SOME ASPECTS OF SYSTEM DESIGN FOR PRODUCTION MANAGEMENT IN THE ROLLING-MILL INDUSTRY

HAMID YAZDIANPOUR

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

December 1989

The University of Aston in Birmingham

SOME ASPECTS OF SYSTEM DESIGN FOR PRODUCTION MANAGEMENT IN THE ROLLING-MILL INDUSTRY

Hamid Yazdianpour

PhD 1989

Summary

Case studies in copper-alloy rolling mill companies showed that existing planning systems suffer from numerous shortcomings. Where computerised systems are in use, these tend to simply emulate older manual systems and still rely heavily on modification by experienced planners on the shopfloor. As the size and number of orders increase, the task of process planners, while they are seeking to optimise the manufacturing objectives and keep within the production constraints, becomes extemely complicated because of the number of options for mixing or splitting the orders into batches.

This thesis develops a modular approach to computerisation of the production management and planning functions. The full functional specification of each module is discussed, together with practical problems associated with their phased implementation.

By adapting the Distributed Bill of Material concept from Material Requirement Planning (MRP) philosophy, the production routes generated by the planning system are broken down to identify the rolling stages required. Then, to optimise the use of material at each rolling stage, the system generates an optimal cutting pattern using a new algorithm that produces practical solution to the cutting stock problem.

It is shown that the proposed system can be accomodated on a micro-computer, which brings it into the reach of typical companies in the copper-alloy rolling industries, where profit margins are traditionally low and the cost of widespread use of mainframe computers would be prohibitive.

KEY WORDS: Computer Integrated Manufacturing (CIM)
Computer Aided Process Planning (CAPP)
Materials Requirement Planning (MRP)
Material Yield Optimisation
Cutting Stock Problem

ACKNOWLEDGEMENTS

This research was mainly funded by a research grant awarded by Aston University to the Department of Mechanical and Production Engineering, in which the author was based before transferring to Aston Business School, where the work was completed.

I am particularly indebted to my Supervisor, Dr. D.A. L. Wilson, for his continuing interest, and invaluable discussions and guidance at all times.

I would also like to express my sincere thanks to my colleague Dr. M. Husseini for his help, cooperation and discussions on many aspects of the project.

Other members of staff at Aston, to whom I am very grateful for support and advice during my research are: Dr. G. Beaumont, Dr. D. M. Love, Dr. J. E. T. Penny and Mr. G. A. Jones.

I would also like to record my gratitude to my wife for her help in typing and proof-reading and for her encouragement and patience.

CONTENTS

Page

CHAPTER THREE

CASE STUDIES AND GOAL RECOGNITION

Page

CHAPTER FOUR

SYSTEMS ANALYSIS AND CONCEPTION

Page

CHAPTER FIVE

DEVELOPMENT OF THE MANUFACTURING MANAGEMENT FUNCTION

Page

Page

CHAPTER SIX

DEVELOPMENT OF THE MANUFACTURING SUPPORT FUNCTIONS AND STOCK CONTROL SYSTEM

Page

CHAPTER SEVEN

DEVELOPMENT OF COMPUTER AIDED MANUFACTURING OPTIMISATION (CAMO)

Page

CHAPTER EIGHT

DEVELOPMENT OF TECHNIQUES OF ONE-DIMENSIONAL CUTTING STOCK PROBLEM

Page

CHAPTER NINE

DEVELOPMENTS OF TECHNIQUES OF TWO /THREE DIMENSIONAL CUTTING STOCK PROBLEM

CHAPTER TEN

IMPLEMENTATION OF CSP SOLUTIONS AND THE DEVELOPMENT OF A ROUNDING TECHNIQUE

Page

CHAPTER ELEVEN

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

LIST OF FIGURES AND TABLES

Figures		Page

<u>Tables</u> <u>Page</u>

CHAPTER 1

INTRODUCTION

Today, more and more companies are benefitting from the latest advances in information technology. Particularly in the manufacturing industry, in such areas as product design, product structure, sales order processing, stock control, process planning, production control and a whole host of other manufacturing activities, a number of complex computerised systems have been developed. Attempts have been made at integrating these 'islands of automation', as they have been called by Lung (1988). Such integration is generally known as Computer Integrated Manufacturing (CIM). CIM systems are attractive in the current competitive market environment, where there is pressure to improve productivity and customer service, lower manufacturing costs and hence achieve increased profits - the ultimate goal of any enterprise.

It cannot be denied, however, that the initial cost of implementing such CIM systems can prove prohibitive for some companies. For example, in its marketing publication 'Computer Integrated Manufacture', British Telecom quotes a British company, who have recently invested some £4 million in such a system. At present, one industy in particular, the rolling mill industry, is facing increasing foreign competition. Due to the low profit margin and high operation costs, the management are not in a position to commit themselves to large expenditure for CIM systems, without the financial assistance from either their parent company, or organisations such as the Department for Trade and Industry (DTI).

In the author's experience, management in the rolling mill industry are aware of the need for some computerized solutions to the problems they face in almost all areas of manufacturing. Lack of finance, as previously mentioned, leads them to search for help and advice from non-commercial organisations, such as government bodies and universities. It was therefore in this area that the research described in this thesis was carried out.

