexibility and high productivity led to the development of Flexible Manufacturing

System (FMS). FMS is an automated manufacturing system designed for small batch and high variety of parts which aims at achieving the efficiency of automated mass production, while utilizing the flexibility of a job shop. The development and implementation of FMS has received wide spread popularity resulting in a prolific bibliography of literature. A comprehensive review of the several frameworks for analysis, planning and control of these systems is given in O'Grady and Menon (1986), and Buzacott and Yao (1986). The benefits of cellular manufacturing stress that FMS justifications must proceed on the basis of explicit recognition of the nature of the cellular manufacturing. In this thesis the focus is on the design of cellular manufacturing systems for a variety of situations.

1.1 Cellular Manufacturing Systems (CMS)

Cellular manufacturing involves processing of similar parts (part families) on dedicated clusters of dissimilar machines or manufacturing processes (machine groups). Thus, a manufacturing cell is a collection of dissimilar machines or manufacturing processes dedicated to a collection of similar parts. Cellular manufacturing is common in metal machining, fabricating and manual assembly operations. In the metal machining industry a cell may consist of three to fifteen machines with an average of six machines. Very rarely will a machine group have identical machines. Parts in a family will require all the machines in the cell, but it does not mean each one of the parts in the family will require processing on all the machines in the cell. Some parts may require machining only on a few machines.

Most manufacturing systems are hybrid systems. That is, they are not either job shops or flow shops or cell shops, but rather a combination of interrelated subsystems. Job shops have jumbled material flows and flow shops have more or less unidirectional, dominant flows. Cell shops, however, do not bear these connotations. The internal flows of a cell can be either straight or jumbled. In essence, a cell can represent job shop or flow shop subsystems. It follows from the definition above that both a transfer line for machining parts and an assembly line could be classified as cells.

The most claimed advantages of CMS are reduced lead time, reduced in-process inventory, smaller lot sizes, reduction in the number of production equipment, reduced labor cost, reduced material handling, reduced tooling requirements, simplified production control procedures, improved productivity and better overall control of the operations.

As any other system CMS has its own disadvantages. A few of these are increased capital investment, lower machine utilization and impossibility of having 100% cellular Manufacturing. By taking a judicious decision one can reap the benefits of CMS which far outweighs the disadvantages.

1.2 Design of Cellular Manufacturing Systems

Designing cellular systems is a complex undertaking with broad implications for the organization. The decisions to be taken in a design process are numerous. A first distinction can be made between decisions related to system structure and operational procedures

(Wemmerlov and Hyer 1986). To the first group belong choice of part types to be processed in the cell, the type and number of machines on which these parts are processed, the routings, the type and number of material handling equipment, system layout, the type and number of operators, and the type and number of tools and fixtures for the cell. Issues related to procedures include detailed job design, organization of supervisory and support personnel around the cellular structure, formulation of inspection and maintenance policies, modifications to cost control and reward systems, design of production planning and control, and related hardware and software acquisitions.

It is not possible to delineate a strict sequence of decisions to be made in connection with cell design. One can say, however, that structure-oriented decisions tend to precede procedure-oriented ones. At this stage it should also be clear that the system structure and procedures can be changed due to experiences derived during the operation of the cell system over time. For example, the part population, the machine routings, or even the machine population are subject to change during the implementation phase or later, due to changing internal or external conditions. Within the group of structural decisions, the identification of part families and machine groups take on a particular significance since most subsequent decisions depend on these choices.

Evaluation of design decisions can be categorized as relating to either the system structure or system operation. Typical considerations related to system structure are listed below:

- Equipment relocation cost (low)
- Equipment and tooling investment (low)
- Existence of inter and intra cell movements (low)
- Floor space requirements (low)
- Extent to which parts are completed in a cell (high)
- Flexibility (process parts on multiple machines and/or in multiple cells - high)

Evaluations of system design decisions are incomplete and not totally meaningful unless they also relate to the operation of the system. A few typical performance variables related to system operation are:

- Equipment utilization (high)
- Work-in-process inventory (low)
- Queue lengths at each work station (short)
- Job output rate (high)
- Job throughput time (short)
- Job lateness (low)
- Time in queue at each station (short)

A major problem throughout the cell design process is the necessity of trading-off objectives related to structural parameters and performance variables against each other. For example, higher machine utilization can be reached if several cells route their parts to the same machine. The drawbacks are increased queuing and control problems. The list of possible trade-offs can be made quite long. It is the nature of the design process to be open-ended, and to have few initial restrictions on the solutions. This is what makes the cell design a complex problem.

1.2.1 Cell Formation Problem

One of the problems in the design of CMS is the cell formation, which is essentially the identification of part families and machine groups. It is a GT philosophy where the manufacturing system, in total or in part, has been converted into cells. The simplest application common in batch manufacturing environments, is to informally rely on part similarities to gain set-up efficiencies when sequencing jobs at a work center. The second application is to create formal part families, dedicate equipment to these families, but let the equipment remain in its original position. The ultimate GT application in manufacturing is to form manufacturing cells. A sequential or simultaneous approach could be adopted for cell formation. The sequential approach first forms the part families or machine groups followed by machine assignment or part allocation respectively. The simultaneous approach determines the part families and machine groups simultaneously. Though, the simultaneous approach is better it usually suffers from computational difficulties. Considering the fact that CM represents a major technological innovation to most organizations, a number of problems are encountered which are situation specific. A few of these problems are discussed in the following sections.

1.2.1.1 New Machines

Many cells today are formed using new and often automated machines and material handling equipment. Flexible manufacturing systems (FMS) are examples of such automated cells where production control activities are also under computer supervision (Buzacott and Yao 1986). Considering the versatility of these machines and high capital investment a judicious selection of the processes and machines is necessary for cell formation. For reasons such as design standardization and computerized process planning, some firms use geometric-features-based grouping to design part families. In a recent survey of 53 respondents in US, 62 % indicated the use of one or more classification and coding schemes in conjunction with GT applications. The part families were determined without the reliance on production methods (Wemmerlov and Hyer 1987b). The problems here are :

1. How to identify the machine groups such that all the parts in a family are processed within the group ?

2. If the number of part families identified is large, then, how to assign one or more part families to the machine groups to form the desired number of cells ?

3. However, if the part families are not established ,then, how to identify the part families and machine groups simultaneously ?

1.2.1.2 Machine Relocation

Cell development is evolutionary and, therefore, always in a state of flux. Steady and predictable demands are desirable features for parts produced in cells. However, over time, as products are phased in and out and parts are redesigned, the composition of the part families in the cells will change. This may lead to capacity imbalance problems where too much or too little work is loaded onto the cells. A survey conducted by Wemmerlov and Hyer (1987b) indicates the percentage of new parts averaged 11.3 % with a range from 5 % to 25 %. With the introduction of new parts and changed demand the new part families and machine groups should be identified. The cost of physically moving machines to new assignments for frequent reassignments is uneconomical. At the same time once the cells are formed it should not be considered an irreversible decision. The planning horizon is not long enough to allow us to buy new machine tools, rather the number of types of machines available is fixed and is normally determined at inception. The cost of moving equipment was also one of the most common reasons given by the firms for not building cells and practicing machine dedication instead (Wemmerlov and Hyer 1987b). Thus, a major problem the industry faces today is how to assign the parts to the known machine groups and which machines are to be relocated such that exclusive part families and machine groups are identified at minimum relocation cost ? If, however, the capacity available is not sufficient should the relocation be accompanied or substituted by a higher degree of investment in new equipment ?

1.2.1.3 Selecting Parts and Machines

Essentially, the objective of cellular manufacturing is to create specialized cells that can process a limited number of different job types. Usually, the manufacturing facility cannot be completely divided into specialized cells. Rather, a portion of the facility remains as a large functional job shop. The odd jobs which cannot be completely machined in a specialized cell are usually assigned to the functional job shop.

The extent to which a company accounts for the manufacturing activities in specialized cells can be measured by the fraction of all annual hours in the plant that are expended in cells. This is referred to as the extent of cellularization or degree of cellularization of the company (Wemmerlov and Hyer 1987b). The majority of the surveyed companies(of 27 companies) had a limited number of cells, and 75 % reported that 25% or less of the annual machine hours were expended in cells. Thus, firms with cells might have to operate two systems simultaneously until it is fully cellularized.

The parts selected to be manufactured in the cells have a great impact on the utilization of cells. Also, which cell among the feasible cells the part should be assigned to is critical because it prescribes the overall balance of the CM system. A load imbalance occurs when some cell machines or processes are more utilized than others. Some of the ad hoc procedures employed in selecting parts to be manufactured in cells are based on selecting parts which require a particular machine, visual examination of part drawings, identifying parts which belong to a particular product line, select parts with same name or function, by examining the codes etc. (Wemmerlov and Hyer 1987b). Thus a few problems that exist in a company that is partly cellularized or is in an inception stage of converting a job shop or adding new machines to have a pilot cells are:

1. How to select the parts to be produced in the machine groups already identified ?

2. If the company decides to start with a few pilot cells and the machine types available for the purpose of cellularization is known then how to simultaneously form the required number of machine groups and select the subset of parts to be produced in these cells ?

3. If, however, only the number of cells to be identified is known, then

how to identify the subset of machines and parts in the cells ?

1.2.1.4 Material Handling and Replacement

The creation of independent cells, i.e., cells where parts are completely processed in the cell and no linkages with other cells in the factory exist, is a common goal for cell formation. However, many times it is not economical or practical to achieve cell independence, especially, when under-utilization, load imbalance and higher capital investment are the potential threats of introducing cellular manufacturing.

Therefore, there is a need to include the additional important cost, namely the material handling cost in cell formation. If possible, the ideal situation would be to eliminate all handling. However, since this cannot be done, the next best alternative is to minimize the negative effect of handling on the facility.

In a typical manufacturing system, the assignment of parts to different cells could take place periodically depending on the change in the volume of production and part mix. This is generally motivated by work load changes and/or machine utilization improvements. In fact, new technology and faster deterioration rate of certain machines in cells could render the previously allocated parts to a cell undesirable. Thus there is a need to replace these machines and substitute with new machines. These new machines could be technologically updated or the same. If the machines are different there is a need to identify the part families again.

1.3 Organization Of the Dissertation

The dissertation is organized and presented in nine chapters as follows. In Chapter 1, an introduction to cellular manufacturing systems and cell formation problem is given. A complete review of existing literature on cell formation problem is given in Chapter 2. Motivation for the proposed research and objectives of the research are also included in Chapter 2. In Chapter 3, mathematical models are developed to illustrate the influence of alternate process plans on cell design. The mathematical models for cell formation considering the purchase of new machines is presented in Chapter 4. The mathematical models for cell formation considering machine relocation are presented in Chapter 5. In Chapter 6, mathematical models for selection of parts and machines for cellularization are presented. Material handling and replacement of machines are considered in the models presented in Chapter 7. Computational experience on test problems is reported in Chapter 8. The contributions of the research, limitations of the work and directions for future research are given in Chapter 9. The solution methodology developed is presented in the respective chapters and illustrated with examples.

CHAPTER 2

LITERATURE REVIEW

Modeling of cellular manufacturing systems has received considerable attention from researchers. Numerous heuristics and analytical methods have been reported in literature for the design of cells. A comprehensive review of the available literature is discussed in Wemmerlov and Hyer (1986). Chu and Pan (1988) provided a state-of-the-art review on the use of clustering techniques in cell formation. In the following section a detailed chronological review of literature including the type of modeling and solution approaches is given.

2.1 Chronological Review

The available literature is classified into one of the following:

1. Coding and classification based methods
2. Evaluative methods
3. Similarity coefficient methods
4. Heuristic methods
5. Matrix based methods
6. Network based methods
7. Mathematical models

The literature is reviewed in a chronological manner in each category. In the evaluative methods, the analysis of the routing information is done manually. When this analysis is done analytically it is referred to as similarity coefficient methods. In the matrix based methods, machine-part groups are formed by manipulation of the matrix using some algorithms. The network based methods represent the machine-part matrix in the form of a bipartite graph and use network decomposition methods or other heuristic procedures.

1. Coding and classification based methods

Opitz (1970) developed a coding scheme for parts based on geometric and technological attributes. Part families are formed manually or using analytical methods so that parts with similar attributes are grouped into a family.

Kusiak (1983) extends the classification and coding system for FMS by taking into account the requirements imposed by tools and fixtures for the manufacturing of a given part. He has proposed the application of a hierarchical clustering algorithm to form part families. A distance matrix is calculated from the parts code using suitable distance metrics. The problem is also modeled as a p-median problem with an objective to minimize the total sum of distances in a part family to the part family median and solved using the effective sub gradient algorithm (Kusiak 1985).

Gongaware and Ham (1981) and Han and Ham (1986) reported multi-objective clustering techniques to form part families based on similarity vectors as well as part codes.

Hyer and Wemmerlov (1985) have discussed the structures, applications and implementation of the GT oriented coding systems. Some of the more common available coding systems available in the U.S are also described.

Dutta et al. (1985) presented a methodology using a coding system derived from Opitz code to form design and tooling families.

2. Evaluative methods

The concept of production flow analysis (PFA) was first introduced by Burbidge (1971). The main feature of PFA is that it involves the systematic listing of the components in various ways, in the expectation that groups of machines and components may be found by careful inspection.

El-Essawy and Torrance (1972) proposed Component flow analysis (CFA) which is almost similar to PFA.

Burbidge (1973) also reported a method called Nuclear synthesis which is based on selecting machines used by a few components as starting points for various cells, or nuclei. The next machine is allocated on the basis that it has the smallest number of components left unassigned to a group. Once the Nuclear synthesis is completed, these nuclei are modified.

Purcheck (1974) has adopted a set-theoretic approach. He suggested the use of a lattice diagram where the edges represent the various combinations of possible manufacturing processes routes that can be generated with the existing production facilities. The lattice diagram grows exponentially as the set is enlarged and hence its usefulness is limited to being an illustrative device. A linear programming technique is used for combining the manufacturing facilities demanded by the process route to form host and guest cells.

De Beer et al. (1976) suggested a modified form of PFA. The method involved first determining what combination of operations should be made to specify a specific 'level of autonomy' followed by determining the requisite capacity per operation in each group. They identify three categories of machines namely primary, secondary and tertiary depending on if a machine type is present in only one cell, more than one cell and all cells respectively. Once the cells are formed, every routing is assigned to a group of routings depending upon whether they fit into all cells, some cells, one cell, or no cell. Every group is then assigned to one or more cells and the maximum and minimum of work load per machine type is computed in every cell.

Burbidge (1977) described how PFA could be carried out manually. De Beer and De Witte (1978) have attempted to extend the PFA approach to consider explicitly the question of machine duplication. This method has been termed as production flow synthesis.

3.Similarity coefficient methods

McAuley (1972) introduced the the zero-one matrix, constructed with rows representing machines and columns representing components. An element of the matrix is 1 if a component visits the corresponding machine and 0 otherwise. He used the Jaccard similarity coefficient (defined as the number of components which visit both machines, divided by the number of components which visit at least one of the machines) followed by a single linkage cluster analysis to form machine groups. The procedure gives a set of solutions at different similarity levels, from which the best could be chosen on the basis of other criteria. This method might however link two clusters whose members may be quite far away from each other in terms of similarity.

Carrie (1973) applied numerical taxonomy for finding part families and machine groups and demonstrated that this method could get the same result that Burbidge (1973) got by hand. In these methods (Carrie 1973, McAuley 1972) one has to specify the threshold level of similarity or, decide in advance the number of groups.

Rajagopalan and Batra (1975) developed a graph theoretic method which uses cliques of the machine-graph as a means for grouping machines. The vertices of this graph are machines and the arcs are jaccard similarity coefficients. The main disadvantage of this approach is that due to the high density of the graph, large cliques are not vertex disjoint.

De Witte (1980) has suggested three similarity coefficients based on routing and machine times to indicate the interdependence of machine types. The method consists of four steps, namely, gathering information, analyzing relations between machines and allocation of machine types to cells, allocation of components to cells and finally counting the work load for each machine type in each cell, and allocation of the requisite number of machines to cells.

Waghodekar and Sahu (1984) have presented a heuristic approach, called MACE, based on the similarity coefficient of the product type. The method yielded a minimum number of exceptional elements when tested on a number of problems and compared with similarity coefficient of additive type. It is computationally straightforward.

Faber and Carter (1986) adopted a different similarity measure based on the number of parts processed by both the machines. They introduced a graph theoretic algorithm for grouping the machines. The polynomially bounded algorithm uses the network approach to optimally find the densest sub graph of a graph.

Steudel and Ballakur (1987) suggested a two stage dynamic programming heuristic which first determines the optimum chain so that the sum of the bonds among the machines is maximized followed by partitioning the chain to form machine groups. A similarity measure, called Cell Bond Strength based upon part routing and production requirement data was introduced.

4.Heuristic methods

Purcheck (1975 a) has proposed a classification scheme which combines machine requirements and machine sequences by coding them in the form of strings of letters and digits. Various mathematical programming formulations has also been suggested by Purcheck (1975 b).

Lemoine and Mutel (1983) presented a dynamic cluster algorithm for part families and machine cell recognition. The method takes into account the capacity and the load of the machines and some other user's constraints. Another characteristic is that they consider the machine tool set as the sum of p similar machines type subsets so that they automatically balance the the machine types on the cells.

Rodriguez and Adaniya (1985) suggested an interactive procedure to obtain the number of cells and machines allocated to each cell such that there is a balance between the average set-up costs and inventory holding costs. The procedure uses ROC algorithm to generate block diagonal matrix. Then it schedules the products using the concept of economic lot sizing and checks for feasibility, machine utilization etc..

Purcheck (1985) tackles the problem of machine-component grouping in a different fashion. Group formation is done in terms of minimum differences between masters and maximum combination of masters. A master is defined as a unique or most complex part that has to be processed in one cell. The heuristic proceeds by computing the master sets and their differences. Then the corresponding work load is computed and the combination of master sets is revised, if required, based on certain acceptability tests.

Askin and Subramanian (1987) proposed a heuristic procedure for economic determination of machine groups and corresponding component families considering costs of work-in-process and cycle inventory, intra group material handling, set-up, variable processing and fixed machine costs. The three stage procedure initially reorders part types based on routing similarity and then attempts to combine adjacent part types to reduce machine requirements. Finally groups are combined where economic benefits of utilization offset those of set-up, work-in-process and material handling.

Ballakur and Steudel (1987) have provided an efficient heuristic considering within-cell utilization, work load restrictions, and cell size restrictions. A two stage clustering procedure is suggested. In the first stage, an 'admit or reject' decision is made for each work centre based on the work load fractions. The actual number and assignment of machines to a work centre are made at the second stage.

Meenakshisundaram and Fu (1987) proposed a technique based on the hospitality and flexibility relationships suggested by Purcheck (1975). Host cells are formed using the route sheet of parts and a reduction in number of cells achieved using the integer programming technique.

Mosier (1989) has reported the development of a number of similarity-based coefficients designed for applying hierarchical cluster analysis to the cell formation problem. He also discusses an experimental investigation applying these and other well known similarity coefficients in conjunction with some well known clustering algorithms.

Harhalakis et al. (1990) have suggested a simple two step heuristic algorithm. The first step of the heuristic is a bottom up aggregation procedure to minimize what is defined by them as 'Normalized Inter-Cell

Traffic'. The second step is a procedure to attempt further improvement, in which the significance of a machine to a cell is validated. A large scale industrial application has also been provided.

5.Matrix based methods

Bhat and Haupt (1976) developed an efficient clustering algorithm known as matching algorithm on the concept of matching between two rows (columns). This method is computationally efficient, as the change in the number of matchings need not be recomputed for each possible arrangement, whenever a row is rearranged.

King (1980 a,b) suggested an algorithm, known as rank order clustering algorithm (ROC) for simultaneous grouping of machines and parts by considering the part-machine incidence matrix as an array of binary numbers and arranging them in increasing (or decreasing) order. By repeating the process for rows and columns he obtained disjoint groups. King and Nakornchai (1982) later extended the algorithm to enable the identification of bottleneck machines in the system. They also suggested ROC2 which takes into account the weaknesses of ROC and is computationally more efficient. Here several rows and columns are sorted at the same time instead of element by element as done by ROC.

Chan and Milner (1982) suggested the direct clustering algorithm which progressively restructures the matrix until there is no more improvement due to restructuring.

Kusiak (1985) suggested the rank energy algorithm which orders the rows and columns of the matrix and its transpose based on their weights. This algorithm generates solutions of acceptable level and is superior to the existing algorithms of this type.

Chandrasekaran and Rajagopalan (1986 b) proposed MODROC, an extension to the ROC algorithm. The deficiencies of ROC are removed to a great extent. This method uses ROC in conjunction with a block and slice method and a hierarchical clustering method.

Seifoddini and Wolfe (1986) improved on the existing methods based on similarity coefficient method by dealing with the duplication of bottleneck machines and employing a special data storage and analysis technique to simplify the machine-part grouping process. The duplication is based on the number of inter-cellular moves and starts with the machine generating the largest number of inter-cellular moves and continues until no machine generates more inter-cellular moves than specified by the threshold value. Alternative solutions can be examined by changing the threshold value. The method, however, does not consider machine duplication cost and production requirements of parts.

Khator and Irani (1987) introduced the heuristic procedure called the Occupancy Value Method. The occupancy values were defined for each part based upon the number of parts and machines and routing. The procedure progressively develops block diagonalization starting from the northwest corner of the matrix to form the groups.

Kusiak and Chow (1987) developed two efficient algorithms to solve the machine-part grouping problem. A cluster identification algorithm suggested to form the groups from matrix is reported to be the most efficient algorithm developed to date. A cost analysis algorithm is developed to solve the augmented formulation of the problem, which associates cost with part and limits the number of machines in each cell.

Kusiak (1987) proposed a generalized GT concept, based on the generation of a number of different process plans for each part.The presence of alternate process plans improves the diagonal structure of the final matrix and reduces the number of bottleneck machines. The problem is also modeled as a 0-1 integer program.

6.Network based methods

Kusiak et al. (1985) and Kumar et al. (1986) modeled the parts and components grouping as an optimal k-decomposition of weighted networks problem. The decomposition problem is approximated as a quadratic assignment problem and is solved in two phases.

Vannelli and Kumar (1986) showed that the problem of identifying the minimal number of bottleneck machines is equivalent to finding the minimal cut nodes of a graph. A heuristic based on dynamic programming approach is developed to solve the problem.

Chandrasekaran and Rajagopalan (1986 a) formulated the grouping problem as a bipartite graph and derived an expression for the upper limit on the number of groups. A non-hierarchical clustering method is adopted for grouping components into families and machines into cells. After diagonally correlating the groups, an ideal-seed method (Ideal-seeds are centroids of the imaginary perfect groups with essentially the same block diagonal structure as the groups formed initially) which reorders rows and columns is used to improve the groups. A quantitative criterion called grouping efficiency which is the weighted average of utilization and inter cell movement is developed to compare alternative solutions.

7. Mathematical models (and heuristic approach)

Choobineh (1988) proposed a two stage approach for cell design. Clustering techniques, with a proximity measure using the manufacturing operations and operations sequences, are used for forming the part families which constitutes the first stage. An integer programming model is used to specify the type and number of machines in each cell and the assignment of part families to the cells. It is assumed an operation can be performed on more than one machine type, which is a very realistic assumption.

Co and Araar (1988) presented a three stage procedure for configuring machines into manufacturing cells, and assigning the cells to process specific sets of parts. First, operations are assigned, with the objective of minimizing the deviation between available capacity and the work load assigned to each machine. This results in a machine-part matrix, which is manipulated using an extension of King's algorithm to form part-machine groups. Then a direct search algorithm is used to determine the number of cells, and the composition of each cell.

Askin and Chiu (1988) have proposed a mathematical model considering costs of inventory, machine depreciation, machine setup and material handling. They decompose the model into two subproblems and suggest a heuristic approach for solution.

Kasilingam (1989) in his dissertation has developed a number of mathematical models for machine allocation problem, machine-part grouping problem and new parts allocation problem. He has devised methodologies to solve large instances of the models developed. New indices to express similarity between two machines and between a part and machine are developed considering the processing requirements and capabilities of machines. He has also extended his machine-part grouping formulation to account for the presence of alternate process plans for parts.

Shtub (1989) has shown that the simple cell formation problem, in which one process plan is considered for a part and the general case , in which several process plans are considered for each part type, is equivalent to the Generalized Assignment Problem. This can be considered as another approach to the problem.

Seifoddini (1989) considers the economic trade off between machine duplication in cells and inter-cell movement in cell formation. This procedure is again based on decomposing part/machine matrix but considers the production volumes and processing times.

Srinivasan et al. (1990) present an assignment model to solve the grouping problem. A similarity coefficient matrix is used as the input to the assignment model. Closed loops in the form of sub tours are identified after solving the problem and are used as the basis for grouping. They have also shown that this method is superior to the p-median model both in terms of quality of solution and computational time.

2.2 Motivation for Proposed Research

A comprehensive review of various approaches to design of cells was discussed in the previous section. Much of this research has been directed towards developing cluster algorithms which decompose a part/machine incidence matrix into nonoverlapping, diagonally adjacent blocks or clusters of cells. The common clustering criterion used is a similarity or dissimilarity index. Among the measures of performance are the number of exceptional elements and bottleneck machines. Real life factors such as costs, processing times and production volumes which influence the cell formation are often ignored. Also each operation of a part is assumed to be performed on one machine. Further, the clustering algorithms do not guarantee the formation of mutually exclusive cells. Attempts have also been made to formulate mathematical programming models, but the presence of alternate process plans have not been considered explicitly.

The process plan for a part is not unique. Two or more process plans can be generated for each part. Each process plan will identify a sequence of operations to manufacture a part. Also operations for a part can be performed on more than one machine. So far only selection of one plan for a part has been considered for a part. In practice using two or more plans for a part and producing it in more than one cell may lead to better operational/routing flexibility and better resource utilization. Therefore, there is a need to integrate the above factors and develop efficient procedures that can generate optimal solutions.

Although numerous heuristics and analytical methods have been reported in literature for cell formation, the redesign of cells giving considerations to machine relocation and machine replacement has not been addressed.

Also, not much work has been reported in the area of selection of parts and machines for cellularization.

All this indicates that there is a need for development for procedures that can generate optimal solutions with consideration to stated design goals. Then, instead of relying more or less on trial and error, such procedures could, in the design stage, overcome operational difficulties with cellular manufacturing systems.

3.3 Objectives of Proposed Research

The objectives of the proposed research are as follows:

1. Identify a number of issues related to cell design with:

    - New machines
    - Machine relocation

    - Selection of parts and machines
    - Material handling and machine replacement

2. Develop mathematical formulations for cell design to address the issues identified.
3. Suggest efficient solution methodologies to solve the models presented and illustrate the application with examples.
4. Report computational experience on randomly generated test problems.