During the course of this project, two companies of different sizes were studied, henceforth referred to as 'Company A' and' Company B'. We are primarily concerned, however, with Company B because Company A experienced many problems (outlined in Chapter 3) and eventually went into liquidation.

The author ascertained that one of the major factors contributing to low efficiency at both companies was the poor manufacturing planning and control. Because of the broad nature of the problems, it would have been impossible to carry out a research programme to cover all aspects of planning and and control. Thus, the project was divided into two areas of reasearch: (a) Computer Aided Process Planning (CAPP) and (b) Computer Aided Material Optimisation (CAMO) and a proposal for the development of a Computer Aided Production Management System (CAPM) for the rolling mill industry. Husseini (1989) tackles the first area of reasearch and the present author the second area.

Chapter 2 provides a background to some of the manufacturing terms used in the rolling mill industry and gives an outline of the processes and operations involved in the manufacture of rolled products. It touches upon the problems involved in the process control that concern the mill schedules for the mill operators.

Chapter 3 provides case studies for both Companies A and B and explains why Company A was eventually abandoned, leading the author to concentrate on Company B alone. An outline is also given of the process planning techniques available to overcome some of the problems, with an outline of solutions generated by the computer aided process planning system developed by Husseini (1989).

The fourth chapter examines the systematic approaches for the development of computerized systems. It outlines the general designs of the proposed computer aided production management system, necessary for the rolling mill environment.

Chapters 5 and 6 examine the detailed functional specifications of the CAPM proposal and suggest means of integrating the developed systems. The functions described are based on a powerful data base management system such as Oracle.

Chapter 7 is concerned with the development of a computerised system, similar to that of Materials Requirement Panning (MRP) systems. It explains a novel approach, by which process planning routes generated by the CAPP system are converted into a structure similar to that of Bills of Material (BOM). The concept based on distributed production planning and control in a cellular CIM system [Love et al. (1986), Love and Barekat (1988)] is used to illustrate how the process structure can be broken down so that Cutting Sock Problem (CSP) techniques can be applied to generate cutting patterns applicable for those processes involved. It also suggests that at each stage of rolling, the lead times and inventory levels are examined to calculate the net material requirements and their respective lead times.

Chapter 8 provides an in-depth study of the techniques available on the one-dimensional cutting stock problem and highlights the problems and constraints each pose for their application. It then selects a suitable method based on the Gilmore-Gomory techniques (1961) and points out areas where the chosen technique lacks the necessary algorithms for practical implementation in a rolling mill company or similar environment.

Chapter 9 examines the various methods generally used for the two or three-dimensional cutting stock problem and explains the problems and constraints involved in using those methods. A system again based on the Gilmore-Gomory (1965, 1966) approach is developed to evaluate the applicability for two-dimensional cutting and implications of their use.

Chapter 10 provides a detailed analysis of the sources of scrap and describes the rounding algorithm which has been developed so that the Gilmore-Gomory (1961) solutions to the cutting stock problem can be modified and incorporated into a practical system. The range of options for rounding solutions to the one-dimensional CSP is fully explored. The overall flow chart of the program logic is presented and the integration of the CSP function with CAMO is described.

Finally, Chapter 11 presents an overview of the proposed system and suggests the areas for further work. All chapters include an introduction and concluding summary.

Chapter 2

PRODUCTION OF ROLLED PLATES, SHEETS AND COILS

2.1 Introduction

Rolled metals, both ferrous and non-ferrous, are in universal use and in almost all branches of manufacturing engineering their applications are evident. The objective of this chapter is to describe briefly the production of plates, sheets and coils as it is the business of the companies involved in this research. The chapter also explains the complex task of manufacturing planning and control. Although a number of companies in this area were studied, greater emphasis has been given to one in particular.

Copper and copper alloys pass through many processes from mines to consumers. To provide a complete review of the whole process is outside of the scope and objectives of this project. However, the sequence of these processes is summarised in Figure 2.1, with an overview of definitions and descriptions of the main technical terms used in this environment.

2.2 Working of Metals and Their Production

Metal deformation processes are normally divided into the two broad categories of 'hot' and 'cold' working operations. In a hot working process, the metal temperature is usually above that of recrystallisation for the workpiece and so the metal does not work-harden. At this temperature (usually about 750 C°), the gauge reduction ratio is fairly high and the main reduction in material thickness takes place at this stage of rolling by only a few passes.