CHAPTER 3

INFLUENCE OF ALTERNATE PROCESS PLANS ON CELL DESIGN

A generalized group technology problem of manufacturing a group of parts in which each part can have alternate process plans and each operation in these plans can be performed on alternate machines is considered. The objective is to model and analyze how alternate process plans influence the resource utilization when part families and machine groups are formed simultaneously. Accordingly, three integer programming models are developed to successively study the effect of alternate process plans and simultaneous formation of part families and machine groups.

3.1 Alternate Process Plans

In this research, it is assumed that a part can have more than one process plan and each operation in a process plan can be performed on more than one machine. For example, consider the manufacture of a gear. If the initial raw material is in the form of a bar stock, the following eight processing steps are required to transform the raw material into a finished gear:

PROCESSING STEPS (PS):

PS#1 : Facing

PS#2 : Turning

PS#3 : Parting-off

PS#4 : Facing

PS#5 : Centering

PS#6 : Drilling

PS#7 : Slotting

PS#8 : Gear teeth cutting

A different set of processing steps can be identified if the raw material is in a different form, say, blanks either cast or forged. Once the processing steps have been identified, the process planner determines the possible sequences of processing before grouping the processing steps into operations. The eight processing steps in the gear manufacture can be grouped into different sets as follows:

	PLAN #1	PLAN #2
Operation 1.	PS#1,2,3	PS#1,2,3
Operation 2.	PS#4,5,6	PS#4,5,6
Operation 3.	PS#7	PS#7,8
Operation 4.	PS#8	

It is possible to alter such grouping to suit the manufacturing system requirements. For example, in the gear manufacture the first six processing steps can be combined to perform them with one setup, say, on a turret lathe. Further, processing step PS#6 can be separated and performed on a drilling machine. Also, each operation in the plans can be performed on a number of compatible machines. For example, the gear teeth cutting operation can be performed either on a milling or a gear hobbing machine if plan # 1 is used. If plan # 2 is used where the gear teeth cutting and slotting operations have been combined, it can only be performed on a milling machine.

3.2 Mathematical Formulations

The objective is to model and analyze how alternate process plans influence the resource utilization when the part families and machine groups are formed simultaneously. Accordingly, three integer programming models are developed to successively study the effect of alternate process plans and simultaneous formation of part families and machine groups. Model-3.1 assigns machines to parts. The part/machine matrix obtained from this model can be an input to one of the currently available models for cell formation (Chandrasekaran and Rajagopalan 1986 a,b, 1987, King 1980 a,b, King and Nakornchai 1982, Waghodekar and Sahu 1984). It may be pointed out that the cells thus formed may result in exceptional elements. However, this predicament can be circumvented if the part families are known. Model-3.2 presented in this chapter assigns machines to known part families to form cells. The part families so known are generally formed based on part attributes. In a recent survey of 53 respondents in the United States (Wemmerlov and Hyer 1987b), 62 % indicated the use of one or more classification and coding scheme in conjunction with GT applications. The part families were determined without the reliance on production methods. Model-3.2 adopts a sequential approach to cell design. This may not uncover natural part families to result in efficient resource utilization. Hence Model-3.3 is developed which identifies part families and machine groups simultaneously. All the models specify the process plan for each part, machine type to perform each operation in the process plan selected and the total numoer of machines required to process all the parts by considering demand, time and resource constraints. The objective function of the models is to minimize capital investment.

Model-3.1:

Assume that there is a set of parts with given demands. Each part has alternative process plans and each operation in a process plan can be performed on alternative machines. Information regarding the processing times and processing costs of operations on different type of machines for all process plans is known. In this section an integer programming model is developed to select machines and a process plan for each part such that the total capital investment on machines is minimized considering demand, time and cost constraints. The budget on operating expense is also known. The model is as follows:

(IPM-1)

$$\text{Minimize } f_1 = \sum_m C_m Z_m \qquad (3.1)$$

subject to:

$$\sum_p Y_{kp} = 1 \qquad \forall \quad k \qquad (3.2)$$

$$\sum_m \alpha_{ms} X_{ms}(kp) = a_s(kp)\, Y_{kp} \qquad \forall \quad s,k,p \qquad (3.3)$$

$$\sum_{kps} d_k X_{ms}(kp)\, t_{ms}(kp) \le b_m Z_m \qquad \forall \quad m \qquad (3.4)$$

$$\sum_{kpms} d_k X_{ms}(kp)\, c_{ms}(kp) \le B \qquad \forall \qquad (3.5)$$

$$X_{ms}(kp) - (0,1) \qquad \forall \quad m,s,k,p$$

$$Y_{kp} - (0,1) \qquad \forall \quad k,p \qquad (3.6)$$

$Z_m \geq 0$ and integer $\forall \quad m$

The objective function (3.1) minimizes the total investment. The constraints of the optimization model are given by (3.2)-(3.6). Constraints (3.2) guarantee that only one process plan is selected for a given part. Constraints (3.3) ensure that all the operations in the selected process plan are performed on one of the available machines. Constraints (3.4) ensure that the capacity of each machine type is not violated. The demand for parts is also accounted for in this constraint. The manufacturing operations identified in a process plan can be performed on more than one machine type. The operation time and cost on these machining centers vary. Constraint (3.5) restricts the operating cost of producing all parts by the process plan and machine selected to the budget. Constraints (3.6) indicate the 0,1 and integer variables. The number of 0-1 variables, integer variables and constraints are ($\sum_k P_k + \sum_{kpms} a_s^{(kp)} \alpha_{ms}$), M , and ($K + \sum_{kp} S(k,p) + M + 1$) respectively.

Model-3.2:

Model-3.2 is an extension of Model-3.1. This model assumes that part families are known and results in cell design by selecting a process plan for each part, machine type for each operation and number of machines of each type in different cells. Constraints (3.4) are modified to ensure that the capacity of each machine type in each cell is not violated. Accordingly, the model is as follows:

(IPM-2)

$$\text{Minimize } f_2 = \sum_{mf} C_m Z_{mf} \qquad (3.7)$$

subject to:

$$\sum_{p} Y_{kp} = 1 \qquad \forall \quad k \qquad (3.8)$$

$$\sum_{m} \alpha_{ms} X_{ms}(kp) = a_{s}(kp)\, Y_{kp} \qquad \forall \quad s,k,p \qquad (3.9)$$

$$\sum_{kps} (\beta_{kf}\; d_{k})\, X_{ms}(kp)\; t_{ms}(kp) \le b_{m} Z_{mf} \qquad \forall \quad m,f \qquad (3.10)$$

$$\sum_{kpms} d_{k} X_{ms}(kp)\; c_{ms}(kp) \le B \qquad \forall \qquad (3.11)$$

$$\begin{array}{lll} X_{ms}(kp) \quad - (0,1) & \forall & m,s,k,p \\ Y_{kp} \quad - (0,1) & \forall & k,p \qquad (3.12) \\ Z_{mf} \ge 0 \text{ and integer} & \forall & m,f \end{array}$$

The objective function (3.7) minimizes the total investment on machines of different types assigned to all the part families. The constraints are given by (3.8)-(3.12).

Constraints (3.2), (3.3), (3.5), and (3.6) of Model-3.1 correspond to Constraints (3.8), (3.9), (3.11), and (3.12) of Model-3.2 respectively. Constraints (3.10) ensure that the capacity of each machine type in each cell is not violated. The number of 0-1 variables, integer variables and constraints for this model are:

$(\sum_{k} P_{k} + \sum_{kpms} a_{s}(kp)\; \alpha_{ms})$, MF, and $(K + \sum_{kp} S(k,p) + MF + 1)$ respectively.

Model-3.3:

The objective of Model-3.3 is to identify part families and machine groups simultaneously. Therefore, the indicator β_{kf} which identifies part k belonging to family f in Model-3.2 is made a decision variable and is redefined as r_{kc}.

where,

$$r_{kc} = \begin{cases} 1 & \text{if part k belongs to cell c} \\ 0 & \text{otherwise} \end{cases}$$

Additional information on the number of cells to be formed and maximum number of machines in a cell is needed for developing this model. Accordingly the model can be stated as follows:

$$\text{(IPM-3) Minimize } f_3 = \sum_{mc} C_m Z_{mc} \qquad (3.13)$$

corresponding to equation (3.7)

subject to the constraints

(3.8), (3.9), (3.10), (3.11), (3.12) and

$$\sum_c r_{kc} = 1 \qquad \forall \quad k \qquad (3.14)$$

$$\sum_m Z_{mc} \leq MAX_c \qquad \forall \quad c \qquad (3.15)$$

where MAX_c denotes the maximum number of machines in each cell. Also, the indicator β_{kf} in constraints (3.10) is replaced by r_{kc}. The product of two 0-1 decision variables makes the constraints (3.10) nonlinear. The method suggested by Glover and Woolsey (1973) can be used to linearize the constraints and is explained below. Consider the product term $r_{kc}.X_{ms}(kp)$, where both $\cdot\cdot_{kc}$ and $X_{ms}(kp)$ are 0-1 integer variables. Each product term is now replaced by a continuous linearization variable $L_{cms}(kp)$ and additional constraints (3.21) and (3.22). The linearized model can be stated as follows.

(IPM-3) Linearized

$$\text{Minimize } f_3 = \sum_{mc} C_m Z_{mc} \qquad (3.16)$$

subject to:

$$\sum_{p} Y_{kp} = 1 \qquad \forall \quad k \qquad (3.17)$$

$$\sum_{m} \alpha_{ms} X_{ms}(kp) = a_s(kp)\ Y_{kp} \qquad \forall \quad s,k,p \qquad (3.18)$$

$$\sum_{kps} d_k\ L_{cms}(kp)\ t_{ms}(kp) \le b_m Z_{mc} \qquad \forall \quad m,c \qquad (3.19)$$

$$\sum_{kpms} d_k\ X_{ms}(kp)\ c_{ms}(kp) \le B \qquad \forall \qquad (3.20)$$

$$r_{kc}\cdot\sum_{msp} \alpha_{ms} a_s(kp) + \sum_{msp} \alpha_{ms} a_s(kp) X_{ms}(kp) - \sum_{msp} L_{cms}(kp) \le \sum_{msp} \alpha_{ms} a_s(kp) \qquad \forall \quad k,c \qquad (3.21)$$

$$r_{kc} \ge L_{cms}(kp)$$

$$\forall \quad c,m,s,k,p \quad (3.22)$$

$$X_{ms}(kp) \ge L_{cms}(kp)$$

$$\sum_{c} r_{kc} = 1 \qquad \forall \quad k \qquad (3.23)$$

$$\sum_{m} Z_{mc} \le MAX_c \qquad \forall \quad c \qquad (3.24)$$

$$\begin{array}{llll} L_{cms}(kp) \geq 0 & \forall & c, m, s, k, p & \\ X_{ms}(kp) \quad - (0,1) & \forall & m, s, k, p & \\ Y_{kp} \quad - (0,1) & \forall & k, p & (3.25) \\ Z_{mc} \geq 0 \text{ and integer} & \forall & m, c & \end{array}$$

The number of 0-1 variables, integer variables and the constraints for MODEL-3 are ($\sum_k P_k + \sum_{kpms} a_s(kp)\, \alpha_{ms} + KC$), MC, and ($K + \sum_{kp} S(k,p) + MC + 1 + K + C$) respectively. Besides ($C. \sum_{kpms} a_s(kp)\, \alpha_{ms}$) additional continuous variables and ($2C. \sum_{kpms} a_s(kp)\, \alpha_{ms} + KC$) constraints are introduced due to linearization.

3.2 Example and Analysis of Results

An illustrative example is given for all the models. Four different part types of known demand are manufactured each having 2, 2, 3 and 2 process plans as given in Table-3.1. Each operation in a plan can be performed on alternate machines. Three types of machines of known capacity and capital cost are available and their compatibility to perform an operation is given in Table-3.2. The time and cost information for performing an operation on compatible machines for a process plan is given in Table-3.3. Using the data given in Table-3.1, 3.2 and 3.3, the first model was solved and the results are given in Table-3.4. For solving the second model the additional information on part families is given in Table-3.5. The results obtained by solving the second model with data from Table-3.1, 3.2, 3.3 and 3.5 is shown in Table-3.6. The information on maximum number of machines in a group (cell) is needed to solve the third model. It is assumed that two cells have to be formed with a restriction of 2 machines in each cell. Using data given in Table-3.1, 3.2 and 3.3 the model was solved and the solution is given in Table-3.7. The budget available for the operating cost is assumed to be $ 200. All the models were solved using LINDO (PC Version). The nonlinear constraints in Model-3.3 were linearized using the method suggested by Glover and Woolsey (1973). The number of 0-1 variables, integer variables,additional continuous variables introduced due to linearization and the total number of constraints in the linearized Model-3.3 are 59, 6, 84 and 214 respectively.

The results shown in Table-3.6 and 3.7 indicate that both the models have selected the same process plans for each part. However, the objective function values (capital investment) given by Model-3.2 and Model-3.3 are 800 and 600 respectively. The presence of alternate machines and simultaneous formation of part families and machine groups by Model-3.3 is the main reason for this resource savings. Moreover, Model-3.3 forms the natural part families which otherwise is assumed to be known for solving Model-3.2. This indicates that the sequential approach of forming part families and assigning machines to part families can lead to inferior performance in terms of resource utilization. Although this observation is made with only one cost vector, the same conclusion could be drawn for any other cost vector. This follows from noting that any solution, specifically, the optimal solution using the sequential approach, provides a feasible solution for the simultaneous approach. Thus, the simultaneous grouping model would provide results at least as good as the sequential model. In Model-3.2 and Model-3.3 it is assumed that exclusive cells are formed without any inter-cell movement. This leads to higher capital investment than that obtained by Model-3.1 where we consider all the parts in one cell.

Further, to analyze the influence of alternate process plans, Model-3.2 and Model-3.3 were solved by considering only one process plan for each part generated by Model-3.1. The results obtained are given in Table-3.8 and 3.9 and the objective function values are 900 and 850 for Model-3.2 and Model-3.3 respectively. However, when multiple process plans are considered, these values are 800 and 600 respectively. Both the models indicate the need for higher resource requirements in the absence of alternate process plans. It is worth mentioning that the third model can also be used to decide the optimal number of cells by specifying an overestimated value on the number of cells (C). Only the required number of cells will be formed. To illustrate this, the third model was solved by specifying C equal to 2 with a restriction of 2 and 3 machines in each cell. The model selected two machines of type 2, one machine of type 1 and all parts were identified in one cell.

3.4 Summary

The design of a cellular manufacturing system is dependent on the choice of process plan and machine tools. The models developed capture the flexibility available in considering alternate process plans and machine types for forming part families and machine groups. The first model gives us the part/machine incidence matrix which can be used for cell formation using one of the currently available techniques. The second model can be used to form machine groups assuming that the part families are known. The third model identifies part families and machine groups simultaneously. All the models select a process plan for a part and assign machines to operations considering demand, time and cost constraints. It was shown that alternate routing in considering alternate process plans/machines and simultaneous formation of part families and machine groups can result in efficient resource utilization. Here, it may be emphasized that these models cannot be implemented for large size problems because of the 0-1 and general integer programming nature of the formulations.

TABLE 3.1: Data on $a_s(kp)$ indicating operation s of part k to be performed for the process plan p; and the demand (d_k) for the part k.

	part k=1		part k=2		part k=3			part k=4	
	process plan p=1	p=2	process plan p=1	p=2	process plan p=1	p=2	p=3	process plan p=1	p=2
operation									
s=1	1*		1		1		1	1	1
s=2	1	1	1	1	1	1	1	1	1
s=3		1	1	1		1	1	1	
DEMAND	10		10		10			10	

*$a_1(11)=1$ indicates operation 1 has to be performed on part 1 if process plan 1 is selected.

TABLE 3.2: Data on α_{ms} indicating if operation s can be performed on machine m; capacity (b_m) on machine m; and the cost (C_m) of machine m.

	MACHINE m=1	m=2	m=3
s=1	1		1
s=2		1	1
s=3	1	1	
CAPACITY	100	100	100
COST	100	250	300

TABLE 3.3: The time ($t_{ms}(kp)$) and cost ($c_{ms}(kp)$) required for machine m to perform operation s on part k using process plan p.

		k=1		k=2		k=3			k=4	
		p=1	p=2	p=1	p=2	p=1	p=2	p=3	p=1	p=2
s=1	m=1	5,3		3,4		2,2		8,1	1,2	9,7
	m=3	7,2		4,3		2,2		9,2	2,1	8,9
s=2	m=2	3,5	9,8	7,8	3,3	3,3	1,2	5,9	2,3	9,8
	m=3	4,3	7,9	7,7	2,3	4,4	2,4	3,10	2,4	10,9
s=3	m=1		8,8	10,9	6,5		11,7	7,4	3,5	
	m=2		7,7	8,9	6,6		8,8	9,5	2,5	

TABLE 3.4: Solution for Model-3.1.

a) Objective function value = 550

b) Indicates the plan selected p and machine selected m for operation s.

	k=1	k=2	k=3	k=4
	p=1	p=2	p=2	p=1
s=1	m=1			m=1
s=2	m=2	m=2	m=2	m=2
s=3		m=1	m=1	m=1

c) Optimum number of machines of type m selected.

	No. of machines
m=1	3
m=2	1
m=3	0

TABLE 3.5: Data on β_{kf} indicating if part k is a member of part family f.

	part			
	k=1	k=2	k=3	k=4
family				
f=1	1	1		
f=2			1	1

TABLE 3.6:Solution for Model-3.2.

a)Objective function value = 800

b)Indicates the plan selected p and machine selected m for operation s.

	k=1	k=2	k=3	k=4
	p=1	p=2	p=1	p=1
s=1	m=1		m=1	m=1
s=2	m=2	m=2	m=2	m=2
s=3		m=1		m=1

c)Optimum number of each machine type m assigned to each cell c

	cell c=1	cell c=2
m=1	2	1
m=2	1	1
m=3	0	0

TABLE 3.7:Solution for Model-3.3.

a) Objective function value = 600

b)Indicates the plan selected p and machine selected m for operation s.

	k=1	k=2	k=3	k=4
	p=1	p=2	p=1	p=1
s=1	m=1		m=1	m=1
s=2	m=2	m=2	m=2	m=2
s=3		m=2		m=2

c)Data on r_{kc} indicating if part k is a member of cell c.

		part		
	k=1	k=2	k=3	k=4
cell				
c=1		1		
c=2	1		1	1

d)Optimum number of each machine type m assigned to each cell c.

	cell	cell
	c=1	c=2
m=1	0	1
m=2	1	1
m=3	0	0

TABLE 3.8:Solution for Model-3.2.

a)Objective function value = 900

b)Indicates the machine selected m for operation s with a fixed plan p for each part k.

	k=1	k=2	k=3	k=4
	p=1	p=2	p=2	p=1
s=1	m=1			m=1
s=2	m=2	m=2	m=2	m=2
s=3		m=1	m=1	m=1

c)Optimum number of each machine type m assigned to each cell c

	cell	cell
	c=1	c=2
m=1	2	2
m=2	1	1
m=3	0	0

TABLE 3.9:Solution for Model-3.3.

a) Objective function value = 850

b)Indicates the machine selected m for operation s with a fixed plan p for each part k.

	k=1	k=2	k=3	k=4
	p=1	p=2	p=2	p=1
s=1	m=1			m=1
s=2	m=2	m=2	m=2	m=2
s=3		m=2	m=2	m=2

c)Data on r_{kc} indicating if part k is a member of cell c.

	part			
	k=1	k=2	k=3	k=4
cell				
c=1	1			1
c=2		1	1	

d)Optimum number of each machine type m assigned to each cell c

	cell c=1	cell c=2
m=1	1	0
m=2	1	2
m=3	0	0

CHAPTER 4

CELL DESIGN WITH INVESTMENT CONSIDERATIONS

Cellular Manufacturing has been recognized as a critical element in the revival of outdated and unproductive plants. New technologies often support and mandate CM. In the past cells were often created by rearranging existing job shops with conventional machinery. The current move towards computer-integrated manufacturing (CIM) has focused our interests in the creation of cells using new and often automated machines.

4.1 Mathematical Formulations

In this chapter, a number of mathematical models are developed for the creation of new cells. It is assumed that a part can be produced through one or more process plans. Each operation in a process plan can be performed on alternate machines. Thus, for each process plan we have a number of production plans depending on the machines selected for each operation. It is also assumed that the demand for a part could be split and can be produced in more than one cell. The plans identified to produce the same part in different cells could be different.

Thus, the cell formation problem, in addition to forming part families and machine groups is to specify the plans selected for each part, quantity to be produced through the plans selected, machine type to perform each operation in the plans and total number of machines required considering demand, time, material handling and resource constraints.

The ultimate application of GT in manufacturing is the formation of mutually exclusive cells. If it is assumed that there is only one unique process plan for each part the creation of mutually independent cells may not be possible without duplication of machines. The duplication of machines require additional capital investment. However, if we permit alternate process plans for each part and assign operations to machines during cell formation, it may be possible to select process plans which can be processed within a cell without additional investment. We therefore consider the minimization of investment as one of the criteria for design of cellular manufacturing systems. Accordingly, four mathematical models are developed for cell formation. All the four models developed are large scale mixed integer programs. Since the number of machines of each type selected is restricted to be non-negative integers, the models implicitly minimize the under-utilization of machines. The option to select more than one plan for a part and produce it in more than one cell allows us to derive benefit from flexibility. The assumptions and notation are stated before developing the models.

Assumptions and Notation:

We have M (m=1,2....M) machine types and K (k=1,2.......,K) parts to be manufactured. The demand for a part k is bounded from below by d_k. Machine type m is available for b_m units of time during the plan horizon. A part k (k=1,2....K) can be manufactured through any of the P_k process plans. A process plan for a part (say part k, plan p) can be viewed as a set of operations :1,2......S(kp) to be performed. Each operation can be performed on alternate machines. For each part and process plan combination, the cost and time for each machine allotted to each operation are given. Accordingly,

$$t_{ms}(kp) = \begin{cases} \text{time for machine } m \text{ to perform operation } s \text{ for} \\ \text{(kp) combination.} \\ \propto \ \text{ if machine } m \text{ cannot perform this operation.} \end{cases}$$

and

$$c_{ms}(kp) = \begin{cases} \text{cost for machine } m \text{ to perform operation } s \text{ for} \\ \text{(kp) combination.} \\ \propto \ \text{ if machine } m \text{ cannot perform this operation.} \end{cases}$$

We now define a production plan to be an assignment of machines to operations for a given (kp) combination. Several production plans could be used to produce part k using process plan p. Thus, we define the following :

$X(lkp)$ = Number of units of part k produced using process plan p and production plan l.

$L(kp)$ = Number of different production plans for (kp) combination.

We have a large number of production plans for (kp)combinations. Every feasible assignment of machines to operations in the process plan gives rise to a production plan. Let a production plan be designated by numbers $a(lkp)$ for $l \ \varepsilon \ L(kp)$ so that:

$$a_{ms}(lkp) = \begin{cases} 1 & \text{if in plan } l \text{ machine } m \text{ is assigned to operation } s \\ & \text{for (kp) combination.} \\ 0 & \text{otherwise.} \end{cases}$$

Model-4.1:

The objective of this model is to assign machines to parts. This is the simplest situation leading to a single cell, when the number of part types and machines required is few. A typical cell consists of 3 to 15 machines (Wemmerlov and Hyer 1987). This can also be referred to as a process plan and operation allocation problem. The model is formulated as follows:

(MIP-1)

Minimize M41 = $\sum_{m} C_m Z_m$

subject to:

$$\sum_{pl} X(lkp) \geq d_k \qquad \forall \quad k \quad (4.1)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkp) \;\; t_{ms}(kp) \right) X(lkp) \leq b_m Z_m \qquad \forall \quad m \quad (4.2)$$

$$\sum_{kpl} \left(\sum_{ms} a_{ms}(lkp) \;\; c_{ms}(kp) \right) X(lkp) \leq B \qquad (4.3)$$

Z_m non-negative integer variables (4.4)

$$X(lkp) \geq 0 \qquad \forall \quad l,k,p$$

The objective function minimizes the total investment on machines assigned to all the parts. Constraints (4.1) guarantee that the lower bound on demand for all parts is met. Constraints (4.2) ensure that the capacity of each machine type selected is not violated. Constraint (4.3) restricts the operating cost of producing all parts to the budget. Constraints (4.4) indicate the integer variables.

Model-4.2:

This model adopts a sequential approach to cell formation by assigning machines to known part families. This model has wide applicability because a number of companies have indicated the use of one or more classification schemes in conjunction with GT applications for determining part families. The part families thus formed were determined without reliance on production methods (Wemmerlov and Hyer 1986, Hyer and Wemmerlov 1985). For developing this model we assume that part families are known. Information on which part belongs to which part family is denoted by an indicator β_{kf}.

$$\beta_{kf} = \begin{cases} 1 \text{ if part k belongs to family f.} \\ 0 \text{ otherwise.} \end{cases}$$

Accordingly, the model is:

(MIP-2)

$$\text{Minimize M42} = \sum_{mf} C_m Z_{mf}$$

subject to:

$$\sum_{pl} X(lkp) \geq d_k \qquad \forall \quad k \qquad (4.5)$$

$$\sum_k \beta_{kf} \sum_{pl} \left(\sum_s a_{ms}(lkp) \; t_{ms}(kp) \right) X(lkp) \leq b_m Z_{mf} \qquad \forall \quad m,f \qquad (4.6)$$

$$\sum_{kpl}\left(\sum_{ms} a_{ms}(lkp)\; c_{ms}(kp)\right) X(lkp) \le B \qquad (4.7)$$

$$Z_{mf} \text{ non-negative integer variables} \qquad (4.8)$$

$$X(lkp) \ge 0 \qquad \forall\ l,k,p$$

The objective function minimizes the total investment on machines assigned to all the part families. Constraints (4.5),(4.7) and (4.8) correspond to (4.1),(4.3) and (4.4) of Model-4.1 respectively. Constraints (4.6) ensure that the capacity of each machine type assigned to all the part families is not violated.

Model-4.3:

For developing Model-4.2 we assumed that the part families are known. This method may not uncover natural part families, because the part families were formed based on part attributes and were not based on production methods. Moreover, not all companies have well developed coding schemes to establish part families. To discover natural part families and machine groups we have to form them simultaneously. The objective of this model is to identify part families and machine groups simultaneously. For developing this model we define the decision variable to reflect the fact that the demand for a part can be allowed to be produced in more than one cell. The plans chosen to produce these parts in the respective cells may be different but they are exclusively processed in that cell with no inter cell movement. The model is stated after defining the following:

$X(lkpc)$= Number of units of part k produced using process plan p and production plan l in cell c.