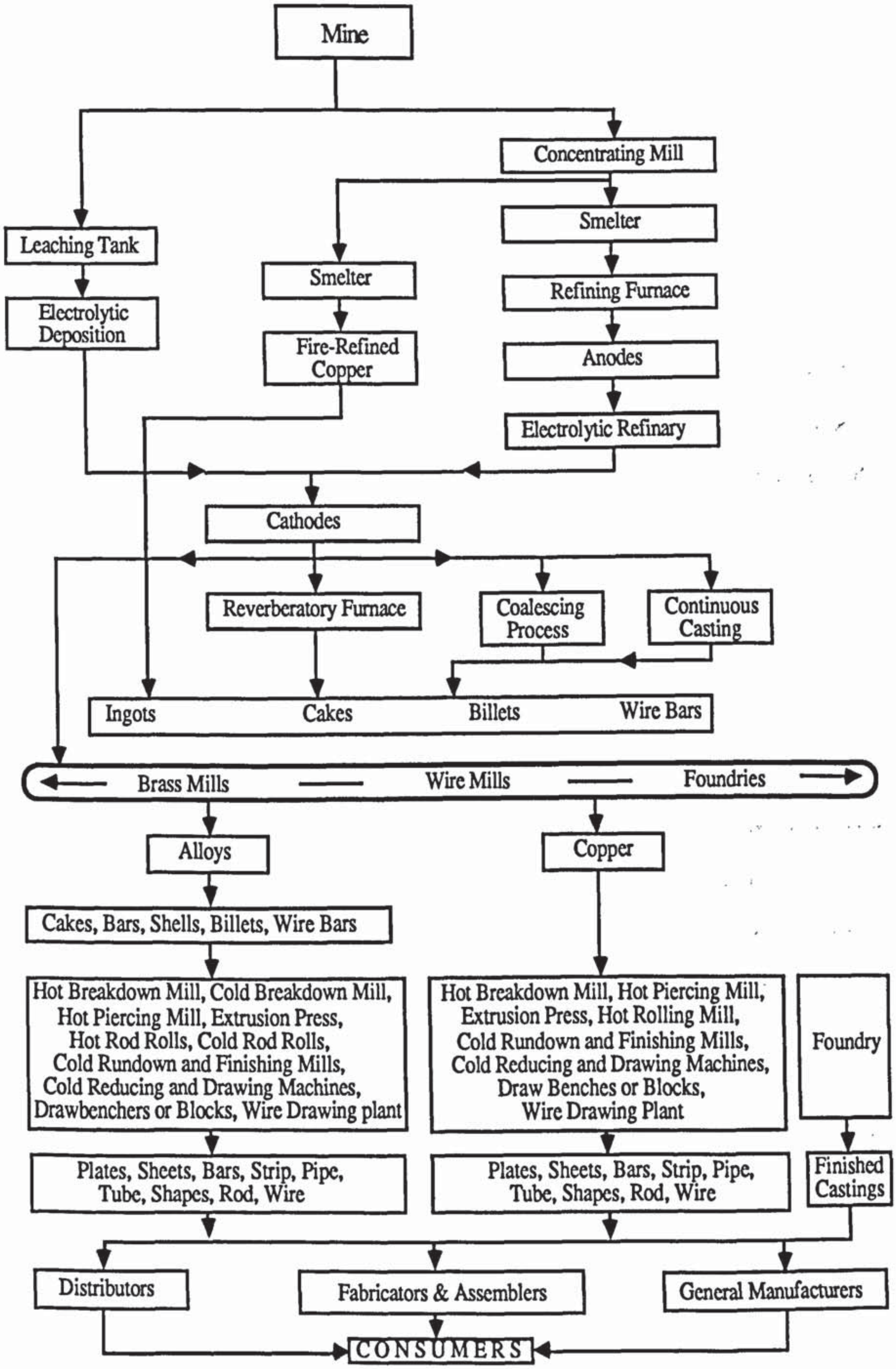


Figure 2.1 The Manufacturing Stages of Rolled Products From Mining to Consumers.

The main disadvantages of hot rolling are its poor surface finish and poor control of dimensions to a high degree of accuracy.

Cold working, however, is applied to most products as an intermediate or final stage in manufacture and as a means of obtaining the required material mechanical properties, accurate dimensions and surface quality.

In cold rolling, tension is applied to the strip between the stands and between the final coiler and the last stand. Strip tension reduces the rolling force and causes a redistribution of power from one stand to another because each stand provides the tension for the stands lying behind and ahead.

Tension applied so as to pull back on strip entering a stand is termed *back tension* while that applied to strip leaving a stand is termed *front tension.*

The varying sizes of moulds required to produce the slabs are dictated by the size of the rolling mill and the techniques of marrying the finished products to minimise scrap. Individual tapered ingots or slabs vary from 500kg to several metric tonnes in weight, larger continuous castings are produced for certain applications. Traditionally, steel is cast into the form of large tapered ingots approximately 630 mm square, up to 3 metres in length and typically weighing several metric tonnes.

These ingots are transferred from the casting plant to the primary rolling plant in a cold or semi-heated state and left to soak in a furnace for several hours at a temperature of the order of 850° C for copper alloys and 1250° C for steel.

The length of the soak, determined by the ingot size and composition, ensures even heat distribution throughout the ingot section, i.e. no temperature gradients. When ready, the ingot is transferred to the blooming or cogging mill for breakdown rolling.

The general rolling technique is to break down the cast ingot in the mill by a number of passes with progressively increasing reductions, after which the stock is reduced in gauge and increased in length and/or width until the required dimensions are reached.

Strip, produced from cast slabs in the slabbing hot mill, is often subjected to a subsequent cold rolling operation in order to improve the surface finish and increase the dimensional accuracy.

2.3 Production Planning Aspects of Rolling

The production aspects of rolled products involves the methods of *routing*, *operation sequencing* and *rolling practices* that achieve the final product. The production routings and rules of practice have been established on the basis of standard technical specifications, which require strict control over dimensions, alloying elements and mechanical hardness. This calls for chemical analysis, and quality control tests on dimensions and finish.

One major factor by which production methods may change drastically and all previous plans are assessed is the *material yield.* The higher the value of raw material and the cost of the processes involved the more important the yield factor becomes. Therefore, planners with many years experience can juggle the sales orders and marry them in such a way as to achieve the best possible yield while satisfying all other constraints. The case studies in the next chapter will demonstrate this in depth.

The material yield in any process is usually defined as the weight ratio of the finished products to the weight of the input raw material necessary to produce it. A detailed analysis of this concept has been carried out by Husseini (1989).

The product yield expected from the rolling mills operating with conventional electromechanical screwdown gears, plus the gauge tolerances demanded of the rolled products, make the task of production planners a complex one, especially, the calculations of gauge scheduling and dimensional control during hot rolling of slabs.

Cyclic gauge variation seemed the most common problem at the two companies studied. The gauge variations result from several mill roll and bearing problems: backup roll journals may be eccentric with respect to the roll body or to each other; backup or work rolls may be oval due to regrinding of rolls; and journal or roll body compliance may be nonuniform around their peripheries.

Whatever the source, the resulting cyclic gauge variations are commonly causing considerable damage and in some cases slabs have to be re-rolled to achieve a better gauge control. This problem has contributed towards the development of many control strategies (i.e. use of automatic gaugemeters, use of hydraulics, etc.) to reduce these problems.

Various attempts have been made to automate the functions of rolling mills and their performances. These are:

i) Mill speed
ii) Roll-gap setting
iii) Mill scheduling
iv) Roller table speed

One such mill with a fully computerised system which controls the above functions is the installation of Rautraakki Plate Mill. This has been described by Dendle (1978). The speed requirement necessary for the operation of the automatic gauge control system is established by the use of an Interdata 7/16 computer with a high speed processor and 64K bytes of core memory. The mill scheduling is accomplished through separate stored tables for both broadside and straight rolling. Various matrix tables containing ranges of drafts applicable to the full range of slab sizes and desired plate or sheet sizes are set up, giving an overall store of more than a million separate schedules (Dendle).

However, the above automation is confined to the rolling mill control room and has no link to any host computer. The mill operator sets the slab size, finished plate or sheet width and thickness, and the computer then selects the appropriate rolling schedule. Individual gap settings are set with reference to a calculated roll force, and from this information the computer adjusts either the screws or the cylinder position. The mill speed is controlled by the computer with reference to the plate length. This, therefore, ensures optimisation of the time taken for mill acceleration, deceleration, reversal and the appropriate speed of the roller tables.

2.3.1 Rolled Products

Blooms and large slabs are the products of the primary rolling process to which an ingot is subjected. Their uses and sizes are given in the chart. Figure 2.2.

It is rather interesting to note that the maximum available size of cold-rolled products, mainly strip, increased from about 200 mm in 1893 up to 900 mm by 1925, and by 1933 it was possible to go up to 2250 mm. Minimum gauge figures were of the order of 1.25 mm at 250 mm wide increasing to 2.25 at 500 mm wide in 1923, and today it is possible to produce tinplate gauges of 0.1 mm at 900 mm wide and 0.4 mm at 2250 mm wide

Edwards (1978).

However, today, the emphasis is not on the basis of product variation or range of products, but more on increased throughput and improvement in terms of gauge variation, shape, surface finish and high yield.

With high throughput and the high degree of product variation come the problems of handling and storage. Consequently, sizes of overhead cranes, forklift trucks, rolling tables and a whole host of other handling equipment become part of the manufacturing tools. With these, of course, come the regulations and safety aspects which in turn bring constraints and restrictions which the process planners have to consider during planning.