$L(kpc)$ = Number of different production plans for (kpc) combination.

$$a_{ms}(lkpc) = \begin{cases} 1 & \text{if in plan } l \text{ machine m is assigned to operation s} \\ & \text{for (kpc) combination.} \\ 0 & \text{otherwise.} \end{cases}$$

(MIP-3)

Minimize $M43 = \sum_{mc} C_m Z_{mc}$

subject to:

$$\sum_{cpl} X(lkpc) \geq d_k \qquad \forall \ k \quad (4.9)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpc) \ t_{ms}(kp) \right) X(lkpc) \leq b_m Z_{mc} \qquad \forall \ m,c \ (4.10)$$

$$\sum_{ckpl} \left(\sum_{ms} a_{ms}(lkpc) \ c_{ms}(kp) \right) X(lkpc) \leq B \qquad (4.11)$$

$$\sum_{m} Z_{mc} \leq Max_c \qquad \forall \ c \quad (4.12)$$

$$Z_{mc} \text{ non-negative integer variables} \qquad (4.13)$$

$$X(lkpc) \geq 0 \qquad \forall \ l,k,p,c$$

The value of C (c=1,...,C), which denotes the number of cells is based on judgment. It is suggested that this value be an overestimate. Only the required number of cells will be formed leaving the other cells empty.

From an operational control point of view if the decision maker feels the demand for a part should be produced in only one cell and does not comply with our assumption, the following additional constraints will ensure that:

$$\sum_{c} r_{kc} = 1 \qquad \forall \quad k \qquad (4.14)$$

$$\sum_{pl} X(lkpc) \leq M\, r_{kc} \qquad \forall \quad k, c,\ M \text{ is a large number} \qquad (4.15)$$

$$r_{kc} = 0 \text{ or } 1 \qquad \forall \quad k, c$$

Constraints (4.14) guarantee that a part is assigned to only one cell and Constraints (4.15) ensure that a plan is identified only in the cell in which the part has been assigned. This model can be briefly stated as:

Model-4.3.1:

(MIP-3.1)

Minimize M43

subject to : Constraints (4.9) to (4.15).

This model achieves the ultimate goal of GT application in manufacturing by grouping parts and machines simultaneously to form mutually exclusive cells.

Model-4.4:

If in Model-4.2 the number of part families identified is large or for some practical reasons the company wants a few sets of parts to be always produced in the same cell then we can assign one or more part families to a cell. We develop Model-4.4 to address this situation. It forms part groups and machine groups simultaneously by assigning one or more part families to cells. We define the following decision variable before stating the model.

$X(lkpcf)$= Number of units of part k in family f produced using process plan p and production plan l in cell c.

$$a_{ms}(lkpcf) = \begin{cases} 1 & \text{if in plan } l \text{ machine m is assigned to operation s for (kpcf) combination.} \\ 0 & \text{otherwise.} \end{cases}$$

and

$$r_{fc} = \begin{cases} 1 & \text{if part family f is assigned to cell c.} \\ 0 & \text{otherwise.} \end{cases}$$

(MIP-4)

Minimize M44 = $\sum_{mc} C_m Z_{mc}$

subject to:

$$\sum_{cpl} X(lkpcf) = \beta_{kf}\, d_k \qquad \forall \quad k,f \qquad (4.16)$$

$$\sum_{fkpl} \left(\sum_{s} a_{ms}(lkpcf)\; t_{ms}(kp) \right) X(lkpcf) \le b_m\, Z_{mc} \qquad \forall \quad m,c \qquad (4.17)$$

$$\sum_{cfkpl} \left(\sum_{ms} a_{ms}(lkpcf)\; c_{ms}(kp) \right) X(lkpcf) \le B \qquad (4.18)$$

$$\sum_{m} Z_{mc} \le Max_c \qquad \forall \quad c \qquad (4.19)$$

$$\sum_{c} r_{fc} = 1 \qquad \forall \quad f \qquad (4.20)$$

$$\sum_{kpl} X(lkpcf) \leq M\, r_{fc} \forall \quad f, c,\ M \text{ is a large number} \qquad (4.21)$$

$$Z_{mc} \text{ non-negative integer variables} \qquad (4.22)$$

$$r_{fc} \text{ 0-1 variables}$$

$$X(lkpcf) \geq \ 0 \qquad \forall \quad l, k, p, c, f$$

In Constraints (4.16) we have an equality sign to ensure that parts are produced only if it belongs to that particular part family. Constraints (4.20) guarantee that a part family is assigned to only one cell. Constraints (4.21) ensure that a plan for a part is selected only if part k belongs to family f and family f is in cell c. The other constraints are similar to those described in the previous models.

4.3 Solution Methodology

The method of solution for Model-4.1 is described in detail first and extended to the other models.

Method of Solution for Model-4.1:

In this formulation L(kp) is large for each (kp) combination and, therefore, the number of variables will be large. Moreover, the integer restriction on Z_m further complicates the model. If we remove the integrality restriction on Z_m , we have a large scale linear program. A direct application of the simplex method is not practical. We will therefore solve this problem by the revised simplex method using a column generation scheme that will be developed in the next section. Once the linear program is solved a branch and bound algorithm on Z_m values gives an optimal solution. As already mentioned each node in the branch and bound tree represents a solution to an augmented continuous problem with additional constraints on the integer variables. These additional constraints are easily incorporated without increasing the size of the problem by the bounded variables procedure. The column generation scheme is given next.

Column Generation Scheme:

The approach to be considered here is a method of generating the desired column at each iteration of the simplex method. This strategy is adapted from the Dantzig-Wolfe Decomposition Principle (Dantzig and Wolfe 1961). The method of generating columns is different in each case, depending upon the special structure of the problem (Chandrasekaran,Aneja and Nair 1984, Gilmore and Gomory 1961). In the problem under consideration column generation is achieved by solving a simple assignment sub-problem. The method of generating the column is given below.

At any general iteration, let us define the simplex multipliers corresponding to (4.1),(4.2) and (4.3) as $\pi_k (k=1,2,.....K)$, $u_m (m=1,2,....M)$ and v respectively. Now the pricing scheme for determining the entering variable, if any, is to look for any variable $X_{(lkp)}$ such that the associated reduced cost $\bar{C}_{(lkp)} = C_{(lkp)} - Z_{(lkp)}$ is negative. Since $X_{(lkp)}$ does not appear in the objective function the

C(lkp) value is zero. Thus we have Z(lkp) > 0. Accordingly,

$$\pi_k + \sum_m u_m \sum_s a_{ms}(lkp)\ t_{ms}(kp) + v \sum_{ms} a_{ms}(lkp)\ c_{ms}(kp) > 0 \qquad \forall \quad k$$

$$\text{or, } \sum_{ms} a_{ms}(lkp)\ [\ (-u_m)\ t_{ms}(kp) + (-v)\ c_{ms}(kp)\ ] < \pi_k \qquad \forall \quad k$$

Defining, $cc_{ms}(kp) = (-u_m)\ t_{ms}(kp) + (-v)\ c_{ms}(kp)$ (4.23)

$$Z(lkp) > 0 \iff \sum_{ms} a_{ms}(lkp)\ cc_{ms}(kp) < \pi_k \qquad \forall \quad k$$

Thus, for a fixed k,p consider the following assignment problem of assigning machines to operations.

Define 0-1 variables a_{ms} as

$$a_{ms} = \begin{cases} 1 \text{ if machine m is assigned to perform operation s.} \\ 0 \text{ otherwise} \end{cases} \qquad \forall \quad m,s$$

Let $cc_{ms}(kp)$ = cost of assigning machine m to perform operation s.

The problem is:

$$\text{Minimize } Z = \sum_{sm} cc_{ms}(kp)\ a_{ms}$$

subject to :

$$\sum_m a_{ms} = 1 \qquad \forall \quad s$$

$$a_{ms} = 0 \text{ or } 1 \qquad \forall \quad m,s$$

The optimal solution to this problem can be obtained by the following simple "greedy" procedure:

Let $m_s = \underset{m}{\text{Min}}\ cc_{ms}(kp) \qquad \forall \quad s$

Optimal assignment is given by:

$$a_{ms} = \begin{cases} 1 \text{ if } m = m_s \\ 0 \text{ otherwise} \end{cases}$$

Let Z^0 be the cost associated with this production plan. If $Z^0 < \pi_k$, then, it is a candidate column to enter the basis. The (K+M+1) column values of the entering column are

First K values $[\ 0\ 0\ \ldots\ldots 1_{k^{th}}\ 0]$ position.

Next M values $[\ \sum_s a_{1s} t_{1s},\ \sum_s a_{2s} t_{2s}, \ldots\ldots\ldots \sum_s a_{Ms} t_{Ms}]$

Last 1 value $[\ \sum_{ms} a_{ms} c_{ms}\ ]$

If for every (kp) combination the optimal assignment $Z^0 > \pi_k$ we check if any of the slack, surplus or Z column can enter. These columns are explicitly known. Thus the relaxed MIP-1 can be solved. Branch and Bound on Z_m variables will solve the original mixed integer program optimally (Dakin 1965).

Branch and Bound on Z_m:

Model-4.1 was solved as a linear programming problem. Some of the variables namely Z_m are required to be integers. Suppose I is an integer. Then, since the range $I < Z_m < I+1$ is inadmissible, divide all the solutions to the constraints into two non-overlapping groups. i.e.,

(i) solutions in which $Z_m \leq I$.

(ii) solutions in which $Z_m \geq I+1$.

Thus we have two subproblems with an additional constraint at each branching node. The linear programs are again solved using the column generation approach. If this solution is integral then it is one solution to the problem and stop branching on that node. If it is non-integral, we again branch on the variable which is non-integral, add each of these constraints to the continuous problem and solve the augmented problems. Repeat the procedure for each of the two solutions so obtained. The branching will always terminate in one of two ways; we may reach an integer solution or we may find that the current set of constraints has no solutions. The solution to the complete problem will be the best integer solution reached in this way. Although we regarded the constraints on Z_m at each node additional to the original constraints in the linear program, it can be easily incorporated without increasing the size of the problem by the bounded variables procedure. Thus, by solving a large scale linear program of the same basis size at branching nodes, an optimal solution is obtained for the mixed integer program. The algorithm for solving Model-4.1 is given next.

Algorithm-4.1:

Step-0: Start the initial basis with all the slack and artificial variables.

Step-1: Choose a part k and process plan p and the assignment cost matrix as given in (4.23).

Step-2: Find a minimal cost assignment such that

$C^1 = \sum_{ms} a_{ms}(lkp)\ cc_{ms}(kp) < \pi_k$ for all $\pi_k \geq 0$, $u_m \leq 0$ and $v \leq 0$.

If $C^1 > \pi_k$ for all (kp) combinations go to Step-5, else go to Step-3.

Step-3: Enter the new column and update the basis. Go to Step-4.

Step-4: Check for the surplus and slack variables to enter the basis. If

$\pi_k < 0$ introduce surplus variable corresponding to part k.

$u_m > 0$ introduce slack variable corresponding to machine m.

$v > 0$ introduce slack variable corresponding to budget.

If any of them can enter go to Step-3, else go to Step-5.

Step-5: Check if any of the Z_m column can enter. If, yes then go to Step-3, else go to Step-6.

Step-6: If Z_m values are integers then stop else branch and bound on Z_m. Add the additional constraints and go to Step-1.

Method of Solution for Model-4.2 and Model-4.3:

The method of generating the columns for MIP-2 and MIP-3 are similar to MIP-1 as described above. The assignment costs for the sub-problem in MIP-2 and MIP-3 are:

$$cc_{ms}(kp) = (-u_{mf}\ \beta_{kf})\ t_{ms}(kp) + (-v)\ c_{ms}(kp) \qquad \forall \quad k,p,f$$

and

$$cc_{ms}(kp) = (-u_{mc}\ r_{kc})\ t_{ms}(kp) + (-v)\ c_{ms}(kp) \qquad \forall \quad k,p,c$$

for a fixed k,p,f and k,p,c respectively. u_{mf} and u_{mc} are the multipliers corresponding to constraints (4.6) and (4.10) respectively. The number of column values of the entering column will be (K+MF+1) and (K+MC+1). Algorithm-4.1 with the new assignment cost matrix solves these two models.

Method of Solution for Model-4.3.1 and Model-4.4:

For developing Model-4.3.1 and Model-4.4 we have introduced additional 0-1 variables, namely, r_{kc} and r_{fc}. They appear in the Constraints (4.14),(4.15) and (4.20),(4.21) respectively. We suggest an implicit enumeration on these variables to solve the models optimally. The algorithm for solving Model-4.3.1 is given next.

Algorithm-4.2:

Step-0: Solve the mixed integer program (MIP-3) optimally using Algorithm-1. If the solution to this program specifies that each part type should be produced in only one cell then you have the required solution hence stop else go to Step-1.

Step-1: Select any part k that has been identified to be produced in more than one cell. This can be known by looking at the plans in the basis, i.e., a few r_{kc} variables are fractions. An implicit enumeration on r_{kc} will solve the model optimally. For example, if part 1 is produced in two cells 1 and 2, to ensure it is produced in only one we branch into two nodes. At node 1 we set r_{11}=1 and at node 2 r_{11}=0. The problem is solved again using Algorithm-1, except that now in the branch r_{11}=1, we select only plans for part 1 in cell 1 to enter the basis while checking for column entering and do not consider other c=2,...C for k=1. We do just the opposite in branch 2 where r_{11}=0. We consider all plans for $c \neq 1$ for k=1. At every branched node we solve a mixed integer program. Model-4.4 can be solved similarly.

4.4 Examples

Consider a hypothetical example of just one part to be manufactured for purpose of illustration of Model-4.1. Two process plans have been identified for manufacturing this part. Each plan requires two operations. Two types of machines costing 100 and 200 each are available to perform these operations. The machine availabilities are 100 time units for each type. The demand for the part is 100 units and has to be produced within an operating budget of 500. Thus we have M=2, K=1, d_1 =100, p_1=2, r(11)=r(12)= 2, b_1= b_2= 100 and B =500. The production cost and time for each of the two plans are given by the following matrices:

$$k=1,\ p=1,\ c_{ms}(11) \overset{m\ s \rightarrow}{\downarrow}= \begin{bmatrix}100 & 100\\ 2 & 2\end{bmatrix} \qquad t_{ms}(11) = \begin{bmatrix}100 & 100\\ 3 & 2\end{bmatrix}$$

$$k=1,\ p=2,\ c_{ms}(12) = \begin{bmatrix}2 & 2\\ 100 & 100\end{bmatrix} \qquad t_{ms}(12) = \begin{bmatrix}3 & 2\\ 100 & 100\end{bmatrix}$$

There are a total of 8 plans for this problem. The solution for this problem is rather obvious. The application of the algorithm to this example problem is illustrated next.

The method is started with all artificial and slacks in the basis. The initial basic variables column is [$\bar{s}_1$,s_2,s_3,s_4], the right hand side column is [100,0,0,500] and the dual variables are [M,0,0,0]. M in this context is a very large number. Since the dual variables corresponding to machines and budget is zero any plan can enter the basis. Say we assign machine type 1 to both operations in p=1. The column for the basis entry is P^1[1,200,0,200]. The basis and the inverse are updated by the usual simplex rules. The new basic column is [$\bar{s}_1$,P^1,s_3,s_4], the right hand side column is [100,0,0,500] and the dual variables are [M,-M/200,0,0]. With the new dual variables find the assignment cost and pick, for each operation, the machine with lowest cost. The column entering, basic column, RHS column and dual variables at following iterations are given below.

Column entering	Basic column	RHS column	Dual variables
P^2 [1,0,5,4]	[$\bar{s}_1$,P^1,P^2,s_4]	[100,0,0,500]	[M,-M/200,-M/5,0]
P^3 [1,5,0,4]	[$\bar{s}_1$,P^3,P^2,s_4]	[100,0,0,500]	[M,-M/5,-M/5,0]

With the present assignment cost no more plans can enter the basis. Now check if any of the Z_m column can enter. The Z column corresponding to machine type 1 is Z_1[0,-100,0,0]. Since Cj-Zj <0 Z_1 can enter the basis. Thus we have,

Column entering	Basic column	RHS column	Dual variables
Z^1 [0,-100,0,0]	[Z_1,P^3,P^2,s_4]	[5,100,0,100]	[5,-1,-1,0]

Check again if any plan, surplus, slack or Z column can enter before terminating.

Let us consider the following data for illustrating the application of Model-4.2 and Model-4.3

$K=4$, $M=2$, $d_1= d_2= d_3= d_4= 10$, $P_1= P_2= P_3= P_4= 1$, $r(11) = r(21) = r(31) = r(41) = 2$, $b_1= b_2= 150$, $B=700$, $Max_1= Max_2= 2$, $C_1= 100$, $C_2= 200$,$r_{11} = r_{31} = r_{22} = r_{42} = 1$, and

$$k=1,\ p=1,\ c_{ms}(11) = \begin{bmatrix} 5 & 5 \\ 100 & 100 \end{bmatrix} \qquad t_{ms}(11) = \begin{bmatrix} 5 & 5 \\ 100 & 100 \end{bmatrix}$$

$$k=2,\ p=1,\ c_{ms}(21) = \begin{bmatrix} 10 & 10 \\ 100 & 100 \end{bmatrix} \qquad t_{ms}(21) = \begin{bmatrix} 10 & 10 \\ 100 & 100 \end{bmatrix}$$

$$k=3,\ p=1,\ c_{ms}(31) = \begin{bmatrix} 100 & 100 \\ 5 & 5 \end{bmatrix} \qquad t_{ms}(31) = \begin{bmatrix} 100 & 100 \\ 5 & 5 \end{bmatrix}$$

$$k=4,\ p=1,\ c_{ms}(41) = \begin{bmatrix} 100 & 100 \\ 10 & 10 \end{bmatrix} \qquad t_{ms}(41) = \begin{bmatrix} 100 & 100 \\ 10 & 10 \end{bmatrix}$$

The column entering, basic column, RHS column and dual variables at each iteration are listed in Table-4.1 and Table-4.2 for Model-4.2 and Model-4.3 respectively. For Model-4.2, the basic column and RHS column in optimal solution are:

Basic Column $[Z_{11}, Z_{12}, Z_{21}, Z_{22}, P^1, P^3, P^4, P^5, s_9]$

RHS Column [10/15, 20/15, 10/15, 20/15, 10, 10, 10, 10, 300]

A branch and bound on Z_{mf} will give $Z_{11}= 1$, $Z_{12}= 2$, $Z_{21}= 1$ and $Z_{22}= 2$. The optimal solution at the first node for Model-4.3 is given in Table-4.2 (Iteration # 13).

4.5 Summary

In this chapter four large scale mixed integer programs were developed for operations allocation and cell design. Model-4.1 assigns machines to parts. Model-4.2 assigns machines to known part families to form mutually exclusive cells. The part families so known are generally formed based on part attributes. This method adopts a sequential approach tւ cell formation and may not uncover natural part families leading to poor resource utilization. Therefore, Model-4.3 was developed, which identifies part families and machine groups simultaneously. If the number of part families identified is large it may be economical to assign more than one part family to a cell and identify the machine groups. Model-4.4 groups part families and machines. All the four models specify the plans selected for each part, quantity to be produced through the plans selected, machine type to perform each operation in the plans and the total number of machines required to process all the parts by considering demand, time and resource constraints. The objective function of the models is to minimize capital investmei.. A column generation scheme was provided for an efficient solution to the relaxed mixed integer programs. The solution technique generates columns (plans) for each part type by solving simple assignment problems. Illustrative examples were solved to illustrate the solution technique.

TABLE 4.1: Computational details for solving Model-4.2.

Iteration # 0:	
Basic Column	$^*[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, s_5, s_6, s_7, s_8, s_9]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700]
Dual Column	[M, M, M, M, 0, 0, 0, 0, 0]
Iteration # 1:	
Column Entering	$P^1[1, 0, 0, 0, 10, 0, 0, 0, 10]$
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, s_6, s_7, s_8, s_9]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700]
Dual Column	[M, M, M, M, -M/10, 0, 0, 0, 0]
Iteration # 2:	
Column Entering	$P^2[1, 0, 0, 0, 0, 200, 0, 0, 200]$
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^2, s_7, s_8, s_9]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700]
Dual Column	[M, M, M, M, -M/10, -M/200, 0, 0, 0]
Iteration # 3:	
Column Entering	$P^3[0, 0, 1, 0, 0, 10, 0, 0, 10]$
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^3, s_7, s_8, s_9]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700]
Dual Column	[M, M, M, M, -M/10, -M/10, 0, 0, 0]
Iteration # 4:	
Column Entering	$P^4[0, 0, 1, 0, 0, 10, 0, 0, 10]$
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^3, P^4, s_8, s_9]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700]
Dual Column	[M, M, M, M, -M/10, -M/10, -M/20, 0, 0]
Iteration # 5:	
Column Entering	$P^5[0, 0, 0, 1, , 0, 0, 0, 20, 20]$

Basic Column $[\bar{s}_1,\bar{s}_2,\bar{s}_3,\bar{s}_4,P^1,P^3,P^4,P^5,s_9]$

RHS Column $[10,10,10,10,0,0,0,0,700]$

Dual Column $[M,M,M,M,-M/10,-M/10,-M/20,-M/20,0]$

Iteration # 6:

Column Entering $Z_{11}[0,0,0,0,-150,0,0,0,0]$

Basic Column $[Z_{11},\bar{s}_2,\bar{s}_3,\bar{s}_4,P^1,P^3,P^4,P^5,s_9]$

RHS Column $[10/15,10,10,10,10,0,0,0,600]$

Dual Column $[M/15,M,M,M,-M/150,-M/10,-M/20,-M/20,0]$

Iteration # 7:

Column Entering $Z_{21}[0,0,0,0,0,-150,0,0,0]$

Basic Column $[Z_{11},\bar{s}_2,Z_{21},\bar{s}_4,P^1,P^3,P^4,P^5,s_9]$

RHS Column $[10/15,10,10/15,10,10,10,0,0,500]$

Dual Column $[M/15,M,M/15,M,-M/150,-M/150,-M/20,-M/20,0]$

Iteration # 8:

Column Entering $Z_{12}[0,0,0,0,0,0,-150,0,0]$

Basic Column $[Z_{11},Z_{12},Z_{21},\bar{s}_4,P^1,P^3,P^4,P^5,s_9]$

RHS Column $[10/15,20/15,10/15,10,10,10,10,0,300]$

Dual Column $[M/15,2M/15,M/15,M,-M/150,-M/150,-M/150,-M/20,0]$

Iteration # 9:

Column Entering $Z_{22}[0,0,0,0,0,0,0,-150,0]$

Basic Column $[Z_{11},Z_{12},Z_{21},Z_{22},P^1,P^3,P^4,P^5,s_9]$

RHS Column $[10/15,20/15,10/15,20/15,10,10,10,10,300]$

Dual Column $[M/15,2M/15,M/15,2M/15,-M/150,-M/150,-M/150,-M/150,0]$

* $\bar{s}_1$ to $\bar{s}_4$ corresponds to constraints (4.5), s_5 to s_8 corresponds to constraints (4.6) and s_9 corresponds to constraint (4.7).

TABLE 4.2: Computational details for solving Model-4.3.

Iteration # 0:

Basic Column	$^*[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]
Dual Column	[M, M, M, M, 0, 0, 0, 0, 0, 0, 0]

Iteration # 1:

Column Entering	P^1[1, 0, 0, 0, 10, 0, 0, 0, 10, 0, 0]
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, s_6, s_7, s_8, s_9, s_{10}, s_{11}]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]
Dual Column	[M, M, M, M, -M/10, 0, 0, 0, 0, 0, 0]

Iteration # 2:

Column Entering	P^2[1, 0, 0, 0, 0, 0, 10, 0, 10, 0, 0]
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, s_6, P^2, s_8, s_9, s_{10}, s_{11}]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]
Dual Column	[M, M, M, M, -M/10, 0, -M/10, 0, 0, 0, 0]

Iteration # 3:

Column Entering	P^3[1, 0, 0, 0, 0, 200, 0, 0, 200, 0, 0]
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^3, P^2, s_8, s_9, s_{10}, s_{11}]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]
Dual Column	[M, M, M, M, -M/10, -M/200, -M/10, 0, 0, 0, 0]

Iteration # 4:

Column Entering	P^4[1, 0, 0, 0, 0, 0, 0, 200, 200, 0, 0]
Basic Column	$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^3, P^2, P^4, s_9, s_{10}, s_{11}]$
RHS Column	[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]
Dual Column	[M, M, M, M, -M/10, -M/200, -M/10, -M/200, 0, 0, 0]

Iteration # 5:

Column Entering	P^5[0, 0, 1, 0, 0, 10, 0, 0, 10, 0, 0]

Basic Column $[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^5, P^2, P^4, s_9, s_{10}, s_{11}]$

RHS Column $[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]$

Dual Column $[M, M, M, M, -M/10, -M/10, -M/10, -M/200, 0, 0, 0]$

Iteration # 6:

Column Entering $P^6[0, 0, 1, 0, 0, 0, 0, 10, 10, 0, 0]$

Basic Column $[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^5, P^2, P^6, s_9, s_{10}, s_{11}]$

RHS Column $[10, 10, 10, 10, 0, 0, 0, 0, 700, 2, 2]$

Dual Column $[M, M, M, M, -M/10, -M/10, -M/10, -M/10, 0, 0, 0]$

Iteration # 7:

Column Entering $Z_{11}[0, 0, 0, 0, -150, 0, 0, 0, 0, 1, 0]$

Basic Column $[Z_{11}, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^5, P^2, P^6, s_9, s_{10}, s_{11}]$

RHS Column $[10/15, 10, 10, 10, 10, 0, 0, 0, 600, 20/15, 2]$

Dual Column $[100/15, M, M, M, -10/15, -M/10, -10/15, -M/10, 0, 0, 0]$

Iteration # 8:

Column Entering $P^7[0, 1, 0, 0, 20, 0, 0, 0, 20, 0, 0]$

Basic Column $[Z_{11}, \bar{s}_2, \bar{s}_3, \bar{s}_4, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]$

RHS Column $[2, 0, 10, 10, 10, 0, 0, 0, 400, 10, 2]$

Dual Column $[M/2, M, M, M, -M/20, -M/10, -M/20, -M/10, 0, -15M/2+100, 0]$

Iteration # 9:

Column Entering $Z_{22}[0, 0, 0, 0, 0, 0, 0, -150, 0, 0, 1]$

Basic Column $[Z_{11}, \bar{s}_2, Z_{22}, \bar{s}_4, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]$

RHS Column $[2, 0, 10/15, 10, 10, 0, 0, 10, 300, 10, 20/15]$

Dual Column $[M/2, M, 200/15, M, -M/20, -20/15, -M/20, -20/15, 0, -15M/2+100, 0]$

Iteration # 10:

Column Entering $P^8[1, 0, 0, 0, 0, 200, 0, 0, 200, 0, 1]$

Basic Column $[Z_{11}, P^8, Z_{22}, \bar{s}_4, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]$

RHS Column $[2, 0, 10/15, 10, 10, 0, 0, 10, 300, 10, 20/15]$

Dual Column [800/3, 1600/3, 40/3, M, -80/3, -4/3, -80/3, -4/3, 0, -3900, 0]

Iteration # 11:

Column Entering P^9[0, 1, 0, 0, 0, 200, 0, 0, 200, 0, 1]

Basic Column [Z_{11}, P^9, Z_{22}, $\bar{s}_4$, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]

RHS Column [2, 0, 10/15, 10, 10, 0, 0, 10, 300, 10, 20/15]

Dual Column

[2000/15, 4000/15, 200/15, M, -200/15, -20/15, -200/15, -20/15, 0, -1900, 0]

Iteration # 12:

Column Entering P^{10}[0, 0, 0, 1, 0, 0, 0, 20, 20, 0, 0]

Basic Column [Z_{11}, P^9, Z_{22}, P^{10}, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]

RHS Column [2, 0, 2, 10, 10, 0, 0, 10, 100, 10, 0]

Dual Column

[2000/15, 4000/15, 200/15, 400/15, -200/15, -20/15, -200/15, -20/15, 0, -1900, 0]

Iteration # 13:

Column Entering Z_{12}[0, 0, 0, 0, 0, -150, 0, 0, 0, 1, 0]

Basic Column [Z_{11}, Z_{12}, Z_{22}, P^{10}, P^1, P^5, P^2, P^6, s_9, P^7, s_{11}]

RHS Column [2, 0, 2, 10, 10, 0, 0, 10, 100, 10, 0]

Dual Column

[100/15, 200/15, 200/15, 400/15, -10/15, -20/15, -10/15, -20/15, 0, 0, 0]

* $\bar{s}_1$ to $\bar{s}_4$ corresponds to constraints (4.15), s_5 to s_8 corresponds to constraints (4.16), s_9 corresponds to constraint (4.17) and s_{10}, s_{11}corresponds to constraints (4.18).