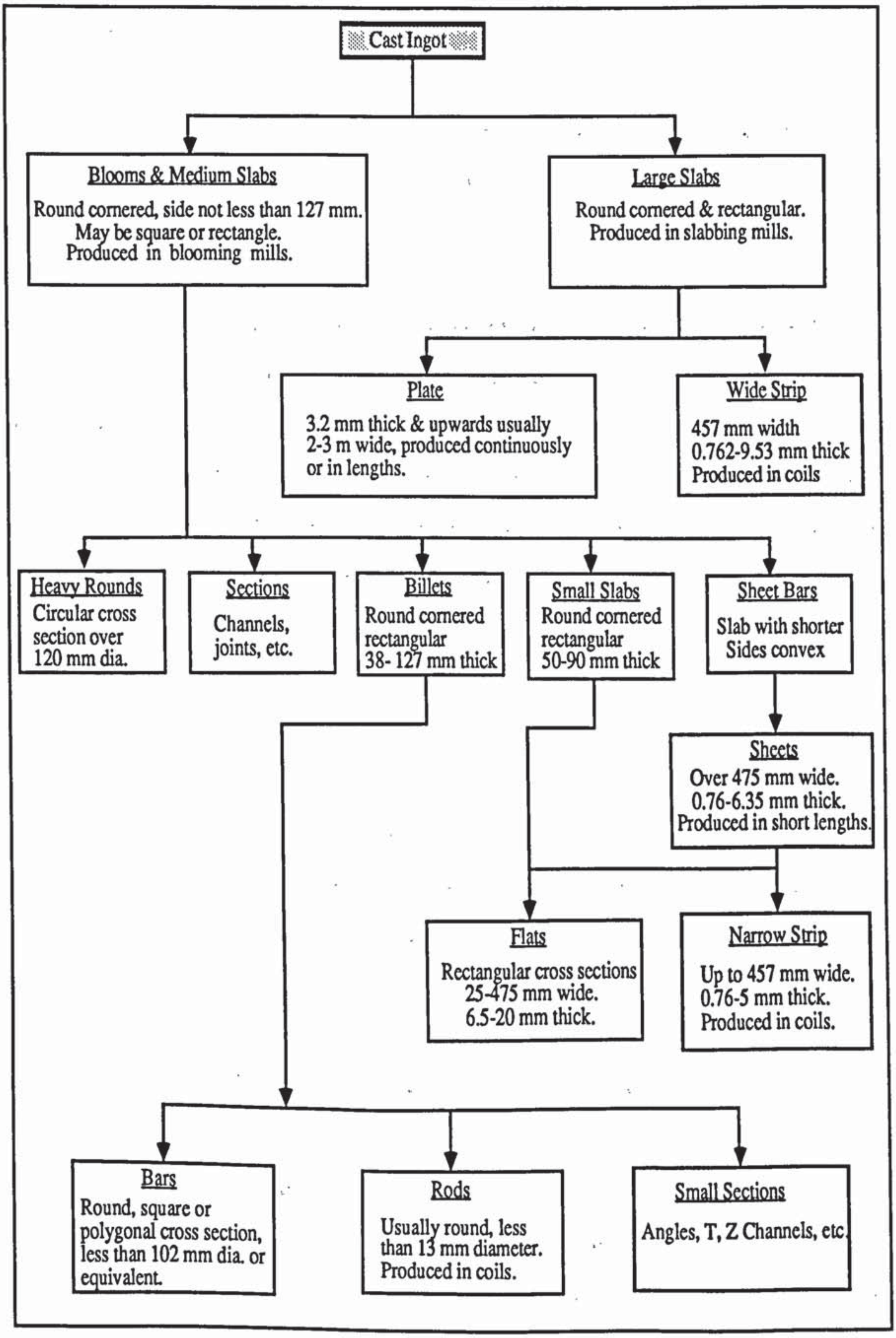


Figure 2.2 The Production Chart of Rolled Products, their Uses and Sizes.

2.4 Copper Slab casting

According to Adlington (1976), modern hot mills require copper slabs with a cross section of 640-1040mm by 127-203mm. Lengths may vary from 2 metres up to maybe 7.5 metres.

Such slabs are mostly cast at a refinery, operated by either a primary copper producer or a secondary metal-recovery company. High-purity electrolytically deposited copper is melted by a gas-fired shaft-type furnace to the continuous or semi-continuous casting machine. Both metal composition and temperature are closely controlled. Water cooled copper or graphite lined moulds are used to make both HC (Half hard Copper) and phosphorous deoxidised copper grades. The castings are then sawn to shorter lengths to suit various hot mill requirements.

2.5 Alloy Slab Casting

Most hot rolling mills have their own associated alloy-slab-casting facilities. Both coreless and channel electric induction furnaces are employed for scrap melting and power ratings up to 2 MVA are not uncommon. A channel melting furnace may have a total capacity of about 20 tonnes, with half that quantity available for transfer to a holding furnace. These holding furnaces can be mounted on rails to provide a collection and delivery facility between the primary melting units and the slab casting machine.

While a small proportion of copper alloys are rolled to finish at 1 metre wide, it is more normal to find a width limitation provided by slabs cast at either 640mm wide or some intermediate widths. Slab thicknesses of these alloys range from 100mm up to about 180mm depending on their relative resistance to deformation at hot rolling temperature.

2.6 Product Definitions

Rolling mills produce a large number of products, but for the purpose of this project, these are categorised into two groups (1) plates and (2) sheets and strips. The two groups involve all the relevant aspects of planning considered in this project.

2.6.1 Plates

Plates are usually defined as flats, 6mm thick upwards and several hundred millimetres wide, produced continuously or in lengths Orr (1960). Plates are also classified into three groups according to how they are produced:

a) *Universal Plates*: These are rolled between horizontal and vertical rolls simultaneously and trimmed on the ends only.

b) *Sheared Plates*: These are rolled between a pair of parallel horizontal rolls and sheared to size on their edges and ends.

c) *Rolled Floorplates*: These are defined as flat hot rolled finished products having a pattern rolled on to their surface in the finishing pass.

2.6.2 Sheets and Strips

The distinction between sheets and strips lies not in the gauge, but in the method of rolling. Sheets are rolled in short lengths, frequently in hand mills, while strips are rolled in coil form, frequently in continuous mills. Strip mills may be classified as *wide* (400-800mm), *medium* (100-400mm) and *narrow* (10-100mm). Wide strips may be sheared into lengths to make sheets.

2.7 Hot Rolling

For the first stage in producing sheet or strip from copper or copper alloy slabs, the slabs are heated to ensure uniform flow of metal, since without this some cracking might occur as pressure is transmitted unevenly through the thick metal. These slabs are pre-heated, often in a walking-beam gas fired furnace, ready for hot rolling. Providing that rolling temperatures are sensibly controlled, then hot-mill routines can be standardised and sometimes even computerised to provide a hot rolled coil or plate products.

The slab ingot is rolled lengthwise, in the hot-mill, into a long slab which is sheared to give slab lengths suitable for rolling into the plates ordered. Due to the high width-to-thickness ratio of the stock involved in rolling these slabs into plates, there is little spread, so the slab width is often insufficient to give the plate width. Therefore, when rolling very wide plates, broadside rolling is used in one of two ways. The slab is rolled lengthwise in the normal way until its length is approximately equal to the required plate width, then it is turned broadside on and rolled to the finished gauge. This is called *length to width rolling* , as the original length becomes the width of the plate and some users of plates feel this is detrimental to its properties for certain uses. If the original slab length is equal to the required plate width, broadside rolling is commenced immediately with no preliminary length rolling. Alternatively, the slab is turned broadside on immediately and rolled until its original width has been elongated to the required width, when it is turned back to its original orientation and the rolling to gauge completed. This is *length to length rolling,* as the original length is still represented by the rolling plate length. With either method, a limited range of slab widths can cover the full range of finished plate widths.

2.8 Cold Rolling

Cold rolling is applied during the finishing stages of production of both strip and sheet and also in the production of very thin materials such as foil. Since most metals work-harden during cold-rolling operations, frequent inter-stage annealing is necessary. This obviously increases the cost of the process and, as far as possible, operations involving the use of hot-rolling are used in preference to cold-rolling processes. However, there are advantages in cold-rolling processes as mentioned below:

- To obtain the necessary combination of strength, hardness and toughness for service. Most steel and non-ferrous materials can be hardened only by cold-rolling.

- To produce a smooth, clean surface finish in the final operation. Hot-rolling generally leaves an oxidised or scaly surface, which necessitates pickling the product in an acid solution.

- To attain greater dimensional accuracy than is possible in hot-rolling processes.

- To improve the machinability of the material by making the surface harder and more brittle.

The rolled coils are cut to size as base supply material for each order. Conversion of the base supply to the finished product is a series of cold rolling operations. The base supply is normally rolled from a gauge range of about 10 mm down to 1.5 mm thickness. The rolling is based on a maximum reduction of about 70% at which annealing will take place and further reductions are therefore possible. The effects of cold rolling are increase in tensile strength, yield strength and elastic limit. The elongation and contraction in area is therefore reduced. The combination of these factors dictates the number of roll passes any sheet can go through to provide the desired hardness.

2.9 Annealing and Cleaning

During production and fabrication, copper and copper alloys may be heated for homogenising, for hot working, reworking and control of final temper, for stress relief, for solution treatment and precipitation hardening. Thereby, stresses are removed and the original ductility of the metal returns. The changes which accompany an annealing process generally occur in three stages; *the relief of stress, recrystalisation* and *grain growth,* definitions of which are given by Higgins (1979).

It is usual to have a number of heating chambers and annealing furnaces with conveyors travelling at various speeds depending on the alloy and its reworking requirements.

Hard-rolled strip may be bright annealed, or annealed and acid cleaned, to provide a soft, lubricant-free strip with a surface roughness of about 0.25-0.5 μm. The cleaning processes are generally categorised as pickling, solvent cleaning, electrolytic cleaning and degreasing. More details of these processes are provided by Higgins (1979).

2.10 Temper Rolling

Intermediate hardnesses are produced by temper rolling of previously annealed strips and the best surface finish can only be obtained by using highly polished two-high mills. Temper rolling is carried out where bulk production of strip with a good surface finish is required by the customer. The cost of temper-rolled strips with a superlative surface finish is often quite considerable when related to the standard commercial finish.

Reduced polishing costs in the customer's factory may, of course, more than justify such a premium, but there are many cases where the coarser finish is more suitable. One such example is when the plate is required for deep-drawing and pressing operations, in this case, coarser finishes promote improved lubrication.

2.11 Temper Annealing

It is also possible to produce strips of certain alloys to intermediate hardnesses by temper annealing from the fully hard condition. Due to its softening characteristics, copper cannot be annealed to temper on a commercial basis. Usually, batch furnaces and continuous furnaces are employed for this purpose.

2.12 Flattening, Shearing and Slitting

Shearing is a process used to bring rolled flat products to final width and length by cutting on a press shear. The operation is performed when it is desired to produce metal with accurately straight edges and with ends at right angles to the edges.