CHAPTER 5

CELL DESIGN WITH MACHINE RELOCATION CONSIDERATIONS

With the introduction of new parts and changed demands, new part families and machine groups have to be identified in a Cellular Manufacturing System. This may involve allocation of parts to existing machine groups, relocation expense on existing machines or additional investment on new machines. The redesign of cellular manufacturing systems warrants the consideration of these practical issues.

5.1 Mathematical Formulations

In this chapter, four mathematical models are developed to address these problems. In order to derive maximum benefits from CMS it is assumed that the cells are exclusive, i.e., no inter-cell movements of parts are allowed. The flexibility available in considering alternate process plans, alternate machines for performing the operations in the process plan and using more than one plan in one or more cells to produce the part is considered in the models developed. The first model is a large scale linear program. Parts are allocated to machine groups without disrupting the existing configuration. The two objectives which could be considered in this situation are either minimizing the operating cost or under utilization of machine groups. The second model is a large scale mixed integer program. Part families and machine groups are identified simultaneously with an objective to minimize the machine relocation cost and operating cost. The third model, in addition to simultaneously forming part families and machine groups, also determines the additional investment to be incurred on new machines. The objective function minimizes the relocation, operating and additional machine costs while assigning these new machines to one of the machine groups. This model is also a large scale mixed integer program. The fourth model considers only the investment on new machines assuming relocation is not allowed due to loss in production and time. A column generation scheme is developed for solving the first model efficiently. For solving the second model which is a mixed integer program, the relaxed linear problem is solved by column generation scheme followed by a branch and bound on the integer variables. At each branched node a relaxed linear problem is solved by the bounded variables procedure. The third and fourth models are solved similarly.

Model-5.1:

The objective here is to assign parts to known machine groups. It is assumed that the plan identified for a part in any machine group should be fully processed in that group. The objectives could be:

1. To minimize the total operating cost and

2. To minimize the under utilization of the machines in the machine groups.

Accordingly, Model-5.1 is stated for the two objectives as LP1 and LP2 respectively.

(LP 1) Minimizing the operating cost:

$$\text{Minimize } L51 = \sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \right) X(lkpg)$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (5.1)$$

$$\sum_{kpl}\left(\sum_{s} a_{ms}(lkpg)\ t_{ms}(kp)\right) X(lkpg) \leq b_{mg} \qquad \forall \quad m,g \qquad (5.2)$$

$$X(lkpg) \geq 0 \qquad \forall \quad l,k,p,g$$

The objective function minimizes the total cost of plans selected for all the parts. Constraints (5.1) guarantee that the demand for all parts are met. Constraints (5.2) ensure the capacity of each machine type in a group is not violated.

(LP 2) Minimizing the under utilization:

$$\text{Minimize } LS2 = \sum_{mg} d_{mg}$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (5.3)$$

$$\sum_{kpl}\left(\sum_{s} a_{ms}(lkpg)\ t_{ms}(kp)\right) X(lkpg) + d_{mg} = b_{mg} \qquad \forall \quad m,g \qquad (5.4)$$

$$\sum_{gkpl}\left(\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp)\right) X(lkpg) \leq B \qquad (5.5)$$

$$X(lkpg)\ ,\ d_{mg} \geq 0 \qquad \forall \quad l,k,p,g,m$$

The objective function minimizes the total under utilization of machines in all the groups. Constraints (5.3) assure the demand for parts is met. Constraints (5.4) ensure the capacity of each machine type in a group is not violated. Constraint (5.5) restrict the operating cost of producing all parts to the budget.

It may so happen that a few parts cannot be fully processed within an existing machine group. To derive maximum benefits from Group Technology it is desirable to process a part completely within a machine group. Two ways to achieve this could be:

1. to subcontract the parts which cannot be processed within a group;
2. to relocate the machines to form exclusive part families and machine groups.

The first solution needs no further comment. Relocation of machines is expensive and has to be done judiciously. For this purpose Model-5.2 is developed.

<u>Model-5.2:</u>

$$\text{Minimize } M52 = \sum_{mig} C_{mig}\, Z_{mig} + \sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg)\; c_{ms}(kp) \right) X(lkpg)$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (5.6)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpg)\; t_{ms}(kp) \right) X(lkpg) \leq b_m \left(N_{mg} + \sum_{i} Z_{mig} - \sum_{i} Z_{mgi} \right)$$

$$\forall \quad m, g \qquad (5.7)$$

$$\sum_{m} N_{mg} + \sum_{ml} Z_{mlg} - \sum_{ml} Z_{mgl} \leq Max_{g} \qquad \forall \quad g \qquad (5.8)$$

$$\sum_{l} Z_{mgl} - \sum_{l} Z_{mlg} \leq N_{mg} \qquad \forall \quad m, g \qquad (5.9)$$

Z_{mlg} non-negative integer variables. (5.10)

$$X(lkpg) \geq 0 \qquad \forall \quad l, k, p, g$$

The objective function consists of two costs terms, relocation cost of machines and operating cost which have to be minimized. If for any part an operation in the plan selected cannot be performed within a group, a very high cost is associated with that operation. This forces a few machines to be relocated such that the relocation costs are minimized and the plans selected can be fully processed within a cell. Constraints (5.6) force the demand for parts to be met. Constraints (5.7) ensure that all the machines required to process the part families identified are available in the machine groups identified. The maximum number of machines that can be in a machine group are imposed by Constraints (5.8). Constraints (5.9) ensure that the machines relocated to other cells do not exceed the number of machines of each type available in that cell. Constraints (5.10) indicate the integer variables.

The above model assumes that the capacity of available machines is enough. However, if the existing capacity is exceeded we need to know if this relocation should be accompanied or substituted by a higher degree of investment in new machines. Model-5.3 is developed for this purpose.

Model-5.3:

$$\text{Minimize } M53 = \sum_{mig} C_{mig} Z_{mig} + \sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \right) X(lkpg)$$

$$+ \sum_{mg} C_{mg} Z_{mg}$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (5.11)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpg)\ t_{ms}(kp) \right) X(lkpg) \leq b_m \left(N_{mg} + \sum_{i} Z_{mig} - \sum_{i} Z_{mgi} + Z_{mg} \right)$$

$$\forall \quad m,g \qquad (5.12)$$

$$\sum_{m} N_{mg} + \sum_{mi} Z_{mig} - \sum_{mi} Z_{mgi} + \sum_{m} Z_{mg} \leq Max_g \qquad \forall \quad g \qquad (5.13)$$

$$\sum_{i} Z_{mgi} - \sum_{i} Z_{mig} \leq N_{mg} \qquad \forall \quad m,g \qquad (5.14)$$

Z_{mig}, Z_{mg} non-negative integer variables. (5.15)

$$X(lkpg) \geq 0 \qquad \forall \quad l,k,p,g$$

The objective function includes the additional cost term C_{mg} due to investment on new machines of type m in group g. Z_{mg} are the additional variables introduced in this model. It indicates the number of machines of type m purchased and added to machine group g. This is added to

Constraints (5.12) to ensure that if a machine type m is purchased and added to group g, the additional capacity is available for parts allocated to the cell. Moreover, if an additional machine is added to a machine group the group size should not exceed the limit. This is ensured by Constraints (5.13).

If, however, the decision maker feels it is not desirable to relocate machines and instead would like to add new machines to the machine groups, a very high penalty can be attached to the relocation cost in Model-5.3. However, Model-5.3 can be simplified and restated to address this situation as follows:

Model-5.4:

$$\text{Minimize M54} = \sum_{mg} C_{mg}\, Z_{mg} + \sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg)\; c_{ms}(kp) \right) X(lkpg)$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (5.16)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpg)\, t_{ms}(kp) \right) X(lkpg) \leq b_m (N_{mg} + Z_{mg}) \qquad \forall \quad m, g \qquad (5.17)$$

$$\sum_{m} Z_{mg} \leq \text{Max}_g \qquad \forall \quad g \qquad (5.18)$$

Z_{mg} are non-negative integer variables (5.19)

$$X(lkpg) \geq 0 \qquad \forall \quad l, k, p, g$$

This model contains only Z_{mg} integer variables and constraints (5.14) are not required.

5.2 Solution Methodology

The column generation approach described in Chapter 4 is used to solve the models developed.

Method of Solution for Model-5.1:

At any general iteration, let us define the simplex multipliers corresponding to (5.1) and (5.2) as π_k (k=1,....K) and u_{mg}(m=1,..M; g=1,..G) respectively. Now the pricing scheme for determining the entering variable, if any, is to look for any variable X(lkpg) such that the reduced cost ($\bar{C}$(lkpg) = C(lkpg) - Z(lkpg)) associated is negative.

$$\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \ < \ \pi_k + \sum_{ms} u_{mg}\ a_{ms}(lkpg)\ t_{ms}(kp) \qquad \forall \quad k,p,g$$

or,

$$\sum_{ms} a_{ms}(lkpg)\ [\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)\ ] < \pi_k \qquad \forall \quad k,p,g$$

Defining, $$cc_{ms}(kp) = [\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)\ ] \qquad (5.20)$$

we have

$$\sum_{ms} a_{ms}(lkpg)\ cc_{ms}(kp) \ < \ \pi_k \qquad \forall \quad k,p,g$$

Thus, for a fixed k,p,g the problem is to find the optimal assignment of machines to operations. This can be easily obtained by selecting the machine with minimum $cc_{ms}(kp)$ for each operation. i.e.,

Let $m_s = \underset{m}{\mathrm{Min}}\ cc_{ms}(kp) \qquad \forall \quad s$

Optimal assignment is given by:

$$a_{ms} = \begin{cases} 1 \text{ if } m = m_s \\ 0 \text{ otherwise} \end{cases}$$

Let Z^0 be the cost associated with this production plan. If $Z^0 < \pi_k$, then enter the following (K+MG) column into the basis:

$$[\quad e_k \quad ,0\ ,0\ ,\ \underbrace{(\sum_s a_{1s} t_{1s},\ \sum_s a_{2s} t_{2s},\quad \sum_s a_{Ns} t_{Ns})}_{\text{for somevalue of } g},\ \underset{\substack{\uparrow \\ m \text{ values}}}{0}\ ,0\ ,\quad]$$

where e_k is a unit K-vector with 1 at the k^{th} place.

If for every (kpg) combination the optimal assignment $Z^0 > \pi_k$,check if any of the slack or surplus variables can enter. If no slack or surplus can enter the optimal solution is obtained. Thus, Model-5.1 has been solved.

Algorithm-5.1:

Step-0: Initial basis will consist of all slack and artificial variables.

Step-1: Choose a part k, process plan p, group g and the assignment cost matrix as given in (5.20).

Step-2: Find the minimal cost assignment such that

$$C^1 = \sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) < \pi_k \text{ for all } \pi_k \geq 0 \text{ and } u_{mg} \leq 0.$$

If $C^1 > \pi_k$ for all (kpg) combination go to Step-5.

Step-3: Enter the new column and update the basis.

Step-4: Check for the surplus and slack variables to enter. If

$\pi_k < 0$ introduce surplus variable corresponding to part k.

$u_{mg} > 0$ introduce slack corresponding to machine m in group g.

If any of them can enter go to Step-3 else go to Step-1.

Step-5: Stop.

Method of Solution for Model-5.2:

Let us first consider the relaxed linear problem where Z_{mig} are not restricted to integer values. At any general iteration, let us define the simplex multipliers corresponding to (5.6),(5.7),(5.8) and (5.9) as π_k (k=1,2......K),u_{mg} (m=1,2....M;g=1,2...G),v_g (g=1,2..G) and w_{mg}(m=1,2....M,g=1,2...G) respectively. Thus, for any variable X(lkpg) to enter the basis we have:

$$\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \ < \ \pi_k + \sum_{ms} u_{mg}\ a_{ms}(lkpg)\ t_{ms}(kp) \qquad \forall \quad k,p,g$$

or,

$$\sum_{ms} a_{ms}(lkpg)\ [\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)\ ] < \pi_k \qquad \forall \quad k,p,g$$

Defining, $$cc_{ms}(kp) = [\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)\ ] \qquad (5.21)$$

we have

$$\sum_{ms} a_{ms}(lkpg)\ cc_{ms}(kp) \ < \ \pi_k \qquad \forall \quad k,p,g$$

which is same as developed for Model-5.1. Since X(lkpg) do not appear in constraints (5.8) and (5.9) the simplex multipliers v_g and w_{mg} do not appear in the assignment costs calculated. The additional step in this model is to check if any of the Z_{mig} columns can enter. These columns are explicitly known. At this stage a relaxed linear problem has been solved. Branch and Bound on Z_{mig} variables will give us the optimal solution.

Algorithm-5.2:

Step-0 to 4: Follow Step-0 to Step-4 of Algorithm-5.1.

Step-5 : Check if any of the Z_{mig} column can enter. If yes, then go to Step-3 , else go to Step-6.

Step-6 : If Z_{mig} values are integers then stop, else, branch and bound on Z_{mig}. Add the additional constraints and go to Step-0.

Method of Solution for Model-5.3 and Model-5.4:

The only additional variables introduced in these models are Z_{mg}, which are integers. These variables receive the same treatment as Z_{mig} . Algorithm-5.2 will solve these models as well.

5.3 Examples

In this section the column generation scheme is applied to a sample problem. Let us assume we have two machine groups (g=2). Two types of machines (m=2) with known capacity ($b_{1g} = b_{2g} = 200$) are available in these groups. Two machines of type 1 are in group 1 and two machines of type 2 are in group 2. Two part types (k=2) have to be produced in either of these machine groups. The demand for these parts are predicted as $d_1 = d_2 = 10$. Assume that only one process plan ($P_1 = P_2 = 1$) has been identified for each part. The production cost and time of the plans for each part are given below:

$$k=1, p=1\ , \qquad c_{ms}(11) \overset{m\downarrow \;\; s\rightarrow}{=} \begin{bmatrix}10 & 10\\ 5 & 100\end{bmatrix} \qquad t_{ms}(11) = \begin{bmatrix}10 & 10\\ 5 & 100\end{bmatrix}$$

$$k=2, p=1\ , \qquad c_{ms}(21) = \begin{bmatrix}100 & 100\\ 10 & 10\end{bmatrix} \qquad t_{ms}(21) = \begin{bmatrix}100 & 100\\ 10 & 10\end{bmatrix}$$

The application of Algorithm-5.1 to this example problem is illustrated next.

The method is started with all artificial and slacks in the basis. The initial basic variables column is [$\bar{s}_1, \bar{s}_2, s_3, s_4, s_5, s_6$], the R.H.S column is [10,10,200,0,0,200] and the dual variables are [M,M,0,0,0,0]. i.e., $\pi_1 = M, \pi_2 = M, u_{11} = 0, u_{12} = 0, u_{21} = 0, u_{22} = 0$. We want to generate a column for introducing into the basis. Since the dual variables $u_{mg} = 0$ for all m and g we have: $cc_{ms}(kp) = c_{ms}(kp)$ for k = 1,2. For k = 1, the optimal assignment is assigning machine type 2 to operation 1 and machine type 1 to operation 2 with a total cost = 15. Since 15 < M (a large number) this column can enter the basis. The column for basis entry = P^1= [1,0,10,0,5,0]. The basis and inverse are updated using the standard revised simplex rules. The new basic column is [$\bar{s}_1, \bar{s}_2, s_3, s_4, P^1, s_6$], the R.H.S column is [10,10,200,0,0,200] and the dual variables are [M,M,0,0,-M/5+3,0]. With this set of dual variables, find an entering column. Say for k=1,p=1 and g=1, the assignment costs are:

$$cc_{ms}(11) = \begin{bmatrix}10 & 10\\ 5 & 100\end{bmatrix} + \begin{bmatrix}10*0 & 10*0\\ 5*(M/5+3) & 100*(M/5+3)\end{bmatrix} = \begin{bmatrix}10 & 10\\ M-5 & 20M-200\end{bmatrix}$$

The optimal assignment is obtained by assigning both the operations to machine type 1. The new column for basis entry is $P^2[1,0,20,0,0,0]$.

The column entering, basic column, RHS column and dual variables at each iteration are given below:

<u>Iteration # 1:</u>

Column Entering $P^1[1,0,10,0,5,0]$

Basic Column $[\bar{s}_1,\bar{s}_2,s_3,s_4,P^1,s_6]$

RHS Column $[10,10,200,0,0,200]$

Dual Column $[M,M,0,0,-M/5+3,0]$

<u>Iteration # 2:</u>

Column Entering $P^2[1,0,20,0,0,0]$

Basic Column $[P^2,\bar{s}_2,s_3,s_4,P^1,s_6]$

RHS Column $[10,10,0,0,0,200]$

Dual Column $[20,M,0,0,-1,0]$

<u>Iteration # 3:</u>

Column Entering $P^3[1,0,0,10,0,5]$

Basic Column $[P^2,\bar{s}_2,s_3,P^3,P^1,s_6]$

RHS Column $[10,10,0,0,0,200]$

Dual Column $[20,M,0,-1/2,-1,0]$

<u>Iteration # 4:</u>

Column Entering $P^4[0,1,0,0,20,0]$

Basic Column $[P^2,\bar{s}_2,P^4,P^3,P^1,s_6]$

RHS Column $[10,10,0,0,0,200]$

Dual Column $[M/2,M,1-M/40,3/2-M/20,1-M/20,0]$

<u>Iteration # 5:</u>

Column Entering $P^5[0,1,0,0,0,20]$

Basic Column $[P^2,P^5,P^4,P^3,P^1,s_6]$

RHS Column $[10,10,0,0,0,0]$

Dual Column [10,20,1/2,1/2,0,0]

<u>Iteration # 6:</u>

Column Entering s_3[0,0,1,0,0,0]

Basic Column $[P^2, P^5, s_3, P^3, P^1, s_6]$

RHS Column [10,10,0,0,0,0]

Dual Column [20,20,0,-1/2,-1,0]

No more new columns can be generated and , therefore, the present solution is optimal. This illustrates the application of Model-5.1.

To illustrate the application of Model-5.2 which considers relocation costs of machines consider another sample problem. Let us consider two machine groups(g=2) having two machines of type 1 in Group 1 and two machines of type 2 in Group 2 (m=2, $N_{11} = N_{22} = 2$, $N_{12} = N_{21} = 0$). The physical layout does not permit more than two machines in a group ($Max_1 = Max_2 = 2$). The capacity of each machine type is $b_1 = b_2 = 100$. Two parts have to be produced with the existing facility. The cost of relocating machines from one cell to other is say $50. i.e; $C_{112} = C_{221}$ = 50. Suppose there is only one process plan for each part. The production cost and time are given below:

k=1,p=1, $$c_{ms}(11) = \begin{bmatrix} 10 & 100 \\ 100 & 10 \end{bmatrix} \qquad t_{ms}(11) = \begin{bmatrix} 10 & 100 \\ 100 & 10 \end{bmatrix}$$

k=2,p=1, $$c_{ms}(21) = \begin{bmatrix} 10 & 10 \\ 100 & 10 \end{bmatrix} \qquad t_{ms}(21) = \begin{bmatrix} 10 & 10 \\ 100 & 10 \end{bmatrix}$$

The basic column and RHS column in the optimal solution are:

Basic Column $[Z_{221}, P^3, Z_{112}, P^4, P^6, P^5, s_7, P^7, s_9, s_{10}]$

RHS Column $[1, 0, 1, 10, 10, 0, 0, 0, 1, 1]$

We see that one machine of type 1 has moved to group 2 and one machine of type 2 has moved to group 1. Relocation expense of $100 is incurred due to this transition. The details of each iteration are listed in Table-5.1.

5.4 Summary

Steady and predictable demand is desirable for parts produced in cells. However, with time the part composition in these cells change. This may involve consideration for reallocating parts to these cells, relocation expense on existing machines or additional expense on new machines. A number of situations might exist. For example, the decision maker would like to consider relocation of machines and purchase of new machines or only purchase of new machines and no relocation of machines, etc.. Large scale linear and mixed integer programs were developed to address these problems and reflect the attitude of the decision maker. Efficient solution schemes have been presented. A few examples were solved to illustrate the models and the solution methodology developed.

TABLE 5.1: Computational details for solving Model-5.2.

<u>Iteration # 0:</u>	
Basic Column	$^{*}[\bar{s}_1, \bar{s}_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}]$
RHS Column	[10,10,200,0,0,200,0,0,2,2]
Dual Column	[M,M,0,0,0,0,0,0,0,0]
<u>Iteration # 1:</u>	
Column Entering	$P^1[1,0,10,0,10,0,0,0,0,0]$
Basic Column	$[\bar{s}_1, \bar{s}_2, s_3, s_4, P^1, s_6, s_7, s_8, s_9, s_{10}]$
RHS Column	[10,10,200,0,0,200,0,0,2,2]
Dual Column	[M,M,0,0,-M/10+2,0,0,0,0,0]
<u>Iteration # 2:</u>	
Column Entering	$P^2[1,0,110,0,0,0,0,0,0,0]$
Basic Column	$[\bar{s}_1, \bar{s}_2, P^2, s_4, P^1, s_6, s_7, s_8, s_9, s_{10}]$
RHS Column	[90/11,10,20/11,0,0,200,0,0,2,2]
Dual Column	[M,M,-M/110+1,0,-M/11+1,0,0,0,0,0]
<u>Iteration # 3:</u>	
Column Entering	$P^3[0,1,20,0,0,0,0,0,0,0]$
Basic Column	$[\bar{s}_1, P^3, P^2, s_4, P^1, s_6, s_7, s_8, s_9, s_{10}]$
RHS Column	[10,10,0,0,0,200,0,0,2,2]
Dual Column	[M,2M/11,-M/110+1,0,-M/11+1,0,0,0,0,0]
<u>Iteration # 4:</u>	
Column Entering	$P^4[1,0,0,10,0,10,0,0,0,0]$
Basic Column	$[\bar{s}_1, P^3, P^2, P^4, P^1, s_6, s_7, s_8, s_9, s_{10}]$
RHS Column	[10,10,0,0,0,200,0,0,2,2]
Dual Column	[M,2M/11,-M/110+1,-M/10+2,-M/11+1,0,0,0,0,0]

Iteration # 5:

Column Entering $P^5[1,0,0,0,0,110,0,0,0,0]$

Basic Variables $[\bar{s}_1, P^3, P^2, P^4, P^1, P^5, s_7, s_8, s_9, s_{10}]$

RHS Column $[90/11, 10, 0, 0, 0, 20/11, 0, 0, 2, 2]$

Dual Column $[M, 2M/11, -M/110+1, -M/11+1, -M/11+1, -M/110+1, 0, 0, 0, 0]$

Iteration # 6:

Entering Column $Z_{112}[0,0,100,-100,0,0,-1,1,1,0]$

Basic Column $[\bar{s}_1, P^3, Z_{112}, P^4, P^1, P^5, s_7, s_8, s_9, s_{10}]$

RHS Column $[90/11, 10, 0, 0, 0, 20/11, 0, 0, 2, 2]$

Dual Column $[M, 20M/11-10, -M/11+3/2, -M/11+1, -M/110+1/2, -M/110+1, 0, 0, 0, 0]$

Iteration # 7:

Entering Column $P^6[0,1,10,0,10,0,0,0,0,0]$

Basic Column $[\bar{s}_1, P^3, Z_{112}, P^4, P^6, P^5, s_7, s_8, s_9, s_{10}]$

RHS Column $[90/11, 10, 0, 0, 0, 20/11, 0, 0, 2, 2]$

Dual Column $[M, 20M/11-10, -M/11+3/2, -M/11+1, -M/11+3/2, -M/110+1, 0, 0, 0, 0]$

Iteration # 8:

Entering Column $P^7[0,1.0,10,0,10,0,0,0,0]$

Basic Column $[\bar{s}_1, P^3, Z_{112}, P^4, P^6, P^5, s_7, P^7, s_9, s_{10}]$

RHS Column $[90/11, 10, 0, 0, 0, 20/11, 0, 0, 2, 2]$

Dual Column $[M, M, -M/20+1, -M/11+1, -M/20+1, -M/110+1, 0, -45M/11-150, 0, 0]$

Iteration # 9:

Entering Column $Z_{221}[0,0,0,0,-100,100,1,-1,0,1]$

Basic Column $[Z_{221}, P^3, Z_{112}, P^4, P^6, P^5, s_7, P^7, s_9, s_{10}]$

RHS Column $[1, 0, 1, 10, 10, 0, 0, 0, 1, 1]$

Dual Column $[110/9, 110/9, 7/18, -1/9, 7/18, 8/9, 0, 0, 0, 0]$

Iteration # 10:

Entering Column	s_6[0,0,0,0,0,1,0,0,0,0]
Basic Column	$[Z_{221},s_6,Z_{112},P^4,P^6,P^5,s_7,P^7,s_9,s_{10}]$
RHS Column	[1,0,1,10,10,0,0,0,1,1]
Dual Column	[110,110,-17/2,-9,-11/18,0,0,0,0,0]

* $\bar{s}_1,\bar{s}_2$ corresponds to constraints (5.6), s_3 to s_6 corresponds to constraints (5.7), s_7, s_8 corresponds to constraints (5.8) and s_9, s_{10} corresponds to constraints (5.9).