Material with sheared edges is first flattened on an automatic blanking machine, then it is brought to final width and length by press shearing. The automatic machines used to flatten and cut the rolled material have an upper and a lower set of rolls staggered so that when the sheet or strip passes through, the metal is flexed repeatedly but is not reduced in thickness during this operation. An automatic shear built into the machine then cuts the strip to the pre-determined length. The entire operation offers a means of flattening and cutting rolled products rapidly and efficiently without the danger of scratching the high quality surface. Annealed or lightly worked material may or may not be stretcher-straightened or patent-levelled to remove buckles, ripples or wavy edges after being flattened and cut to length.

When it is desired to produce metal with accurately straight edges and with ends at right angles to the edges, the material is press-sheared and the product is classified as *square sheared metal.*

A slitting process is used to bring rolled flat products to final width by cutting them on a rotary slitter. The operation serves the dual function of eliminating imperfect edges and of bringing the rolled flat material to the desired width. The machines used for slitting consist of rotary cutting shears mounted on two parallel shafts that are rotated in opposite directions and are driven by a motor through suitable reduction gearing. The rotary cutters are from about 100mm to 330mm in diameter, and from 3mm to 18mm thick. When numerous pairs of properly spaced cutting shears are employed, several different widths of strip or several strips of the same width can be slit simultaneously from the same roll. A coiling mechanism at the exit of the slitter is used for rewinding the slit strip. The maximum thickness that can be slit on modern slitting machines in the copper and brass industry is about 6mm, however, this value depends on the capacity of the slitter, the number of cuts per roll, the alloy, and the temper of the rolled material. Rolled flat products more than 6mm thick must be sheared or sawed to width.

2.13 Blanking, Cropping and Piercing

Blanking is a process used to cut or to punch flat pieces of desired shape from rolled flat products, and offers a rapid method of cutting identical flat pieces. The blanking operation is performed through a shearing action applied by means of a punch and blanking die, each of which has a sharp edge and clearance in proportion to the thickness of metal cut. Although blanking falls in the same process category, as shearing, it is defined by Lloyd (1986) as "... a shearing operation of closed outline where the metal within the outline is utilised".

Fine-blanking is defined as a blanking operation in which, by means of specially designed tools, dimensional accuracy and good surface texture are imparted to the sheared edge (so that any further treatment of the cut edge is virtually eliminated) Lloyd (1986).

A further method of producing blanks is by cropping, often combined with piercing. Cropping and cropping-and-piercing operations are used when the material from which a component is being made is the width of the finished product. For articles with parallel sides the cost of blanking tools is eliminated and scrap is reduced to a minimum. Such cropping tools are described as being easy to operate and not expensive to manufacture. Tools for the combined operations can effect a considerable saving in both tool and operating costs over the alternative method of blanking and piercing, whether used simultaneously or separately.

Again, the well documented handbook by Lloyd (1986) describes the above operations in depth.

2.14 Summary

In this chapter, a brief discussion of the working of metals and their production aspects are provided. An attempt is made in Section 2.3, to describe the complex problems of gauge control and rolling speeds for the purpose of rolling schedules. The section also provides a background to rolling products, their uses and sizes. These are summarised in Figures 2.1 and 2.2.

Sections 2.4 to 2.6, explain the processes of the copper slab castings, their sizes and manufacturing methods. Product definitions of plate, sheet and strips are also provided.

Two further Sections 2.7 and 2.8, discuss the implications of hot and cold rolling and their definitions. Some of the terms used in the hot and cold rolling of metals are also discussed.

Finally, Sections 2.9 to 2.13 provide the reader with a general background to some of the terms and methods of annealing, flattening, shearing and cleaning processes involved in this environment.

CHAPTER 3

CASE STUDIES AND GOAL RECOGNITION

3.1 Introduction

The objective of this chapter is to highlight the general requirements of the rolling mill industry. Having studied two major companies in this field, their functional requirements and development of manufacturing techniques in terms of computerisation are identified.

Manual production planning and control systems, although not different in principle from computerised systems, do suffer from natural time lag. As observed by Drucker (1970), time is a unique resource which has a totally inelastic supply and is totally irreplaceable. Sabaratnam (1982) concludes that the main purpose of using computers in production control applications is to compensate for these inherent limitations of time.

The role played by a computer in an organisation is seen by many as acting as an information depot and a calculating machine. According to this view, its function is to capture and process data relating to a large amount of transactions which continuously take place within the company. The processed data is usually presented to management in the form of reports on past events and plans for action. Although computers do not make important decisions in manufacturing planning and control functions of a company, they can assist in providing timely and accurate information based on the much valued data base system.

Most production control decisions can usually be arrived at best by combining the information storage capacity and processing power of the computer with the human intuition of the production planner/controller. The use of a computer makes it possible to plan production activities in detail, quickly and more easily than would be possible using manual methods. It could also provide an additional facility to management to evaluate the various production planning and control methods and improve current practices if possible.