CHAPTER 6

SELECTION OF PARTS AND MACHINES FOR CELLULARIZATION

The parts selected to be manufactured in the cells have a great impact on the utilization of cells. Also, which cell among the feasible cells the part should be assigned to is critical because it prescribes the overall balance of the CM system. In this chapter four mathematical models are developed to address the issues related to selection of parts and machines for cellularization. Utilization is considered implicitly in the models developed by maximizing the parts produced in the cells.

6.1 Mathematical Formulations

It is assumed, that, a part can be produced through one or more process plans. Each operation in a process plan can be performed on alternate machines. Thus for each process plan we have a number of production plans depending on the machines selected for each operation. This is a very realistic assumption considering the fact that in a Flexible Manufacturing System a machine is capable of performing a variety of operations. It is also assumed that the demand for a part could be split and can be produced in more than one cell. The plans identified to produce the same part in different cells could be different but the part should be completely processed in that cell. With these assumptions the models are presented next.

Model-6.1:

It is assumed that the number of machine groups, machine types and capacity of each machine type in a group is known. The objective is to select a subset of parts to be produced in these machine groups. It is desired to maximize the total demand of the parts produced in these machine groups. The model is stated as follows:

$$\text{Maximize } L61 = \sum_{gkpl} X(lkpg)$$

subject to:

$$\sum_{gpl} X(lkpg) \le d_k \qquad \forall \quad k \qquad (6.1)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpg) \ t_{ms}(kp) \right) X(lkpg) \le b_{mg} \qquad \forall \quad m,g \qquad (6.2)$$

$$\sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg) \ c_{ms}(kp) \right) X(lkpg) \le B \qquad (6.3)$$

$$X(lkpg) \ge 0 \qquad \forall \quad l,k,p,g$$

The objective function maximizes the demand of parts produced in the cells. Constraints (6.1) restrict the quantity of part produced to the demand. Constraints (6.2) ensure the capacity of machine hours available in each group is not violated. Constraint (6.3) restricts the operating cost to the budget.

Model-6.2:

For developing this model, assume the number of cells to be formed is known. Also, the maximum number of machines which could be placed in a cell and the number of machines of each type available are known. The objective of this model is to identify the subset of parts demand to be produced in cells and the machines to be selected for cellularization simultaneously. Accordingly,

$$\text{Maximize} \quad M62 = \sum_{ckpl} X(lkpc)$$

subject to:

$$\sum_{cpl} X(lkpc) \le d_k \qquad \forall \quad k \qquad (6.4)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpc) \;\; t_{ms}(kp) \right) X(lkpc) \le b_m Z_{mc} \qquad \forall \quad m, c \qquad (6.5)$$

$$\sum_{ckpl} \left(\sum_{ms} a_{ms}(lkpc) \;\; c_{ms}(kp) \right) X(lkpc) \le B \qquad (6.6)$$

$$\sum_{m} Z_{mc} \le Max_c \qquad \forall \quad c \qquad (6.7)$$

$$\sum_{c} Z_{mc} \le N_m \qquad \forall \quad m \qquad (6.8)$$

$$Z_{mc} \text{ non-negative integers} \qquad (6.9)$$

$$X(lkpc \ge 0 \qquad \forall \quad l, k, p, c$$

Constraints (6.4) restrict the quantity of part produced to the demand.

Constraints (6.5) ensure that all the machines and the capacity required to process all the parts identified in a cell are available. Constraint (6.6) restricts the operating cost to the budget. The maximum number of machines that can be assigned to a cell are imposed by Constraints (6.7). Constraints (6.8) ensure the machines assigned to the cells do not exceed the available number of machines of each type. Constraints (6.9) indicate the integer variables.

The two models developed might identify only a portion of the demand for a part to be produced in cells. The rest of the demand for the part has to be produced in the functional job shop. This situation is advantageous if the decision maker does not want to depend on the cells for the only source of supply during implementation. However, if he feels that the total demand for a part identified for cell production should be completely produced in one or more cells, we present Models-6.3 and 6.4 corresponding to Models-6.1 and 6.2 respectively. The models can be stated as follows:

Model-6.3:

$$\text{Maximize} \quad M63 = \sum_{k} I_k \quad \text{or} \quad \sum_{gkpl} X(lkpg)$$

subject to:

$$\sum_{gpl} X(lkpg) = d_k I_k \qquad \forall \quad k \qquad (6.10)$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpg) \; t_{ms}(kp) \right) X(lkpg) \le b_{mg} \qquad \forall \quad m,g \qquad (6.11)$$

$$\sum_{gkpl} \left(\sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \right) X(lkpg) \le B \tag{6.12}$$

$$I_k \quad \text{(0-1) variables} \tag{6.13}$$

$$X(lkpg) \ge 0 \qquad \forall \quad l,k,p,g$$

Model-6.4:

$$\text{Maximize} \quad M64 = \sum_k I_k \quad \text{or} \quad \sum_{ckpl} X(lkpc)$$

subject to:

$$\sum_{cpl} X(lkpc) = d_k I_k \qquad \forall \quad k \tag{6.14}$$

$$\sum_{kpl} \left(\sum_{s} a_{ms}(lkpc)\ t_{ms}(kp) \right) X(lkpc) \le b_m Z_{mc} \qquad \forall \quad m,c \tag{6.15}$$

$$\sum_{ckpl} \left(\sum_{ms} a_{ms}(lkpc)\ c_{ms}(kp) \right) X(lkpc) \le B \tag{6.16}$$

$$\sum_{m} Z_{mc} \le Max_c \qquad \forall \quad c \tag{6.17}$$

$$\sum_{c} Z_{mc} \le N_m \qquad \forall \quad m \tag{6.18}$$

$$Z_{mc} \text{ non-negative integer and } I_k \text{ (0-1) variables} \tag{6.19}$$

$$X(lkpc) \ge 0 \qquad \forall \quad l,k,p,c$$

The I_k(0-1) variables in Constraints (6.10) and (6.14) ensure that if a part is selected then the total demand for that part is produced in the cells. The other constraints of Models-6.3 and 6.4 correspond to Models-6.1 and 6.2 respectively.

6.2 Solution Methodology

The column generation approach described in Chapter 4 is used to solve the models developed.

Method of Solution for Model-6.1:

At any general iteration, let us define the simplex multipliers corresponding to (6.1),(6.2) and (6.3)as π_k (k=1,....K), u_{mg}(m=1,..M;g=1,..G) and v respectively. Now the pricing scheme for determining the entering variable, if any, is to look for any variable $X_{(lkpg)}$ such that the reduced cost $C_{(lkpg)}-Z_{(lkpg)}$ associated is positive. Thus $C_{(lkpg)} > Z_{(lkpg)}$

$$1 > \pi_k + \sum_{ms} u_{mg} a_{ms}(lkpg)\ t_{ms}(kp) + v \sum_{ms} a_{ms}(lkpg)\ c_{ms}(kp) \qquad \forall \quad k,p,g$$

or,

$$\sum_{ms} a_{ms}(lkpg)\ [(-v)\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)] > \pi_k - 1 \qquad \forall \quad k,p,g$$

Defining, $cc_{ms}(kp) = [(-v)\ c_{ms}(kp) + (-u_{mg})\ t_{ms}(kp)\ ]$ (6.20)

we have:

$$\sum_{ms} a_{ms}(lkpg)\ cc_{ms}(kp) \ > \ \pi_k - 1 \qquad \forall \quad k,p,g$$

Thus, for a fixed k,p,g the problem is to find the optimal assignment of machines to operations. This can be easily obtained by selecting the machine with minimum $cc_{ms}(kp)$ for each operation. i.e.,

Let $m_s = \underset{m}{\text{Min}}\ cc_{ms}(kp) \qquad \forall \quad s$

Optimal assignment is given by:

$$a_{ms} = \begin{cases} 1 \text{ if } m = m_s \\ 0 \text{ otherwise} \end{cases}$$

Any plan so identified whose value is greater than π_k-1, is a candidate column to enter the basis. The (K+MG+1) values defining the entering column are :

First K values $[0\ 0\ 0\ 0 \qquad \underset{\uparrow k^{th}\ \text{value}}{1}\ 0\ 0]$

Next MG values $[\underset{\uparrow \atop m\ \text{values}}{0,0,0,} \quad \underset{\text{for somevalue of } g}{(\sum_s a_{1s} t_{1s},\ \sum_s a_{2s} t_{2s},\quad \sum_s a_{Ms} t_{Ms})},0,0\ ]$

Last 1 value $[\sum_{ms} a_{ms} c_{ms}]$

If for every (kpg) combination the optimal assignment is $\le \pi_k$-1, check if any of the slack or surplus variables can enter. If no slack or surplus can enter the optimal solution is obtained. Thus, Model-6.1 has been solved.

Algorithm-6.1:

Step-0: Initial basis will consist of all slack variables.

Step-1: Choose a part k, process plan p, group g and the assignment cost matrix as given in (6.20).

Step-2: Find the minimal cost assignment such that

$C^1 = \sum a_{ms}(lkpg)\ c_{ms}(kp) > \pi_k - 1$ for all $\pi_k \ge 0$, $u_{mg} \ge 0$ and $v \ge 0$.

If $C^1 < \pi_k$-1 for all (kpg) combination go to Step-5.

Step-3: Enter the new column and update the basis. Go to Step-4.

Step-4: Check for the surplus and slack variables to enter. If

$\pi_k < 0$ introduce slack corresponding to part k.

$u_{mg} < 0$ introduce slack corresponding to machine m in group g.

$v < 0$ introduce slack corresponding to budget constraint.

If any of them can enter go to Step-3 else go to Step-5.

Step-5: Stop.

<u>Method of Solution for Model-6.2:</u>

Let us first consider the relaxed linear problem where Z_{mc} are not restricted to integer values. At any general iteration, let us define the simplex multipliers corresponding to (6.4), (6.5), (6.6), (6.7) and (6.8) as π_k (k=1,2......K), u_{mc}(m=1,2....M; c=1,2...C), v, v_c (c=1,2..C) and w_m (m=1,2....M) respectively. Thus, for any variable X(lkpc) to enter the basis we have:

$$1 > \pi_k + \sum_{ms} u_{mc} a_{ms}(lkpc)\ t_{ms}(kp) + v \sum_{ms} a_{ms}(lkpc)\ c_{ms}(kp) \qquad \forall \quad k,p,c$$

or,

$$\sum_{ms} a_{ms}(lkpg)\ [(-v)\ c_{ms}(kp) + (-u_{mc})\ t_{ms}(kp)] > \pi_k - 1 \qquad \forall \quad k,p,c$$

Defining, $cc_{ms}(kp) = [(-v)\ c_{ms}(kp) + (-u_{mc})\ t_{ms}(kp)\ ]$ (6.21)

we have:

$$\sum_{ms} a_{ms}(lkpg)\ cc_{ms}(kp) \ > \ \pi_k - 1 \qquad \forall \quad k,p,c$$

which is same as developed for Model-6.1. Since X(lkpc) do not appear in constraints (6.7) and (6.8) the simplex multipliers v_j and w_m do not appear in the assignment costs calculated. The additional step in this model is to check if any of the Z_{mc} column can enter. These columns are explicitly known. At this stage the relaxed linear problem has been solved. A branch and bound on Z_{mc} will solve the model optimally.

Algorithm-6.2:

Step-0 to 4: Follow Step-0 to Step-4 of Algorithm-6.1.

Step-5 : Check if any of the Z_{mc} column can enter. If yes, then go to Step-3 , else go to Step-6.

Step-6 : If Z_{mc} values are integers then stop, else branch and bound on Z_{mc}. Add the additional constraints and go to Step-0.

Method of Solution for Model-6.3 and Model-6.4:

The only additional variables introduced in Model-6.3 and Model-6.4 are I_k. It is suggested that Algorithm 6.1 and 6.2 be used to solve these models respectively by treating I_k as continuous variables. In Step 0 , however, we will start with artificial and slack. An implicit enumeration on I_k will solve the models optimally.

6.3 Examples

Let us assume we have two machines groups (g=2). Three types of machines (m=3) with known capacity ($b_{1g}=b_{2g}=b_{3g}= 100$) are available in these groups. One machine of each type 1 and type 2 are in group 1. In group 2 we have one of type 2 and type 3 each. We have to select the subset of parts to be produced in these machine groups from four different part types. The demand for these parts are $d_1= 10$, $d_2= 40$, $d_3= 60$ and $d_4= 100$. Let us assume only one process plan ($P_1=P_2=P_3=P_4=$

1) for each part type and the budget allocated B = \$ 300. The production cost and time for each part are given below.

$$k = 1,\ p = 1,\quad c_{ms}(11) = t_{ms}(11) = \begin{array}{ccc} & s=1 & s=2 \\ m=1 & 3 & 4 \\ m=2 & 6 & 7 \end{array}$$

$$k = 2,\ p = 1,\quad c_{ms}(21) = t_{ms}(21) = \begin{array}{ccc} & s=1 & s=2 \\ m=1 & 3 & 10 \\ m=2 & 5 & 5 \end{array}$$

$$k = 3,\ p = 1,\quad c_{ms}(31) = t_{ms}(31) = \begin{array}{ccc} & s=1 & s=2 \\ m=1 & 2 & 3 \\ m=3 & 3 & 2 \end{array}$$

$$k = 4,\ p = 1,\quad c_{ms}(41) = t_{ms}(41) = \begin{array}{ccc} & s=1 & s=2 \\ m=2 & 6 & 5 \\ m=3 & 5 & 4 \end{array}$$

The method is started with all slack variables in the basis. The initial basic variables column is $[\ s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}]$. The slacks s_1 to s_4 corresponds to constraints (6.1), s_5 to s_{10} corresponds to constraints (6.2)and s_{11} corresponds to constraint (6.3). The RHS column and dual variables are [10,40,60,100,100,100,0,0,100,100,300] and [0,0,0,0,0,0,0,0,0,0,0] respectively. Since the dual variables are all zero any plan selected has an assignment cost < 1. Say, for k=1,g=1 assign both operations to machine type 1. The column for basic entry is P^1[1,0,0,0,7,0,0,0,0,0,7]. The basis and inverse are updated by the simplex rules. The new basic column is $[P^1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}]$, the RHS column is [10,40,60,100,30,100,0,0,100,100,230] and dual variables are [1,0,0,0,0,0,0,0,0,0,0]. With the new dual variables find the assignment cost and select for each operation, the machine with lowest cost. The column entering, basic column, RHS column and dual variables at each iteration are given in Table 6.1. After iteration # 20, no more new columns can be generated and, therefore, the present solution is optimal. Thus, we have selected 20 units of part 3 , 10 units of part 2 to be produced in machine group 1 and 20 units of part 3 to be produced in machine group 2. The plans identify the machine assignment for each operation in the process plan.

For illustrating an example for Model-6.2 we need additional information on maximum number of machines in a cell and number of cells to be formed. The number of machines available of each type is assumed to be as given before. Let not more than two machines be allowed in machine group. i.e., $Max_1 = Max_2 = 2$ and c=2. The details of each iteration are listed in Table 6.2.

6.4 Summary

A number of companies are switching to cellular manufacturing. The selection of parts and machines is an important problem faced by them during transition. Four mathematical models were developed to address the issues related to this situation. The objective of the models is to maximize the production in cells, with restriction on cell size, available machine time and budget. The presence of alternate process plans has been incorporated in the models developed. Alternate objective of maximizing the total variety of parts produced in cells is also presented. An efficient solution scheme based on column generation approach has been presented and illustrated with examples.

TABLE 6.1: Computational details for solving Model-6.1.

Iteration # 1:	
Column Entering	P^1[1,0,0,0,7,0,0,0,0,0,7]
Basic Column	$[P^1,s_2,s_3,s_4,s_5,s_6,s_7,s_8,s_9,s_{10},s_{11}]$
RHS Column	[10,40,60,100,30,100,0,0,100,100,230]
Dual Column	[1,0,0,0,0,0,0,0,0,0,0]
Iteration # 2:	
Column Entering	P^2[0,1,0,0,13,0,0,0,0,0,13]
Basic Column	$[P^1,s_2,s_3,s_4,P^2,s_6,s_7,s_8,s_9,s_{10},s_{11}]$
RHS Column	[10,490/13,60,100,30/13,100,0,0,100,100,200]
Dual Column	[6/13,0,0,0,1/13,0,0,0,0,0,0]
Iteration # 3:	
Column Entering	P^3[1,0,0,0,0,13,0,0,0,0,13]
Basic Column	$[P^1,s_2,s_3,s_4,P^2,P^3,s_7,s_8,s_9,s_{10},s_{11}]$
RHS Column	[30/13,5670/169,60,100,1090/169,100/13,0,0,100,100,100]
Dual Column	[6/13,0,0,0,1/13,7/169,0,0,0,0,0]
Iteration # 4:	
Column Entering	P^4[0,1,0,0,0,10,0,0,0,0,10]
Basic Column	$[P^1,s_2,s_3,s_4,P^2,P^4,s_7,s_8,s_9,s_{10},s_{11}]$
RHS Column	[10,4680/169,60,100,390/169,10,0,0,100,100,100]
Dual Column	[6/13,0,0,0,1/13,1/10,0,0,0,0,0]
Iteration # 5:	
Column Entering	P^5[0,1,0,0,3,5,0,0,0,0,8]
Basic Column	$[P^1,s_2,s_3,s_4,P^5,P^4,s_7,s_8,s_9,s_{10},s_{11}]$
RHS Column	[10,25,60,100,10,5,0,0,100,100,100]
Dual Column	[-1/6,0,0,0,1/6,1/10,0,0,0,0,0]
Iteration # 6:	

Column Entering $s_1[1,0,0,0,0,0,0,0,0,0,0]$

Basic Column $[P^1, s_2, s_3, s_4, P^5, s_1, s_7, s_8, s_9, s_{10}, s_{11}]$

RHS Column $[40/7, 20, 60, 100, 20, 30/7, 0, 0, 100, 100, 100]$

Dual Column $[0,0,0,0,1/7,4/35,0,0,0,0,0]$

Iteration # 7:

Column Entering $P^6[0,0,0,1,0,0,0,0,6,4,10]$

Basic Column $[P^1, s_2, s_3, s_4, P^5, s_1, s_7, s_8, s_9, s_{10}, P^6]$

RHS Column $[40/7, 20, 60, 90, 20, 30/7, 0, 0, 40, 60, 10]$

Dual Column $[0,0,0,0,3/70,1/70,0,0,0,0,1/10]$

Iteration # 8:

Column Entering $P^7[0,0,1,0,3,0,2,0,0,0,5]$

Basic Column $[P^1, s_2, s_3, s_4, P^5, s_1, P^7, s_8, s_9, s_{10}, P^6]$

RHS Column $[40/7, 20, 60, 90, 20, 30/7, 0, 0, 40, 60, 10]$

Dual Column $[0,0,0,0,3/70,1/70,13/70,0,0,0,1/10]$

Iteration # 9:

Column Entering $P^8[0,0,1,0,5,0,0,0,0,0,5]$

Basic Column $[P^8, s_2, s_3, s_4, P^5, s_1, P^7, s_8, s_9, s_{10}, P^6]$

RHS Column $[8, 20, 52, 90, 20, 10, , 0, 0, 40, 60, 10]$

Dual Column $[0,0,0,0,1/10,-1/50,1/10,0,0,0,1/10]$

Iteration # 10:

Column Entering $s_6[0,0,0,0,0,1,0,0,0,0,0]$

Basic Column $[P^8, s_2, s_3, s_4, P^5, s_1, P^7, s_8, s_6, s_{10}, P^6]$

RHS Column $[16, 100/3, 44, 250/3, 20/3, 10, 0, 0, 200/3, 100/3, 50/3]$

Dual Column $[0,0,0,0,6/50,0,31/50,0,1/30,0,-6/50]$

Iteration # 11:

Column Entering $s_{11}[0,0,0,0,0,0,0,0,0,0,1]$

Basic Column $[P^8, s_2, s_3, s_4, s_{11}, s_1, P^7, s_8, s_6, s_{10}, P^6]$

RHS Column [20,40,40,250/3,100/3,10,0,0,10C,100/3,50/3]

Dual Column [0,0,0,0,1/5,0,7/10,0,1/6,0,0]

Iteration # 12:

Column Entering P^9[1,0,0,0,0,13,0,0,0,0,13]

Basic Column $[P^8, s_2, s_3, s_4, P^9, s_1, P^7, s_8, s_6, s_{10}, P^6]$

RHS Column [20,40,40,250/3,100/39,290/39,0,0,200/3,100/3,50/3]

Dual Column [0,0,0,0,8/65,0,81/130,0,1/26,0,1/13]

Iteration # 13:

Column Entering P^{10}[0,1,0,0,0,10,0,0,0,0,10]

Basic Column $[P^8, s_2, s_3, s_4, P^{10}, s_1, P^7, s_8, s_6, s_{10}, P^6]$

RHS Column [20,110/3,40,250/3,10/3,10,0,0,200/3,100/3,50/3]

Dual Column [0,0,0,0,1/10,0,6/10,0,0,0,1/10]

Iteration # 14:

Column Entering P^{11}[1,0,0,0,0,0,0,7,0,0,7]

Basic Column $[P^8, s_2, s_3, s_4, P^{10}, s_1, P^7, P^{11}, s_6, s_{10}, P^6]$

RHS Column [20,110/3,40,250/3,10/3,10,0,0,200/3,100/3,50/3]

Dual Column [0,0,0,0,1/10,0,6/10,3/70,0,0,1/10]

Iteration # 15:

Column Entering P^{12}[0,1,0,0,0,0,0,3,5,0,8]

Basic Column $[P^8, s_2, s_3, s_4, P^{10}, s_1, P^7, P^{12}, s_6, s_{10}, P^6]$

RHS Column [20,110/3,40,250/3,10/3,10,0,0,200/3,100/3,50/3]

Dual Column [0,0,0,0,1/10,0,6/10,3/45,0,0,1/10]

Iteration # 16:

Column Entering P^{13}[0,0,1,0,0,0,0,0,0,5,5]

Basic Column $[P^8, s_2, s_3, s_4, P^{13}, s_1, P^7, P^{12}, s_6, s_{10}, P^6]$

RHS Column [20,40,100/3,250/3,20/3,10,0,0,100,0,50/3]

Dual Column [0,0,0,0,0,0,1/2,-1/5,-1/6,0,1/5]

Iteration # 17:

Column Entering	s_9[0,0,0,0,0,0,0,0,1,0,0]
Basic Column	[P^8, s_2, s_3, s_4, P^{13}, s_1, P^7, P^{12}, s_6, s_9, P^6]
RHS Column	[20,40,100/3,250/3,20/3,10,0,0,100,0,50/3]
Dual Column	[0,0,0,0,1/6,0,2/3,11/45,0,1/6,1/30]

Iteration # 18:

Column Entering	P^{14}[0,1,0,0,0,10,0,0,0,0,10]
Basic Column	[P^8, s_2, s_3, s_4, P^{13}, s_1, P^7, P^{12}, P^{14}, s_9, P^6]
RHS Column	[20,30,0,100,20,10,0,0,10,100,0]
Dual Column	[0,0,0,0,1/6,1/15,2/3,11/45,0,1/6,1/30]

Iteration # 19:

Column Entering	P^{15}[0,1,0,0,0,0,0,0,10,0,10]
Basic Column	[P^8, s_2, s_3, s_4, P^{13}, s_1, P^7, P^{12}, P^{14}, s_9, P^{15}]
RHS Column	[20,30,20,100,20,10,0,0,10,100,0]
Dual Column	[0,0,0,0,1/10,0,6/10,1/15,0,1/10,1/10]

Iteration # 20:

Column Entering	P^{16}[0,0,1,0,0,0,0,2,0,2,4]
Basic Column	[P^8, s_2, s_3, s_4, P^{13}, s_1, P^7, P^{16}, P^{14}, s_9, P^{15}]
RHS Column	[20,30,20,100,20,10,0,0,10,100,0]
Dual Column	[0,0,0,0,1/10,0,6/10,1/5,0,1/10,1/10]

TABLE 6.2: Computational details for solving Model-6.2.

Iteration # 0:

$[s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

Iteration # 1:

P^1[0,0,1,0,2,0,2,0,0,0,4,0,0,0,0,0]

$[s_1, s_2, s_3, s_4, P^1, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,1/2,0,0,0,0,0,0,0,0,0,0,0]

Iteration # 2:

P^2[0,0,1,0,0,0,5,0,0,0,5,0,0,0,0,0]

$[s_1, s_2, s_3, s_4, P^1, s_6, P^2, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,3/10,0,1/5,0,0,0,0,0,0,0,0,0]

Iteration # 3:

P^3[0,1,0,0,0,10,0,0,0,0,10,0,0,0,0,0]

$[s_1, s_2, s_3, s_4, P^1, P^3, P^2, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,0,0,0,0,0,0,0,0,0]

Iteration # 4:

P^4[0,0,1,0,0,0,0,2,0,2,4,0,0,0,0,0]

$[s_1, s_2, s_3, s_4, P^1, P^3, P^2, P^4, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,1/2,0,0,0,0,0,0,0,0]

Iteration # 5:

P^5[0,0,1,0,0,0,0,0,0,5,5,0,0,0,0,0]

$[s_1,s_2,s_3,s_4,P^1,P^3,P^2,P^4,s_9,P^5\ ,s_{11},s_{12},s_{13},s_{14},s_{15},s_{16}\ ]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,3/10,0,1/5,0,0,0,0,0,0]

Iteration # 6:

P^6[0,1,0,0,0,0,0,0,10,0,10,0,0,0,0,0]

$[s_1,s_2,s_3,s_4,P^1,P^3,P^2,P^4,P^6,P^5\ ,s_{11},s_{12},s_{13},s_{14},s_{15},s_{16}\ ]$

[10,40,60,100,0,0,0,0,0,0,300,2,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,3/10,1/10,1/5,0,0,0,0,0,0]

Iteration # 7:

Z_{31}[0,0,0,0,0,0,-100,0,0,0,0,1,0,0,0,1]

$[s_1,s_2,s_3,s_4,P^1,P^3,P^2,P^4,P^6,P^5\ ,s_{11},s_{12},s_{13},s_{14},s_{15},Z_{31}\ ]$

[10,40,40,100,0,0,20,0,0,0,0,200,1,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,3/10,1/10,1/5,0,0,0,0,0,20]

Iteration # 8:

Z_{21}[0,0,0,0,0,-100,0,0,0,0,0,0,1,0,0,1,0]

$[s_1,s_2,s_3,s_4,P^1,P^3,P^2,P^4,P^6,P^5\ ,s_{11},Z_{21},s_{13},s_{14},s_{15},Z_{31}\ ]$

[10,30,40,100,0,10,20,0,0,0,100,1,2,1,1,1]

[0,0,0,0,3/10,1/10,1/5,3/10,1/10,1/5,0,10,0,0,0,10]

Iteration # 9:

Z_{11}[0,0,0,0,-100,0,0,0,0,0,0,0,1,0,1,0,0]

$[s_1,s_2,s_3,s_4,P^1,P^3,P^2,P^4,P^6,P^5\ ,s_{11},Z_{21},s_{13},Z_{11},s_{15},Z_{31}\ ]$

[10,40,10,100,50,0,0,0,0,0,100,0,2,1,2,1]

[0,0,0,0,3/10,1/10,1/5,3/10,1/10,1/5,0,10,0,20,0,10]

Iteration # 10:

$Z_{32}[0,0,0,0,0,0,0,0,0,-100,0,0,1,0,0,1]$

$[s_1,s_2,s_3,s_4,P^1,P^3,Z_{32},P^4,P^6,P^5,s_{11},Z_{21},s_{13},Z_{11},s_{15},Z_{31}]$

$[10,40,10,100,50,0,0,0,0,0,100,0,2,1,2,1]$

$[0,0,0,0,3/10,1/10,1/5,3/10,1/10,1/5,0,10,0,10,0,20]$

Iteration # 11:

$Z_{22}[0,0,0,0,0,0,0,0,-100,0,0,0,1,0,1,0]$

$[s_1,s_2,s_3,s_4,P^1,P^3,Z_{32},P^4,P^6,P^5,Z_{22},Z_{21},s_{13},Z_{11},s_{15},Z_{31}]$

$[10,30,10,100,50,0,0,0,10,0,1,0,1,1,1,1]$

$[0,0,0,0,2/10,0,1/10,2/10,0,1/10,1/10,0,0,20,0,10]$

s_1 to s_4 corresponds to constraints (6.4), s_5 to s_{10}corresponds to constraints (6.5), s_{11} corresponds to constraint (6.6), s_{12},s_{13} corresponds to constraints (6.7) and s_{14} to s_{16} corresponds to constraints (6.8)

CHAPTER 7

CELL DESIGN WITH MATERIAL HANDLING AND REPLACEMENT CONSIDERATIONS

The creation of independent cells, i.e., cells where parts are completely processed in the cell and no linkages with other cells in the factory exist, is a common goal for cell formation. However, many times it is not economical or practical to achieve cell independence, especially, when under-utilization, load imbalance and higher capital investment are the potential threats to introducing cellular manufacturing. In fact, new technologies and faster deterioration rate of certain machines in cells could render the previously allocated parts to a cell undesirable. Thus there is a need to replace these machines and substitute with new ones. These new machines could be technologically updated or the same.

7.1 Mathematical Formulations

In this chapter, four mathematical models are developed considering the material handling cost and allowing for replacement of machines in cells. The first model addresses the design of new cells considering the investment on machines, operating cost and material handling cost. The objective is to group parts and machines such that all the above costs are minimized. A trade off exists between these costs. With time as new parts are introduced and part mix and volume of parts required change there is a need to consider the allocation of parts to identify natural part families. The second model is developed for this purpose. The objective of this model is to minimize the material handling and operating costs. On a few occasions we would also like to relocate machines to minimize material handling or add new machines to meet the capacity requirements. The third model addresses this issue. In a reassignment of this nature the material handling could impose a severe constraint. Moreover, with time as machines get older the operating costs and times change, forcing a reassignment. The fourth model includes the issue of replacement of machines in one or more cells by new machines. The additional objective of this model is the selection of proper replacement alternative. The column generation approach discussed so far can also be extended to solve these models.

In Chapters 4 to 6 a production plan is defined to be an assignment of machines to operations for a given (kp) or (kpc) combination. However, to account for material handling between cells the production plan definition should contain the information about not only the machine type but also the group to which it belongs. Let a production plan be designated by numbers $a(lkpg)$ for $l \varepsilon L(kpg)$ so that

$$a_{mg's}(lkpg) = \begin{cases} 1 & \text{if in plan } l \text{ machine m in group g' is assigned} \\ & \text{to operation s for (kpg) combination.} \\ 0 & \text{otherwise.} \end{cases}$$

Also let us define the following before presenting the models:

Operating Cost:

$$OC \quad = \quad \sum_{kpglg'} \left(\sum_{ms} a_{mg's}(lkpg) \; c_{ms}(kp) \right) X(lkpg)$$

Material handling cost:

$$MH \quad = \quad \sum_{kpgl} \left(\sum_{g'} cc_{gg'}(k) \sum_{ms} a_{mg's}(lkpg) \right) X(lkpg)$$

$cc_{gg'}(k)$ -Cost of moving part k from group g to g'.

Material handling movement:

$$MV \quad = \quad \sum_{kpgl} \left(\sum_{g'} d_{gg'} \sum_{ms} a_{mg's}(lkpg) \right) X(lkpg)$$

$d_{gg'}$ - Distance between group g to g'.

Interest on Capital:

$$= r \sum_{m} W_{m} + \sum_{m} D_{m}$$

This is for one machine of each type. It can be extended to account for more than one machine of each type.

The cost $cc_{gg'}(k)$ denotes the cost of moving part type k from cell g to cell g' . It is assumed from a GT point of view that a part whenever it visits a cell other than the parent cell it has been allocated to it is always returned to the parent cell for storage. It is routed to other cells whenever the required machine is free.

The models considering the material handling are given next.

Model-7.1:

The objective of this model is to group parts such that the total cost consisting of investment on machines, operating cost and material handling cost is minimized. A trade off exists between these costs. The complete model is stated as follows:

(MIP-1)

$$\text{Minimize M71} = \sum_{mg} C_{mg}\ Z_{mg} + OC + MH$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall\ k \qquad (7.1)$$

$$\sum_{kpl}\left(\sum_{gs} a_{mg's}(lkpg)\ t_{ms}(kp)\right) X(lkpg) \leq b_m\ Z_{mg'}, \qquad \forall\ m,g' \qquad (7.2)$$

$$\sum_{n} Z_{mg} \leq Max_g \qquad \forall\ g \qquad (7.3)$$

$$Z_{mc} \text{ non-negative integer variables} \qquad (7.4)$$

$$X(lkpg) \geq 0 \qquad \forall\ l,k,p,g$$

The Constraints (7.1) guarantee that the lower bound on demand for all parts is met. Constraints (7.2) ensure that the capacity of each machine type selected is not violated. The maximum number of machines which can be allocated to each group are restricted by Constraints (7.3). Constraints (7.4) indicate the integer variables.

Model-7.2:

In any manufacturing system with time new parts are introduced and part mix and volume of parts require change. In a situation like this there is a need to consider the allocation of parts to known machine groups to identify natural part families. This is a sequential approach to cell design where the machine groups were determined first. The model presented here addresses this issue. The objective of this model is to minimize the material handling and operating costs. The model is presented next.

(MIP-2)

Minimize M72 = MH + OC

subject to:

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \quad k \qquad (7.5)$$

$$\sum_{kpl} \sum_{g} \left(\sum_{s} a_{mg's}(lkpg) \; t_{ms}(kp) \right) X(lkpg) \leq b_{mg'} \qquad \forall \quad m, g' \quad (7.6)$$

$$\sum_{kpgl} \left(\sum_{g'} d_{gg'} \sum_{ms} a_{mg's}(lkpg) \right) X(lkpg) \leq D \qquad (7.7)$$

$$X(lkpg) \geq 0 \qquad \forall \quad l, k, p, g$$

Constraints (7.5) guarantee the demand for parts to be met. Constraints (7.6) ensure that the capacity of each machine type in a group is not violated. The upper limit on material handling capacity is imposed by Constraint (7.7).

<u>Model-7.3:</u>

On a number of occasions material handling capacity might pose a severe constraint. One possible way to minimize the inter-cell movement is to relocate machines. Moreover, material handling is a recurring expense and it might be worth avoiding if the parts mix is expected to be stable for the planning horizon. This model is developed to form part families and machine groups by trying to minimize the material handling cost and relocation expense in addition to operating cost and investment on additional capacity. It is assumed that the capacity of current material handling system is known.

(MIP-3)

$$\text{Minimize } M73 = \sum_{mlg} C_{mlg} \, Z_{mlg} + \sum_{mg} C_{m} \, Z_{mg} + OC + MH$$

subject to:

$$\sum_{gpl} X(lkpg) \geq d_{k} \qquad \forall \quad k \qquad (7.8)$$

$$\sum_{kpl} \left(\sum_{gs} a_{mg's}(lkpg) \, t_{ms}(kp) \right) X(lkpg) \leq b_{m} \left(N_{mg'} + \sum_{l} Z_{mlg'} - \sum_{l} Z_{mg'l} + Z_{mg'} \right)$$
$$\forall \quad m, g' \qquad (7.9)$$

$$\sum_{kpgl} \left(\sum_{g'} d_{gg'} \sum_{ms} a_{mg's}(lkog) \right) X(lkpg) \leq D \qquad (7.10)$$

$$\sum_{m} N_{mg} + \sum_{ml} Z_{mlg} - \sum_{ml} Z_{mgl} + \sum_{m} Z_{mg} \leq Max_{g} \qquad \forall \quad g \qquad (7.11)$$

$$\sum_{l} Z_{mgl} - \sum_{l} Z_{mlg} \leq N_{mg} \qquad \forall \quad m, g \qquad (7.12)$$

$$Z_{mlg}, \; Z_{mg} \text{ non-negative integer variables.} \qquad (7.13)$$

$$X(lkpg) \geq 0 \qquad \forall \quad l, k, p, g$$

Constraints (7.8) force the demand for parts to be met. Constraints (7.9) ensure that all the machines required to process the parts are available. An upper limit on material handling system is imposed by Constraint (7.10). The maximum number of machines that can be in a machine group are imposed by Constraints (7.11). Constraints (7.12) ensure that the machine relocated to other cells do not exceed the number of machines of each type available in that cell. The integer variables are indicated by Constraints (7.13).

Model-7.4:

New technology and faster deterioration of certain machines could render a previously desirable part assignment undesirable. There is need to identify these machines and replace them with new or technologically updated machines. The following model addresses this issue. For developing this model it is assumed that j = 1 to M' denote the old machine types and m' = M'+1 to M identify the new machine types. Also let m = 1 to M denote all machine types.

(MIP-4)

$$\text{Minimize } M74 = OC + MH + \Big(\sum_{jg} (N_{jg} - R_{jg}) D_j + \sum_{m'g} Z_{m'g} D_{m'}$$

$$+ r \sum_{jg} (N_{jg} - R_{jg}) W_j + r \sum_{m'g} Z_{m'g} W_{m'} \Big)$$

subject to :

$$\sum_{gpl} X(lkpg) \geq d_k \qquad \forall \; k \qquad (7.14)$$

$$\sum_{kpl} \Big(\sum_{gs} a_{jg's}(lkpg) \; t_{js}(kp) \Big) X(lkpg) \leq b_j (N_{jg'} - R_{jg'}) \qquad \forall \; j, g' \qquad (7.15)$$

$$\sum_{kpl} \Big(\sum_{gs} a_{m'g's}(lkpg) \; t_{m's}(kp) \Big) X(lkpg) \leq b_{m'} Z_{m'g'} \qquad \forall \; m', g' \qquad (7.16)$$

$$\sum_{kpgl} \Big(\sum_{g'} d_{gg'} \sum_{ms} a_{mg'u}(lkpg) \Big) X(lkpg) \leq D \qquad (7.17)$$

$$\sum_{j} (N_{jg} - R_{jg}) + \sum_{m'} Z_{m',g} \le Max_{g} \qquad \forall \; g \qquad (7.18)$$

$$R_{jg} \le N_{jg} \qquad \forall \; j,g \qquad (7.19)$$

$$R_{jg} \; , \; Z_{m',g} \text{ non-negative integer variables} \qquad (7.20)$$

The objective function of this model minimizes the loss in interest on capital, material handling and operating costs. Constraints (7.14) to (7.18) are similar to Constraints (7.8) to (7.11). Constraints (7.9) are presented as two sets of constraints (7.15) and (7.16). Constraints (7.19) ensure that the total number of machines removed from a group does not exceed the available number of machines. The integer restrictions are shown by Constraints (7.20).

7.2 Solution Methodology

The column generation approach described in Chapter 4 is used to solve the models developed.

Method of Solution for Model-7.1:

At any general iteration, let us define the simplex multipliers corresponding to (7.1) and (7.2) as π_k (k=1,....K) and $u_{mg'}$, (m=1,..M;g'=1,..G) respectively. Now the pricing scheme for determining the entering variable, if any, is to look for any variable $X(lkpg)$ such that the reduced cost ($\bar{C}(lkpg) = C(lkpg) - Z(lkpg)$) associated is negative.

$$\sum_{ms} a_{mg's}(lkpg)\ c_{ms}(kp) + \sum_{g'ms} a_{mg's}(lkpg)\ cc_{gg'}(k)$$

$$< \pi_k + \sum_{g'ms} u_{mg'} a_{mg's}(lkpg)\ t_{ms}(kp) \qquad \forall \quad k,p,g$$

or,

$$\sum_{g'ms} a_{mg's}(lkpg)\ [\ c_{ms}(kp) + cc_{gg'}(k) + (-u_{mg'})\ t_{ms}(kp)\ ] < \pi_k$$

Defining,

$$cc_{mg's}(kp) = [\ c_{ms}(kp) + cc_{gg'}(k) + (-u_{mg'})\ t_{ms}(kp)\ ] \qquad (7.21)$$

we have

$$\sum_{g'ms} a_{mg's}(lkpg)\ cc_{mg's}(kp) \ < \ \pi_k \qquad \forall \quad k,p,g$$

Thus, for a fixed k,p,g consider the following assignment problem of assigning machines to operations.

Define 0-1 variables $a_{mg's}$ as

$$a_{mg's} = \begin{cases} 1 \text{ if machine m in group g' is assigned to perform} \\ \quad \text{operation s} \\ 0 \text{ otherwise} \end{cases} \qquad \forall \quad m,s$$

Let $cc_{mg's}(kp)$ = cost of assigning machine m in group g' to perform operation s.

The problem is:

Minimize $Z = \sum_{smg'} cc_{mg's}(kp)\ a_{mg's}$

subject to:

$$\sum_{mg'} a_{mg's} \ = 1 \qquad \forall \quad s$$

$$a_{mg's} \ = 0 \text{ or } 1 \qquad \forall \quad m,s,g'$$

The optimal solution to this problem can be obtained by the following simple procedure:

Let $m_s = \underset{mg'}{\mathrm{Min}}\ cc_{mg',s}(kp) \qquad \forall \quad s$

Optimal assignment is given by:

$$a_{mg',s} = \begin{cases} 1 \text{ if } m = m_s \\ 0 \text{ otherwise} \end{cases}$$

Let Z^0 be the cost associated with this production plan. If $Z^0 < \pi_k$, then enter the following (K+MG+G) column into the basis:

The K+MG+G values are:

$$[\quad e_k \quad , \quad ee_{g'} \quad , \quad eee_g \quad]$$

where,

e_k is a unit K-vector with 1 at the k^{th} place.

$ee_{g'} = [(\sum_s a_{1g',s} t_{1s} \ , \ \sum_s a_{2g',s} t_{2s} \ , \quad \sum_s a_{Mg',s} t_{Ms}), \ ]$

eee_g is a vector with G '0' values.

If for every (kpg) combination the optimal assignment $Z^0 > \pi_k$,check if any of the slack or surplus variables can enter. If no slack or surplus can enter the optimal solution is obtained. Thus, Model-6.1 has been solved. Branch and Bound on Z_{mg} gives an optimal solution to the model.

<u>Algorithm-7.1:</u>

Step-0: Initial basis will consist of all slack and artificial variables.

Step-1: Choose a part k, process plan p, group g and the assignment cost matrix as given in (7.21).

Step-2: Find the minimal cost assignment such that

$C^1 = \sum_{g',ms} a_{mg',s}(lkpg)\ cc_{mg',s}(kp) < \pi_k$ for all $\pi_k \geq 0$ and $u_{mg} \leq 0$.

If $C^1 > \pi_k$ for all (kpg) combination go to Step-5.

Step-3: Enter the new column and update the basis.

Step-4: Check for the surplus and slack variables to enter. If

$\pi_k < 0$ introduce surplus variable corresponding to part k.

$u_{mg} > 0$ introduce slack corresponding to machine m in group g.

If any of them can enter go to Step-3 else go to Step-5.

Step-5: Check if any Z_{mg} column can enter. If, yes then go to Step-3 else go to Step-6.

Step-6: If Z_{mg} values are integers then stop else branch and bound on Z_{mg}. Add the additional constraints and go to Step-1.

Method of Solution for Model-7.2 and Model-7.3:

The assignment cost for the entering column for this model is

$$cc_{mg',s}(kp) = [\ c_{ms}(kp) + cc_{gg'}(k) + (-u_{mg'})\ t_{ms}(kp) + (-v)\ d_{gg'}]$$

and the entering column should satisfy the condition:

$$\sum_{g',ms} a_{mg',s}(lkpg)\ cc_{mg',s}(kp) < \pi_k \qquad \forall \quad k,p,g$$

π_k , u_{mg}, and v are simplex multipliers corresponding to Constraints (7.5) to (7.7) in Model-7.2 and (7.8) to (7.10) in Model-7.3 respectively. The number of column values entering in Model-7.2 and 7.3 is (K+MG+1) and (K+MG+1+G+MG) respectively.

Method of Solution for Model-7.4:

The assignment cost $cc_{mg',s}(kp)$ for the entering column for this model is

$$=[(c_{ms}(kp) + cc_{gg'}(k)) + (-u_{jg'})t_{js}(kp) + (-u_{m',g'})t_{m',s}(kp) + (-v)\ d_{gg'}]$$

and the entering column should satisfy the condition:

$$\sum_{g',ms} a_{mg',s}(lkpg)\ cc_{mg',s}(kp) < \pi_k \qquad \forall \quad k,p,g$$

π_k , $u_{jg'}$, $u_{m'g}$, and v are simplex multipliers corresponding to Constraints (7.14), (7.15), (7.16) and (7.17) respectively. The number of column values entering is (K+M'G+(M-M')G+1+G+MG).

7.3 Example

For the purpose of exposition of the column generation scheme, let us consider Model-7.1. The objective of this model is to identify part families and machine groups such the total cost of investment, operating and material handling is minimized. Assume that we have to form two cells (G=2). The available floor space restricts the maximum number of machines to four and two in the two cells respectively (i.e., Max_1=4, Max_2=2). We have six parts to be manufactured on four available machine types. The demand for the parts is : $d_1=d_3=d_5= 20$, $d_2=d_4=d_6= 10$. The cost of assigning a machine to a cell is known. This cost can be different for a particular machine type to different cells, based on factors such as equipping the cell with foundation, accessories etc. Let us assume the following costs: $C_{11}=C_{12}=C_{42}=200$, $C_{21}=C_{31}= C_{22}=250$, $C_{41}=C_{42}=350$. The inter-cell movement cost for each part type are: $cc_{gg'}(1) = cc_{gg'}(2) = 3$, $cc_{gg'}(3) = cc_{gg'}(5) = cc_{gg'}(6) = 2$, $cc_{gg'}(4) = 1$ for all $g \neq g'$. Within a cell the material handling cost is taken to be zero. The production cost and time data for all parts are given in the Table-7.1. This problem has 16 constraints and 8 integer variables. The number of explicit columns in the model is 28 columns corresponding to the plans and 8 columns corresponding to the machine variables. All these columns need not be explicitly listed, instead they are obtained by solving assignment problems. The procedure of generating the columns for this problem is explained next.

The method is started with all artificial and slacks in the basis. The initial basic variables column is $[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}]$, the right hand side column is [20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2], and the dual variables are [M,M,M,M,M,M,0,0,0,0,0,0,0,0,0,0]. M in this context is a very large number. We can take any part, say k=1,p=1,g=1 and find the assignment cost $cc_{mg's}(11)$, which is given by:

$cc_{mg's}(11)$		s=1	s=2
g'=1	m=1	3	7
	m=3	6	2
g'=2	m=1	6	10
	m=3	9	5

For each operation, we pick up the machine with minimum cost. In this case we pick machine 1 in cell 1 for operation 1 and machine 3 in cell 1 for operation 2 (material handling cost of 3 was included for operations performed in cell 2, because it was assumed that the part is allocated to cell 1). Thus we have a plan with a cost of 5. This plan does not require any material handling because both operations are performed in cell 1. Since this cost is less than M (a large value) it qualifies to enter the basis. The plan column entering the basis is P^1 [1,0,0,0,0,0,3,0,2,0,0,0,0,0,0,0]. The basis and the inverse are updated by the usual simplex rules. The column entering , basic column, RHS column and dual variables for the complete problem are given in Table-7.2.

The optimal solution to the sample problem identifies that we require two machines of each type 1 and 3 in cell 1 and one each of type 2 and 4 in cell 2. Parts 1,3 and 5 are assigned to cell 1, while parts 2,4 and 6 are assigned to cell 2. The assignment of operations to machines in the plan selected are shown in Table-7.3. The numbers in the table indicate the operation time for an operation on the machine selected. Since the number of machines of each type selected are already integers, we did not have to branch and bound on integer variables.

7.4 Summary

Material handling device is a part of automated manufacturing systems. This could pose a severe constraint to assignment of machining parts. For example, if the workload exceeds the capacity of the transfer system, congestions and high in-process inventory are likely to occur. Therefore there is a need to consider the cost and restriction imposed by these systems in the design of cells. Also, certain machines deteriorate faster than others. These machines should be replaced for better performance. Mathematical models were developed to address these issues. The solution methodology developed in Chapter 4 was extended to these models for efficient solution. An illustrative example was solved to demonstrate the solution scheme.

TABLE 7.1: Production cost and time data for parts.

$c_{ms}(kp)=t_{ms}(kp)$		m=1	m=2	m=3	m=4
k=1	s=1	3		6	
	s=2	7		2	
k=2	s=1		8		3
	s=2		2		5
k=3	s=1	2			8
	=2	9		5	
	s=3				3
k=4	s=1		6	10	
	s=2			4	8
	s=3		8		5
k=5	s=1	5		7	
	s=2	7		1	
k=6	s=1		4		1
	s=2		2		9

* a blank space indicates that the operation cannot be performed on the machine i.e. an infinite cost is associated with it.

TABLE 7.2: Computational details for solving Model-7.1.

Iteration # 0:

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\ ]$

[20, 10, 20, 10, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2]

[M, M, M, M, M, M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Iteration # 1:

$P^1\ [1,0,0,0,0,0,3,0,2,0,0,0,0,0,0,0]^1 \qquad [5^*, 0^{**}]$

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\ ]^2$

$[20, 10, 20, 10, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2]^3$

$[M, M, M, M, M, M, -M/3+5/3, 0, 0, 0, 0, 0, 0, 0, 0, 0]^4$

Iteration # 2:

$P^2\ [1,0,0,0,0,0,0,0,8,0,0,0,0,0,0,0]\ [8,0]$

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^2, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\ ]$

[20, 10, 20, 10, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2]

[M, M, M, M, M, M, -M/4+1, 0, -M/8+1, 0, 0, 0, 0, 0, 0, 0]

Iteration # 3:

$P^3\ [1,0,0,0,0,0,0,0,0,0,3,0,2,0,0,0]\ [5,6]$

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^2, s_{10}, P^3, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}\ ]$

[20, 10, 20, 10, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2]

[M, M, M, M, M, M, -M/4+1, 0, -M/8+1, 0, -M/3+11/3, 0, 0, 0, 0, 0]

Iteration # 4:

$P^4\ [1,0,0,0,0,0,0,0,0,0,0,0,8,0,0,0]\ [8,6]$

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^2, s_{10}, P^3, s_{12}, P^4, s_{14}, s_{15}, s_{16}\ ]$

[20, 10, 20, 10, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 4, 2]

[M, M, M, M, M, M, -M/4+1, 0, -M/8+1, 0, -M/4+5/2, 0, -M/8+7/4, 0, 0, 0]

<u>Iteration # 5:</u>

P^5 [1,0,0,0,0,0,0,0,2,0,3,0,0,0,0,0] [5,3]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^5, s_{10}, P^3, s_{12}, P^4, s_{14}, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,0,-M/8+1/4,0,-M/4+5/2,0,-M/8+7/4,0,0,0]

<u>Iteration # 6:</u>

P^6 [0,1,0,0,0,0,0,0,0,0,0,2,0,3,0,0] [5,0]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^5, s_{10}, P^3, P^6, P^4, s_{14}, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,0,-M/8+1/4,0,-M/4+5/2,-M/2+5/2,-M/8+7/4,0,0,0]

<u>Iteration # 7:</u>

P^7 [0,1,0,0,0,0,0,0,0,0,0,0,0,8,0,0] [8,0]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, s_8, P^5, s_{10}, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,0,-M/8+1/4,0,-M/4+5/2,-5M/16+1,-M/8+7/4,-M/8 +1,0,0]

<u>Iteration # 8:</u>

P^8 [0,1,0,0,0,0,0,2,0,3,0,0,0,0,0,0] [5,6]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, P^8, P^5, s_{10}, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,-M/2+11/2,-M/8+1/4,0,-M/4+5/2,-5M/16+1,-M/8+ 7/4,-M/8+1,0,0]

Iteration # 9:

P^9 [0,1,0,0,0,0,0,0,0,8,0,0,0,0,0,0] [8,6]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, P^8, P^5, P^9, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,-5M/16+23/8,-M/8+1/4,-M/8+7/4,-M/4+5/2,-5M/1

6+1,-M/8+7/4,-M/8+1,0,0]

Iteration # 10:

P^{10}[0,1,0,0,0,0,0,0,0,5,0,0,0,3,0,0] [8,3]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, P^8, P^5, P^{10}, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,-5M/16+31/10,-M/3+1/4,-M/8+8/5,-M/4+5/2,-5M/1

6+1,-M/8+7/4,-M/8+1,0,0]

Iteration # 11:

P^{11}[0,1,0,0,0,0,0, ,0,0,0,0,0,3,0,0] [5,3]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, P^{11}, P^5, P^{10}, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-M/4+3/2,-5M/16+5/2,-M/8+1/4,-M/8+8/5,-M/4+5/2,-5M/1

6+1,-M/8+7/4,-M/8+1,0,0]

Iteration # 12:

Z_{11}[0,0,0,0,0,0,-100,0,0,0,0,0,0,0,1,0]

$[\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5, \bar{s}_6, P^1, P^{11}, Z_{11}, P^{10}, P^3, P^6, P^4, P^7, s_{15}, s_{16}]$

[20,10,20,10,20,10,0,0,0,0,0,0,0,0,4,2]

[M,M,M,M,M,M,-2,-5M/16+5/2,-M/2+11/2,-M/8+8/5,-M/4+5/2,-5M/16+1,

-M/8+7/4,-M/8+1,0,0]

Iteration # 13:

Z_{31}[0,0,0,0,0,0,0,0,-100,0,0,0,0,0,1,0]

$[Z_{31},\bar{s}_2,\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},P^{10},P^3,P^6,P^4,P^7,s_{15},s_{16}]$

[2/5,10,20,10,20,10,20,0,3/5,0,0,0,0,0,3,2]

[16,M,M,M,M,M,-2,-5M/16+5/2,-5/2,-M/8+8/5,-3/2,-5M/16+1,-1/4,

-M/8+1,0,0]

Iteration # 14:

Z_{22}[0,0,0,0,0,0,0,0,0,0,0,-100,0,0,0,1]

$[Z_{31},\bar{s}_2,\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},P^{10},P^3,P^6,P^4,Z_{22},s_{15},s_{16}]$

[2/5,10,20,10,20,10,20,0,3/5,0,0,0,0,0,3,2]

[16,M,M,M,M,M,-2,-1,-5/2,1/5,-3/2,-5/2,-1/4,-M/8+10/3,0,0]

Iteration # 15:

s_{10}[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]

$[Z_{31},\bar{s}_2,\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^3,P^6,P^4,Z_{22},s_{15},s_{16}]$

[2/5,10,20,10,20,10,20,0,3/5,0,0,0,0,0,3,2]

[16,M,M,M,M,M,-2,-1,-5/2,0,-3/2,-5/2,-1/4,-M/8+10/3,0,0]

Iteration # 16:

Z_{42}[0,0,0,0,0,0,0,0,0,0,0,0,0,-150,0,1]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^3,P^6,P^4,Z_{22},s_{15},s_{16}]$

[2/5,1/5,20,10,20,10,20,0,3/5,0,0,10,0,1/5,3,8/5]

[16,14,M,M,M,M,-2,-1,-5/2,0,-3/2,-5/2,-1/4,-4/3,0,0]

Iteration # 17:

P^{12}[0,0,0,0,1,0,0,0,0,0,0,0,9,0,0,0][9,4]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^3,P^6,P^{12},Z_{22},s_{15},s_{16}]$

[2/5,1/5,20,10,20,10,20,0,3/5,0,0,10,0,1/5,3,8/5]

[16,14,M,M,M,M,-2,-1,-5/2,0,2M/27-71/27,-5/2,-M/9+13/9,-4/3,0,0]

Iteration # 18:

s_{11}[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},s_{10},s_{11},P^6,P^{12},Z_{22},s_{15},s_{16}]$

[2/5,1/5,20,10,20,10,20,0,3/5,0,0,10,0,1/5,3,8/5]

[16,14,M,M,M,M,-2,-1,-5/2,0,0,-5/2,-M/9+13/9,-4/3,0,0]

Iteration # 19:

P^{13}[0,0,0,0,1,0,0,0,1,0,5,0,0,0,0,0][6,2]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,\bar{s}_5,\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^{13},P^6,P^{12},Z_{22},s_{15},s_{16}]$

[2/5,1/5,20,10,20,10,20,0,3/5,0,0,10,0,1/5,3,8/5]

[16,14,M,M,M,M,-2,-1,-5/2,0,-M/5+21/10,-5/2,-M/9+13/9,-4/3,0,0]

Iteration # 20:

P^{14}[0,0,0,0,1,0,5,0,1,0,0,0,0,0,0,0][6,0]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,P^{14},\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^{13},P^6,P^{12},Z_{22},s_{15},s_{16}]$

[3/5,1/5,20,10,20,10,20,0,8/5,0,0,10,0,1/5,9/5,8/5]

[16,14,M,M,37/2,M,-2,-1,-5/2,0,-8/5,-5/2,-11/18,-4/3,0,0]

Iteration # 21:

P^{15}[0,0,0,0,1,0,0,0,1,0,0,0,8,0,0,0][9,2]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,P^{14},\bar{s}_6,P^1,P^{11},Z_{11},s_{10},P^{13},P^6,P^{15},Z_{22},s_{15},s_{16}]$

[3/5,1/5,20,10,20,10,20,0,8/5,0,0,10,0,1/5,9/5,8/5]

[16,14,M,M,37/2,M,-2,-1,-5/2,0,-8/5,-5/2,-5/8,-4/3,0,0]

Iteration # 22:

P^{16}[0,0,0,0,0,1,0,2,0,0,0,0,0,1,0,0][3,2]

$[Z_{31},Z_{42},\bar{s}_3,\bar{s}_4,P^{14},\bar{s}_6,P^1,P^{16},Z_{11},s_{10},P^{13},P^6,P^{15},Z_{22},s_{15},s_{16}]$

[3/5,1/5,20,10,20,10,20,0,8/5,0,0,10,0,1/5,9/5,8/5]

[16,14,M,M,37/2,M,-2,-M/2+19/6,-5/2,0,-8/5,-5/2,-5/8,-4/3,0,0]

Iteration # 23:

P^{17}[0,0,0,0,0,1,0,0,0,0,0,2,0,1,0,0][3,0]

$[Z_{31}, Z_{42}, \bar{s}_3, \bar{s}_4, P^{14}, P^{17}, P^{1}, P^{16}, Z_{11}, s_{10}, P^{13}, P^{6}, P^{15}, Z_{22}, s_{15}, s_{16}]$

[3/5,4/15,20,10,20,10,20,0,8/5,0,0,10,0,2/5,9/5,4/3]

[16,14,M,M,37/2,28/3,-2,-3/2,-5/2,0,-8/5,-5/2,-5/8,-4/3,0,0]

Iteration # 24:

P^{18}[0,0,1,0,0,0,2,0,0,3,0,0,5,0,0,0][10,2]

$[Z_{31}, Z_{42}, \bar{s}_3, \bar{s}_4, P^{14}, P^{17}, P^{1}, P^{16}, Z_{11}, s_{10}, P^{13}, P^{6}, P^{18}, Z_{22}, s_{15}, s_{16}]$

[3/5,4/15,20,10,20,10,20,0,8/5,0,0,10,0,2/5,9/5,4/3]

[16,14,M,M,37/2,28/3,-2,-3/2,-5/2,0,-8/5,-5/2,-M/5+16/5,-4/3,0,0]

Iteration # 25:

P^{19}[0,0,1,0,0,0,2,0,5,3,0,0,0,0,0,0][10,0]

$[Z_{31}, Z_{42}, \bar{s}_3, \bar{s}_4, P^{14}, P^{17}, P^{1}, P^{16}, Z_{11}, P^{19}, P^{13}, P^{6}, P^{18}, Z_{22}, s_{15}, s_{16}]$

[3/5,4/15,20,10,20,10,20,0,8/5,0,0,10,0,2/5,9/5,4/3]

[16,14,M,M,37/2,28/3,-2,-3/2,-5/2,-M/3+53/6,-8/5,-5/2,-M/5+16/5,-4/3,0,0]

Iteration # 26:

P^{20}[0,0,1,0,0,0,2,0,5,0,0,0,0,3,0,0][10,2]

$[Z_{31}, Z_{42}, P^{20}, \bar{s}_4, P^{14}, P^{17}, P^{1}, P^{16}, Z_{11}, P^{19}, P^{13}, P^{6}, P^{18}, Z_{22}, s_{15}, s_{16}]$

[8/5,2/3,20,10,20,10,20,0,2,0,0,10,0,2/5,2/5,14/15]

[16,14,65/2,M,37/2,28/3,-2,-3/2,-5/2,-2,-8/5,-5/2,-33/10,-4/3,0,0]

Iteration # 27:

P^{21}[0,0,0,1,0,0,0,6,4,0,0,0,0,5,0,0][15,2]

$[Z_{31}, Z_{42}, P^{20}, \bar{s}_4, P^{14}, P^{17}, P^{1}, P^{21}, Z_{11}, P^{19}, P^{13}, P^{6}, P^{18}, Z_{22}, s_{15}, s_{16}]$

[8/5,2/3,20,10,20,10,20,0,2,0,0,10,0,2/5,2/5,14/15]

[16,14,65/2,M,37/2,28/3,-2,-M/6+101/18,-5/2,-2,-8/5,-5/2,-33/10,-4/3,0,0]

Iteration # 28:

P^{22}[0,0,0,1,0,0,0,0,4,0,0,6,0,5,0,0][15,1]

[Z_{31},Z_{42},P^{20},P^{22},P^{14},P^{17},P^{1},P^{21},Z_{11},P^{19},P^{13},P^{6},P^{18},Z_{22},s_{15},s_{16}]

[2,1,20,10,20,10,20,0,2,0,0,10,0,1,0,0]

[16,14,65/2,143/3,37/2,28/3,-2,-5/2,-5/2,-2,-8/5,-5/2,-33/10,-4/3,0,0]

Iteration # 29:

P^{23}[0,0,1,0,0,0,0,0,5,0,2,0,0,3,0,0][10,2]

[Z_{31},Z_{42},P^{20},P^{22},P^{14},P^{17},P^{1},P^{21},Z_{11},P^{19},P^{23},P^{6},P^{18},Z_{22},s_{15},s_{16}]

[2,1,20,10,20,10,20,0,2,0,0,10,0,1,0,0]

[16,14,65/2,143/3,37/2,28/3,-2,-5/2,-5/2,-2,-2,-5/2,-33/10,-4/3,0,0]

1 Column Entering
2 Basic Column
3 RHS Column
4 Dual Column
* operation cost
** material handling or inter-cell cost

$\bar{s}_1$ to $\bar{s}_6$ corresponds to constraints (7.1), s_7 to s_{14} corresponds to constraints (7.2) and s_{15}, s_{16} corresponds to constraints (7.3).

TABLE 7.3: Assignment of operations to machines in the optimal plans selected.

$t_{ms}(kp)$		g=1				g=2			
		m=1	m=2	m=3	m=4	m=1	m=2	m=3	m=4
k=1	s=1	3							
	s=2			2					
k=2	s=1								3
	s=2						2		
k=3	s=1	2							
	s=2			5					
	s=3								3
k=4	s=1						6		
	s=2			4					
	s=3								5
k=5	s=1	5							
	s=2			1					
k=6	s=1								1
	s=2						2		

CHAPTER 8

COMPUTATIONAL EXPERIENCE

The models developed in this research are large scale linear or mixed integer programs. For the efficient and optimal solution of the linear and relaxed mixed integer programs a column generation based solution methodology was developed. A branch and bound scheme on the integer variables leads to an optimal solution for the mixed integer programs. Each node in the branch and bound tree represents a solution to an augmented continuous problem with additional constraints on the integer variables. These additional constraints are easily incorporated without increasing the size of the problem by the bounded variables procedure.

The amount of computation involved in finding a solution depends on two principal factors: the number of minimizations and the amount of computation involved in each minimization. The number of minimizations is strongly dependent upon the branching and the number of integer variables. This is not unique for a given problem since at any stage we are at liberty to form the next constraint using any non-integral variable. This has large effect on the computer time and the need for limiting it is apparent. The latter depends on the particular type of problem and the sub-algorithm used. This will be discussed next.

The sub-algorithm used in this research involves solving a large scale linear program (with lower or upper bounds on integer variables) by an efficient column generation scheme. The method of generating the column (i.e.,the plans) in the models developed in this research are by solving simple assignment problems. Only the assignment cost calculations are different in each case. The other columns, namely the slack, surplus and the integer variable columns are explicitly known in each model. The performance of the sub-algorithm will thus be similar for the models developed. The aim of this computational study is therefore to take Model-4.1 which is a mixed integer program and study its behavior for a few sample test problems.

The algorithm is given in chapter 4 (Algorithm-4.1). In this algorithm it is suggested that the first improving column be chosen and introduced into the basis. This is almost identical with the situation encountered in integer programming. In integer programming, a dual simplex method is used and there is a vast array of unwritten inequalities in the problem. No maximizations over the unwritten inequalities (rows) is possible. Since the dual method is used, rows play the same role that columns do in a primal method. For many types of integer programming problems, there is a problem sensitivity unknown in ordinary linear programming. Some problems run easily, others drone on and on. Experiments reported in Gilmore and Gomory (1963) suggest that the peculiarities of integer programming are not special to it but they are to be expected in large linear programming problems where there is no effective method of choosing among the multitude of rows in the dual simplex or among columns in the primal simplex.

With the peculiarities discussed, our interest here is in the number of pivot operations (iterations) performed in solving a linear program and the number of problems to be solved for an integer solution. The iteration count includes iterations in which the tableau is updated in the normal manner and also cases when a bounded variable goes from its lower to its upper bound or vice-versa.

The order in which the nodes are selected affects the computations. If this algorithm is used in hand computation, then it is sensible to proceed, at each stage, from the lowest cost terminal node which has been reached. This will minimize the searching of unlikely (i.e., high cost) sub-trees. One can use this nodal solution as a starting point for the optimization if the sub-algorithm permits it, since this is likely to be near to the next solution, and should require less computation than if we started at an arbitrary point. However, such a procedure is impracticable for implementation since the recording of the required information for each node involves an excessive amount of storage. Moreover, the sub-algorithm does not permit this implementation for the columns are known only implicitly. For the purpose of implementation the nodes are selected in the order in which they were created (first created first selected). To limit on the complete enumeration the maximum number of nodes created were restricted to 63 nodes. Thus, in case an optimal solution is not obtained, the corresponding node must be at a depth of 6 or more in the branch and bound tree.

8.1 Generation of Data Sets

The algorithm was coded in FORTRAN 77 and run on IBM 4381. The main parameters for computational results were considered to be:

1. Number of products K 10,20,50

2. Number of machine types M	2,3
3. Number of operations per part NOO and	3,5
4. Number of process plans per part NPP	1,2

These parameters determine the number of variables in the problem. It is assumed that each machine type can perform all operations. For the test problem with K= 20, M=3, NOO=5 and NPP=1, the minimum number of explicit columns is given by:

$NOO^{M} * K * NPP = 5^{3} * 20 * 1 = 2500$

However, all these columns need not be generated. The column generation scheme is used to generate the new columns as needed. The size of the basis for this test problem is 24 X 24. The number of integer variables are dependent on the number of types of machines. Since these are not more than 10 to 15 types, we have a problem of manageable size for the branch and bound scheme.

The IBM random number generator subroutine called RAND was used to generate the $c_{ms}(kp)$, $t_{ms}(kp)$, C_{m} and d_{k} values for the test problems from uniform distribution. The range considered for each parameter are:

1. Machine availability time (b_{m})	1000
2. Cost of machine type m (C_{m})	between 100 to 200
3. Processing time on machines ($t_{ms}(kp)$)	between 5 to 15
4. Operating cost on machines ($c_{ms}(kp)$)	between 5 to 15
5. Available budget (B)	10*NOO*K1*150
6. Demand for part type k (d_{k})	between 100 to 200

The machine availability time was assumed as 1000 time units. The available budget was set by taking the product of mean of operating cost, mean of demand, the number of part types and the number of operations per part. The results on the test problems are given next.

8.2 Results

To see the performance of the algorithm, the statistics collected is the number of pivot operations performed, number of nodes created, number of assignment problems solved and time for execution. The execution times are however dependent on the computer, load on the computer and efficiency of coding and by no means reflect the true efficiency of the algorithm. The number of pivot operations and the number of assignment problems solved reflects the performance of the algorithm better. A representative example of the computational results for ten part types, two types of machines, three operations on each part and one process plan for each part (i.e. k=10, m=2 , s=3, p=1) is shown in Table 8.1. The results on the test problems are shown in Table 8.2. The averages reported are over the number of minimizations performed for obtaining an integer optimal solution. In cases where the branching was stopped due to stopping criteria of a maximum of 63 nodes, the best integer solution, if found, is reported and the average is over the number of actual minimizations performed. An efficient implementation of this approach could be using a commercial code for solving linear/mixed integer programs and replacing the subroutine for checking the entering column with the sub-problem posed in this research. The limitation of applicability can then be mapped to the commercial code selected.

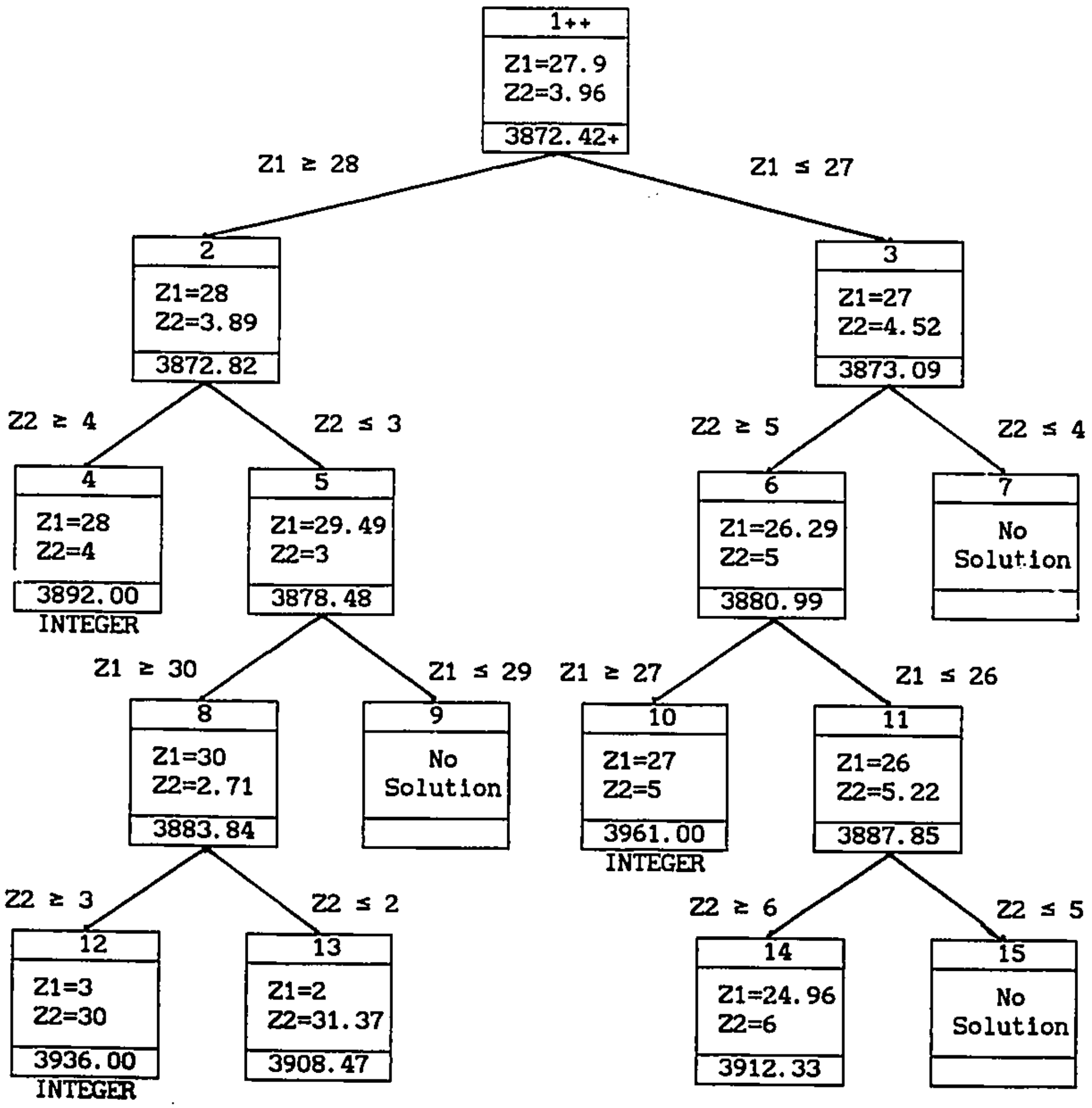
8.3 Precision in Computation

In the algorithms, discussed in this research, for calculating the pivot ratios it is necessary to check whether there exists a non-zero entry in a row of the current tableau. For this we have to look at each entry in that row and decide whether it is zero or not. This task is easy while solving a small problem by hand using integer arithmetic. However, to solve a large problem, we have to implement the algorithm by writing a computer program for it. Digital computers are finite precision machines, and usually any fraction encountered in the computation is converted into decimal form and rounded off to a fixed number of significant digits. This introduces a small round-off error at this stage, and as computations continue, these errors accumulate in every entry of the current tableau. If an entry in the current tableau is nonzero, but has a very small absolute value, it is very hard to decide whether that entry would have been zero or a nonzero number of small absolute value if all computations are performed exactly. Also, because of its finite precision feature, computers cannot distinguish between zero and a number whose absolute value is less than the machines precision limit. For this reason, algorithms that require checking to determine whether an entry in the current tableau is zero or nonzero could be very hard to carry out correctly. Normally, a control parameter known as the tolerance is specified, and any entry in the current tableau whose absolute value is less than the tolerance is treated as being equal to zero. Because of this, when solving large problems, the final answer produced by the computer may not be the true answer to the problem; it may only be approximately correct to within the limits of precision that we achieve (Murty 1983). The tolerance set for this purpose was .000001. Also, to avoid cumulation of these errors the basis was reinverted periodically after every 20 iterations. Nevertheless, it is impossible to eliminate roundoff errors. These roundoff are even more serious in the method of integer forms, where it is essential to identify integers. On a computer, a real number r is called integer, if min $(1-f_r, f_r) < e$, where e is specified by the program and f_r is the fractional part of r. The value of e was set to .03. Failing to recognize an integer could cause unnecessary iterations and even loss of optimal solution.

8.4 Summary

A representative mixed integer program (Algorithm 4.1) was coded in FORTRAN 77 and run on IBM 4381. A few test problems were randomly generated and statistics on number of nodes, number of plans generated, number of pivot operations, number of assignment problems solved and time for execution were collected. The precision in computation and limitation of applicability of the algorithm were briefly discussed.

TABLE 8.1: Branch and Bound tree for a test problem



+ Objective value

++ Node number

TABLE 8.2: Computational Results

p	k	m	s	NON	OAN	NPS$^{\bullet}$	NP$^{\bullet}$	NASS$^{\bullet}$	NAPP$^{\bullet}$	TIME$^{\bullet}$
1	10	2	3	15	4	34.2	35.6	162.3	4.75	333.1
			5	11	4	37.8	39.5	167	4.42	397.6
		3	3	44	22	73.6	77.2	281.2	3.82	847.1
			5	25	8	104.7	107.6	372.3	3.56	1230
	20	2	3	11	10	61.5	63.1	501.3	8.15	1636.6
			5	11	11	123.1	124.7	716.4	5.82	3334.5
		3	3	63	56	189.8	198	1093.5	5.76	5640.4^{+}
			5	63	28	239.6	246.7	1266.2	5.28	7231.5^{+}
2	10	2	3	19	11	52.6	54.7	370.7	7.05	513.7
			5	15	12	83.2	84.7	560.1	6.73	890.6
		3	3	21	18	109.4	112.9	659.8	6.03	1311.6
			5	7	7	144.3	148.9	894.1	6.2	1824
	20	2	3	3	2	89	91	1286.1	14.45	2486
			5	3	3	116.3	119	1554.3	13.36	3377.3
		3	3	56	33	210.4	215.9	2209.4	10.5	6377^{+}
			5	55	45	461.1	469.4	4000.9	8.68	14032.8^{+}
	50	3	5	7	6	792.6	813.9	15296.1	19.3	108458.8

The following symbols are used in the table:

p - process plan

k - part

m - machine

s - operation

NON - Number of Nodes or Number of Minimizations

OAN - Optimal solution obtained at this node

NPS - Average number of plans selected to enter basis

NP - Average number of pivot operations performed

NASS - Average number of assignment problems solved

NAPP - Average number of assignment problems solved per plan selected

TIME - Average time for execution in milliseconds

• The averages reported are for each minimization

+ The problems stopped due to stopping criteria

CHAPTER 9

CONCLUSIONS

The contributions of the research to the "Design of Cellular Manufacturing Systems" are summarized in this chapter. The limitations of the work and directions for future research are also given.

9.1 Contributions of the Research

In cellular manufacturing, parts with design and manufacturing similarities are grouped into part families and the associated machine groups are formed such that one or more part families can be fully processed within a single cell. The identification of part families and machine groups in the design of cellular manufacturing systems is commonly referred to as "cell formation". In this research a number of issues has been identified which should be considered in cell formation for the successful adoption of the cellular manufacturing systems. A number of mathematical formulations has been developed to help identify part families and machine groups for the design of CM systems. The inputs required for solving all the models and the outputs generated from them are given in Appendix A.2. A representation of the framework for cell design without inter-cell movements and with inter-cell movements is given in Appendices A.3 and A.4 respectively. The contributions of the research to design of cellular manufacturing systems are given below:

1) The influence of alternate process plans on resource utilization was

discussed. Three mathematical models were developed to illustrate that the simultaneous grouping model will always provide results at least as good as the sequential model.

2) The creation of cells using new and often automated machines under the following conditions were considered:

   i) Identify machine groups such that all the parts in a family are processed within the group.

   ii) Form the desired number of cells by assigning one or more known part families and machines, to cells.

   iii) Simultaneously determine the part families and machine groups.

   Four mathematical models were developed to address these issues and presented in chapter 4. Model 4.1 assigns machines to parts. The assignment of machines to known part families was considered in Model 4.2. Model 4.3 identifies part families and machine groups simultaneously. Forming cells by permitting one or more part families in a cell was considered in Model 4.4.

3) The issues related to redesign of cells such as relocating machines, change in part mix, etc. while identifying part families and machine groups were addressed. The following situations in the redesign of cells were considered:

   i) Allocation of parts to known machine groups.

   ii) Allocation of parts to known machine groups allowing for relocation of machines between cells to form mutually independent cells.

   iii) If sufficient capacity is not available, then determine if relocation be accompanied or substituted by a higher degree of investment.

Four mathematical models were developed for this purpose and presented in chapter 5. Model 5.1 allocates parts to known machine groups without disrupting the existing configuration. Model 5.2 identifies part families and machine groups simultaneously considering machine relocation. In addition to simultaneously forming part families and machine groups, Model 5.3 also determines the additional investment to be incurred on new machines. Model 5.4 considers only additional investment on new machines assuming relocation is not allowed due to loss in production and time.

4) The issues related to determination of part and machine spectra for cellular manufacturing were addressed. The problems considered in this context are:

   i) Selection of a subset of parts to be produced on known machine groups.

   ii) Simultaneously identifying the subset of machines and parts to form the required number of cells.

Four mathematical models were developed for this purpose and presented in chapter 6. Model 6.1 selects the portion of parts demand to be produced on known machine groups. Model 6.2 identifies the portion of parts demand to be produced in cells and the machines to be selected for cellularization simultaneously. Models 6.1 and 6.2 may identify only a portion of the demand to be produced in cells. This situation is advantageous if the decision maker does not want to depend on the cell as the only source of supply during implementation. However, if one feels that the total demand for a part identified for cell production should be completely produced in one or more cells, Models 6.3 and 6.4 are presented for this purpose.

5) Issues such as material handling and replacement of old machines by new or technologically updated machines were considered. For this purpose four mathematical models were developed and presented in chapter 7.

6) A number of objectives such as minimization of investment, operating cost, machine relocation cost, material handling cost and maximization of output were considered.

7) Physical limitations such as upper bound on cell size, machine capacity and material handling capacity in the cell design process were explicitly modeled.

8) Efficient solution schemes based on the column generation approach for the linear and relaxed mixed integer programs, were developed.

9) A number of illustrative examples was solved. Computational experience on test problems giving statistics on number of nodes, number of plans generated, number of pivot operations, number of assignment problems solved and time for execution was also reported.

In summary, the cell formation problem addressed in this research, in addition, to identifying part families and machine groups specifies the plans selected for each part, quantity to be produced through the plans selected, machine type to perform each operation in the plans, total number of machines required, machines to be relocated, machines to be replaced and parts and machines selected for cellularization considering demand, time, material handling, resource constraints, etc..

9.2 Limitations of the Research

The limitations of the research reported are:

1) The demand for parts in the given time horizon is assumed to be deterministic. The cost and time of operations are also assumed to be constant.
2) A number of other factors such as fixtures available, tool magazine capacity on machines, tool life, number of tools available, etc. influence the cell formation. These aspects have been ignored.
3) Different strategies for generating the columns for basis entry and node selection criteria for branch and bound have not been examined.

9.3 Directions for Future Research

1. Other factors such as number of fixtures available, tool magazine capacity on machines, tool life, number of tools available etc. may be considered for cell design.
2. The framework for cell design provided may be extended to multi-period situations.
3. Simulation models can be developed to evaluate the performance of cells developed by mathematical programming in this thesis.
4. A number of manufacturing companies have repetitive manufacturing (closed job shop) where sequence dependence need to be considered

in cell design.

5. Another area of interest could be cell layouts. Flow line cells seem to have several advantages over job shop cells. The interesting issue here is: How to achieve flow line cells and to what extent should this be done ?

6. The consideration of uncertainities in the product mix, demand, availability of machines, etc., in cell design will be an interesting research area.

REFERENCES

Askin,R.G and Chiu,K.S,'A graph partitioning procedure for machine assignment and cell formation in group technology', Working Paper #87-009,College of Engineering,The University of Arizona,Tuscon,Arizona 85721,(1988).

Askin,R.G and Subramanian,S.P,'A cost based heuristic for GT configuration',International Journal of Production Research,Vol.25,No.1,pp101-113,(1987).

Ballakur,A and Steudel,H.J,'A within cell utilization based heuristic for designing cellular manufacturing systems', International Journal of Production Research,Vol.25,No.5, pp639-665,(1987).

Bhat,M.V and Haupt,A,'An efficient clustering algorithm', IEEE Transactions on Systems,Man and Cybernetics,SMC.6,No.1, pp61-64,(1976).

Burbidge,J.L,'Production flow analysis',Production Engineer, Vol.50,No.4/5,pp139-152,(1971).

Burbidge,J.L,'Production flow analysis on the computer',Third Annual Conference of the Institute of Production Engineers, Group Technology Division,(1973).

Burbidge,J.L,'A manual method for production flow analysis', Production Engineer,Vol.56,No.10,pp34-38,(1977).

Buzacott,J.A and Yao,D.D,'Flexible manufacturing systems:a review of analytical models',Management Science,Vol.32,No.7, pp890-905,(1986).

Carrie,A.S,'Numerical taxonomy applied to GT and plant layout',International Journal of Production Research,Vol.11, No.4,pp399-416,(1973).

Chan,H.M and Milner,D.A,'Direct clustering algorithm for group formation in cellular manufacture',Journal of Manufaturing Systems,Vol.1,No.1,pp64-76,(1982).

Chandrasekaran,M.P and Rajagopalan,R,'An ideal seed non-heirarchical clustering algorithm for cellular manufacturing' International Journal of Production Research,Vol.24,No.2, pp451-464,(1986).

Chandrasekaran,M.P and Rajagopalan,R,'MODROC:an extension of rank order clustering for group technology',International Journal of Production Research,Vol.24,No.5,pp1221-1233,(1986).

Chandrasekaran,R,Aneja,Y.P and Nair,K.P.K,'Production planning in assembly line systems',Management Science,Vol.30, No.6,pp713-719,(1984).

Choobineh,F,'A framework for the design of cellular manufacturing systems',International Journal of Production Research,Vol.26,No.7,pp1161-1172,(1988).

Chu,C.H and Pan,P,'The use of clustering techniques in manufacturing cellular formation',Proceedings: International Industrial Engineering Conference, Orlando, Florida, pp495-500,(1988).

Co,H,C and Arrar,A,'Configuring cellular manufacturing systems',International Journal of Production Research,Vol.26, No.9,pp1511-1522,(1988).

Dakin,R.J,'A tree search algorithm for mixed integer programming problems,Computer Journal,Vol.8,pp250-255,(1965).

Dantzig,G and Wolfe,P,'The decomposition algorithm for linear programming',Econometrica,Vol.9,No.4,(1961).

De Beer,C,Van gerwen,R and De Witte,J,'Analysis of engineering production systems as a base for production oriented reconstruction',Annals of CIRP,Vol.25,No.1,pp439-441,(1976).

De Beer,C and De Witte,J,'Production flow synthesis',Annals of CIRP,Vol.27,pp389-341,(1978).

De Witte,J,'The use of similarity coefficients in production flow analysis',International Journal of Production Research, Vol.18,No.4,pp503-514,(1980).

Dutta,S.P,Lashkari,R.S,Padhye,A, Nadoli,G and Ravi,T,'The cell formation problem in FMSs',Proceedings:Fourth International Conference on Flexible Manufacturing Systems, Stockholm,Sweden,(1985).

El Essawy,I.F.K and Torrance,J,'Component flow analysis:an effective approach to production systems design',The Production Engineer,Vol.25,No.5,pp159-164,(1972).

Faber,Z and Carter,M.W,'A new graph theory approach for forming machine cells in cellular production systems',FMS:Methods and Studies,A.Kusiak (ed.),North-Holland,pp301-315,(1986).

Gilmore,P.C and Gomory,R.E,'A linear programming approach to the cutting stock problem',Operations Research, Vol.9,No.6, pp849-859,(1961).

Gilmore,P.C and Gomory,R.E,'A linear programming approach to the cutting stock problem-part 2,Operations Research,Vol.10, No.6,pp863-887,(1963).

Glover,F and Woolsey,E,'Further reduction of zero-one polynomial programming problems to a 0-1 linear program', Operations Research,Vol.21,No.1,pp156-161,(1973).

Gongaware,T.A and Ham,I,'Cluster analysis applications for GT manufacturing systems',Proceedings:IX North American Metal Working Research Conference,SME,Dearborn,Michigan,pp503-508(1981).

Han,C and Ham,I,'Multiobjective cluster analysis for part family formations',Journal of Manufacturing Systems,Vol.5, No.4,pp223-230,(1986).

Harhalakis,G,Nagi,R and Proth,J.M,'An efficient heuristic in manufacturing cell formation for group technology applications',International Journal of Production Research, Vol.28,No.1,pp185-198,(1990).

Hyer,N.L and Wemmerlov,U,'Group technology oriented coding systems:structures,applications and implementation', Production and Inventory Management,Vol.26,No.1,pp55-78,(1985).

Kasilingam,R.G,'Mathematical programming approach to cell formation problems in flexible manufacturing systems',Ph.D Dissertation,University of Windsor,Windsor,Ontario,(1989).

Khator,S.K and Irani,S.A,'Cell formation in group technology -a new approach',Computers and Industrial Engineering,Vol.12, No.2,pp131-142,(1987).

King,J.R,'Machine-component grouping in production flow analysis:an approach using rank order clustering algorithm', International Journal of Production Research,Vol.18,No.2 pp213-232,(1980).

King,J.R,'Machine-component group formation in group technology', OMEGA,Vol.8,No.2,pp193-199,(1980).

King,J.R and Nakornchai,V,'Machine-component group formation in group technology:review and extension',International Journal of Production Research,Vol.20,No.2,pp117-133,(1982).

Kumar,K.R,Kusiak,A and Vannelli,A,'Grouping of parts and components in flexible manufacturing systems',European Journal of Operational Research,Vol.24,No.3,pp387-397,(1986).

Kusiak,A,'Part families selection model for flexible manufacturing systems',Proceedings:Annual Industrial Engineering Conference,Louisville,Kentucky,pp 575-580,(1983).

Kusiak,A,Vannelli,A and Kumar,K.R,'Grouping problem in scheduling flexible manufacturing systems',Robotica,Vol.3 pp245-252,(1985).

Kusiak,A,'The part families problem in FMSs',Annals of Operations Research,Vol.3,pp279-300,(1985).

Kusiak,A,'The generalized group technology concept', International Journal of Production Research,Vol.25,No.4, pp561-569,(1987).

Kusiak,A and Chow,W.S,'Efficient solving of GT problem', Journal of Manufacturing Systems,Vol.6,No.2,pp117-124,(1987).

Lemoine,Y and Mutel,B,'Automatic recognition of production cells and part families',Advances in CAD/CAM,T.M.R.Ellis and O.I.Semenkov (eds.),North Holland Publishing Co. Amsterdam, pp 239-248,(1983).

McAuley,J,'Machine grouping for efficient production', The Production Engineer,Vol.51,No.2,pp53-57,(1972).

Meenakshisundaram,R and Shong-shun fu,'Group technology cell formation-some new insights',Proc.of 9th Annual conf. on Comp. and Ind.Engg.,Vol.13,pp267,(1987).

Mosier,C.T,'An experiment investigating the application of clustering procedures and similarity coefficients to the GT machine cell formation problem',International Journal of Production Research,Vol.27,No.10,pp1811-1835,(1989).

Murty,K.G,Linear Programming,John Wiley & Sons,(1983).

O'Grady,P.J and Menon,U,'A concise review of flexible manufacturing systems and FMS literature,Computers and Industry, Vol.7,pp155-167,(1986).

Opitz,H,'A classification system to describe work pieces', Pergamon Press,Oxford,(1970).

Purcheck,G,'Combinatorial grouping-a lattice theoretic method for the design of manufacturing systems',Journal of Cybernetics,Vol.4,No.3,pp27-60,(1974).

Purcheck,G,'A mathematical classification as a basis for the design of group technology production cells',The Production Engineer,January,Vol.54,No.1,pp35-48,(1975).

Purcheck,G,'A linear programming method for the combinatoric grouping of an incomplete power set',Journal of Cybernetics, Vol.5,No.4,pp51-76,(1975).

Purcheck,G,'Machine-component group formation:a heuristic method for flexible production cells and flexible manufacturing systems',International Journal of Production Research,Vol.23,No.5,pp911-943,(1985).

Rodriguez,R and Adaniya,O,'Group technology cell allocation', Proceedings:Fall Industrial Engineering Conference,Chicago, Illinois,pp125-129,(1985).

Rajagopalan,R and Batra,J.L,'Design of cellular production systems-a graph theoretic approach',International Journal of Production Research,Vol.13,No.6,pp567-579,(1975).

Seifoddini,H,'Duplication process in machine cells formation in group technology',IIE Transactions,Vol.21,No.4,pp382-388,(1989).

Seifoddini,H and Wolfe,P.M,'Application of similarity coefficient method in GT',IIE Transactions,Vol.18,No.3, pp271-277,(1986).

Shtub,A,'Modelling group technology cell formation as a genearlized assignment problem',International Journal of Production Research,Vol.27,No.5,pp775-782,(1989).

Srinivasan,G,Narendran,T.T and Mahadevan,B,'An assignment model for the part-families problem in group technology', International Journal of Production Research,Vol.28,No.1, pp145-152,(1990).

Steudel,H.J and Ballakur,A,'A dynamic programming based heuristic for machine grouping in manufacturing cell formation',Computers and Industrial Engineering,Vol.12,No.3, pp215-222,(1987).

Vannelli,A and Kumar,K,R,'A method for finding minimal bottle neck cells for grouping part-machine families', International Journal of Production Research,Vol.24,No.2, pp387-400,(1986).

Waghodekar,P.H and Sahu,H,'Machine-component cell formation in GT: MACE',International Journal of Production Research, Vol.22,No.6,pp937-948,(1984).

Wemmerlov,U and Hyer,N.L,'Procedures for the part family/ machine group identification problem in cellular manufacturing',Journal of Operations Management,Vol.6,No.2, pp125-147,(1986).

Wemmerlov,U and Hyer,N.L,'Research issues in cellular manufacturing',International Journal of Production Research, Vol.25,No.3,pp413-431,(1987).

Wemmerlov,U and Hyer,N.L,'Cellular manufacturing in the US industry:a survey of users',Working paper,Graduate School of Business Administration,University of Wisconsin,Madison, Wisconsin,(1987).

APPENDIX A.1

The Column Generation Procedure

In the simplex algorithm using canonical tableaux, many computations are performed at every pivot step. Every time a pivot is performed, it is carried out on every column of the tableau. This can be very time consuming. On the other hand, in the revised simplex method, in any step, we only need the data contained in the present basic columns and then we need a subroutine to check the dual feasibility of the present dual basic solution. This makes it possible for us to solve large scale linear programming problem by the simplex algorithm without having the data in the model explicitly in hand at any time. Suppose the linear program (LP) in the standard form is

$$\begin{aligned} &\text{Minimize} \quad z(x) = cx \\ &\text{subject to} \quad Ax = b \qquad\qquad (1) \\ &\qquad\qquad\quad x \geq 0 \end{aligned}$$

where $b \geq 0$ and A is of order $m * n$.

The subroutine for checking the dual feasibility takes as input the present dual vector, π and should be able to determine whether " $\pi A_{.j} \leq c_j$ " holds for all j (in which case the present primal Basic Feasible Solution is optimal), or produce a non basic column $\left(\ldots\begin{matrix} A_{.j} \\ c_j \end{matrix}\ldots\right)$ in (1) that violates dual feasibility, that is it satisfies $\pi A_{.j} > c_j$. This column vector is all that we need to move to the next basic vector (essentially the variable corresponding to this column vector is treated as the entering variable, and the basis change is carried out by the revised simplex format) and the whole process is repeated starting with the new basic vector. It is not necessary to know what procedure the subroutine employs to check dual feasibility; in the sense the subroutine for checking dual feasibility can be treated as a black box for executing the revised simplex algorithm on (1). In some applications, the problem of checking the dual feasibility of a given vector can itself be posed as an auxiliary problem and the subroutine for checking dual feasibility may be an algorithm for solving this auxiliary problem. In this procedure, we only maintain the present set of basic columns in the original linear programming model; the entering columns are generated one by one as they are needed, using the subroutine for checking dual feasibility. Hence, this procedure is known as column generation procedure (Murty 1983).

APPENDIX A.2

INPUTS AND OUTPUTS FOR ALL THE MATHEMATICAL MODELS

Inputs required:

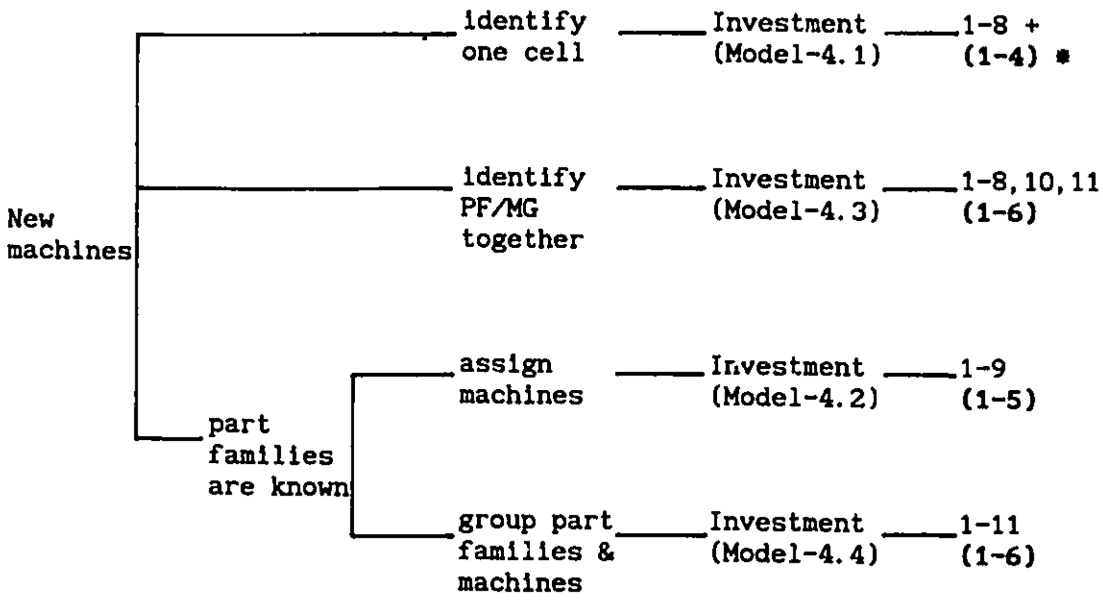
1. Number of part types
2. Demand for parts
3. Process plans for each part type
4. Number of machine types
5. Cost of each machine type
6. Capacity on each machine type
7. Cost and time of operations
8. Operating budget
9. Known part families
10. Number of cells to be formed
11. Maximum number of machines in a cell/group
12. Cost of assigning a machine to a cell/group
13. Known machine groups
14. Cost of relocating a machine
15. Cost of moving a part between groups
16. Distance between machine groups
17. Capacity of material handling equipment
18. Cost of capital
19. Book value of each machine type
20. Depreciation value of each machine type

Outputs obtained:

1. Plans selected for each part type
2. Number of units to be produced through the plans selected
3. Machine type to perform each operation
4. Total number of machines of each type
5. Total number of machines of each type in a cell/group
6. Part types in each cell
7. Machines to be relocated
8. Total number of inter-cell movements
9. Machines to be replaced

APPENDIX A.3

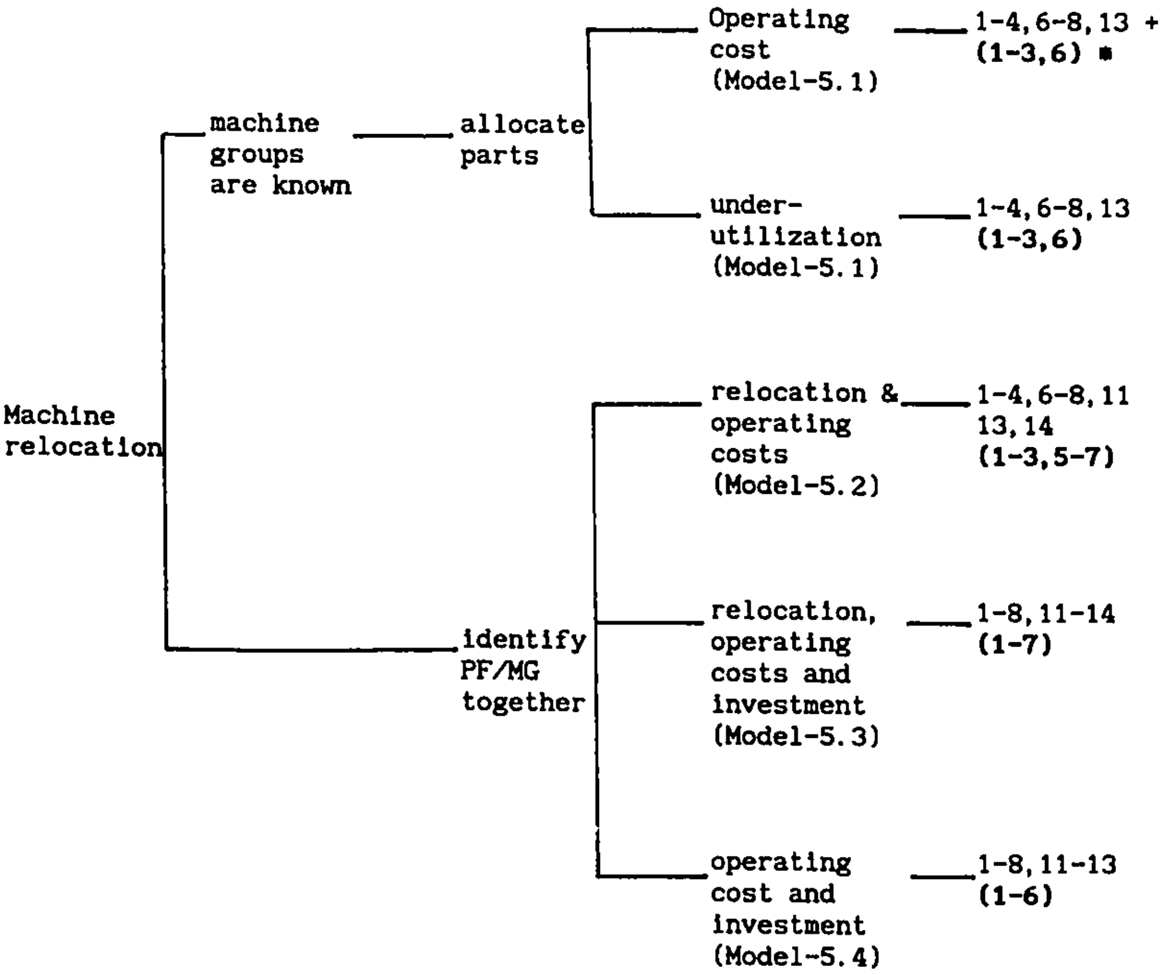
CELL DESIGN WITHOUT INTER-CELL MOVEMENTS



PF-Part family
MG-Machine group
+ Inputs (see Appendix A.2)
* Outputs

APPENDIX A.3

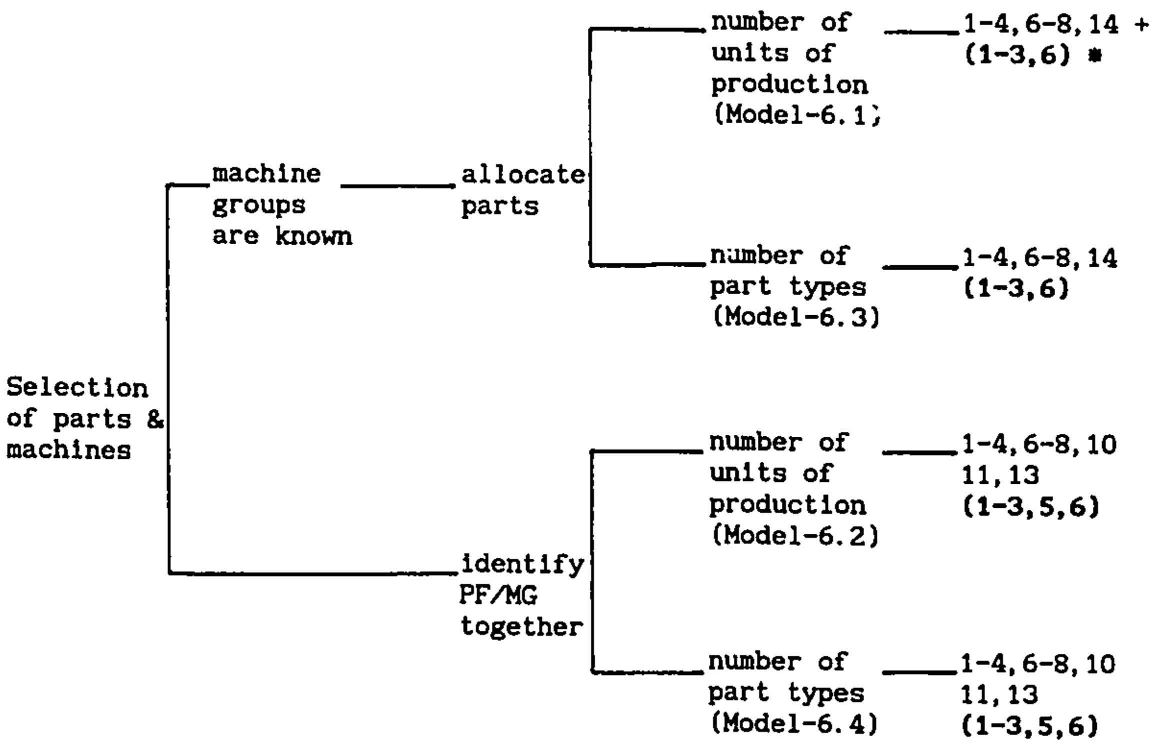
CELL DESIGN WITHOUT INTER-CELL MOVEMENTS


Machine relocation
machine groups are known
allocate parts
Operating cost (Model-5.1)
1-4,6-8,13 +
(1-3,6) *
under-utilization (Model-5.1)
1-4,6-8,13
(1-3,6)
identify PF/MG together
relocation & operating costs (Model-5.2)
1-4,6-8,11
13,14
(1-3,5-7)
relocation, operating costs and investment (Model-5.3)
1-8,11-14
(1-7)
operating cost and investment (Model-5.4)
1-8,11-13
(1-6)

PF-Part family
MG-Machine group
+ Inputs (see Appendix A.2)
* Outputs

APPENDIX A.3

CELL DESIGN WITHOUT INTER-CELL MOVEMENTS


Selection of parts & machines
machine groups are known
allocate parts
number of units of production (Model-6.1)
1-4,6-8,14 +
(1-3,6) *
number of part types (Model-6.3)
1-4,6-8,14
(1-3,6)
identify PF/MG together
number of units of production (Model-6.2)
1-4,6-8,10
11,13
(1-3,5,6)
number of part types (Model-6.4)
1-4,6-8,10
11,13
(1-3,5,6)

PF-Part family
MG-Machine group
+ Inputs (see Appendix A.2)
* Outputs

APPENDIX A.4

CELL DESIGN WITH INTER-CELL MOVEMENTS

New machines		identify PF/MG together	investment, operating & material handling costs (Model-7.1)	1-8, 10-12, 15 + (1-6,8) *
Machine relocation	machine groups are known	allocate parts	operating & material handling costs (Model-7.2)	1-4, 6-8, 13 15-17 (1-3,6,8)
		identify PF/MG together	relocation, operating, material handling costs and investment (Model-7.3)	1-8, 10-17 (1-8)
Machine replacement		identify PF/MG together	operating, material handling costs & loss of interest on capital (Model-7.4)	1-8, 10-20 (1-9)

PF-Part family
MG-Machine group
+ Inputs (see Appendix A.2)
* Outputs

VITA AUCTORIS

1964 Born in Madras, India on the 2nd of January

1981 Completed High school education from St. Johns Sr. Sec. School, Meerut, India

1985 Graduated from Institute of Technology, Banaras Hindu University, Varanasi, India with a Bachelor's degree (Honours) in Mechanical Engineering

1986 Graduated from Indian Institute of Technology, Delhi, India with a Master's degree in Industrial Engineering

1987 Worked in the capacity of Assistant Manager (Production-Planning and Control) with Fusebase Eltoro Pvt.Ltd, India

1990 Currently a candidate for the degree of Doctor of Philosophy in Industrial Engineering at the University of Windsor, Windsor, Canada.