Burbridge (1978) classifies production systems into three categories as viewed from the end product (Figure 3.1):

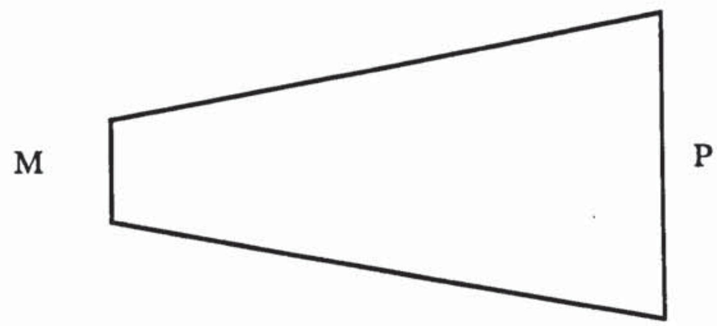
1) implosive systems;
2) process systems;
3) explosive systems.

The foundry and rolling mill environment may be categorised as an implosive system, where one basic raw material (metal) is used to manufacture different products. As observed by Watts (1978), there are three main features which distinguish production control in this environment from those of general engineering type industries.

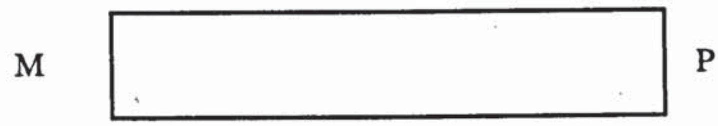
a) The production line is unidirectional, i.e. operations are carried out in a fixed sequence through the various work centres.

b) There is considerable variation in the work content at various work centres due to casting shape, slab size, complexity, metal, inspection requirement and batch size.

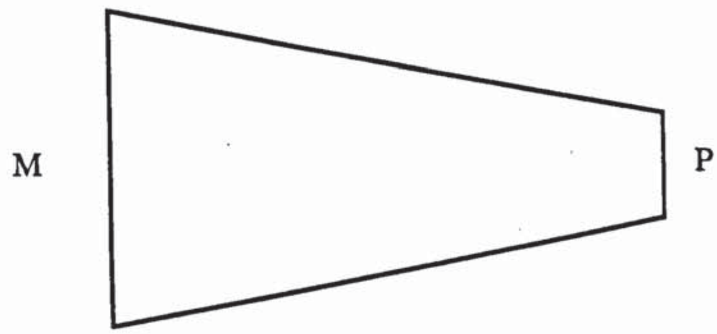
c) There is uncertainty of yield due to rejects or order mix.



3.1 a) Implosive (e.g. Rolling Mills, Foundry)



3.1 b) Process (e.g. Chemical, Cement)



3.1 c) Explosive (e.g. Assembly)

M = Input of materials - Number of Varieties

P = Output of products - Number of varieties

<u>Figure 3.1 (a-c) Types of Production Systems</u>

Due to the special nature of producing rolled metals or foundry production, the computer software needs to differ from that provided in the general-purpose production control packages, which are commonly available. The existing packages in the market have had very little impact in this environment and their success has been limited. There are two major reasons for this failure. Firstly, the systems are based on expensive mainframe and mini computers. The majority of metal forming companies are small in size (Table 3.1) and their turnover does not justify expenditure in hardware and software in excess of £200,000. Even those larger companies where the hardware is already installed find it difficult to justify the cost of software. Secondly, these systems are sophisticated in design but not sufficiently flexible in their function to cater for the individual rolling mill company. These systems are usually more orientated towards commercial data processing activities (such as payroll and accounting, i.e. maintaining various ledgers and invoicing) than production control. Therefore a very high proportion of the rolling mill companies (even larger companies with mainframe and satellite computers) still use manual systems.

The case studies enabled the author to identify three major areas urgently needing attention and to see that much research was required to formalise the manufacturing techniques which should be used by these companies. These areas are:

A) Computer Aided Process Planning Functions (CAPP)

B) Computer Aided Material Optimisation (CAMO)

C) Computer Integrated Manufacturing and Management System (CIMS)

As explained in Chapter 1, each area represents a mammoth task in itself and requires much research and financial backing to develop comprehensive systems from the beginning to the end. However, it is the concern of the later chapters of this thesis to provide solutions to Parts B and C above and provide an overview of Part A in this chapter.

3.2 Case Study at Company A

At the beginning of this research, this case study was instigated by Company A at the request of their managing director. The company was experiencing difficulties in meeting due dates, keeping track records of manufacturing processes, achieving high yields and maintaining the order book. Some of these problems were later found to be due to the lack of knowledge and experience of staff (mainly the production planner), lack of communication with the production manager and the increase in tension between staff as a result of the company going into the red and redundancy actions that had followed resulting in lack of confidence between staff.

3.2.1 Background

The company manufactures copper and copper alloy plates, sheets and circles. It has one of the largest rolling mills in the UK, producing plates up to 300mm in gauge, 3500mm width and 6500mm in length. Furthermore, these plates were being tested to MoD specification and tested using ultrasonic techniques. The orders were not generally repeat orders and very small in quantities, e.g. plates from 1 to 10 maximum and sheets or circles up to 100.

The company was divided into three units;

1) Foundry (or Refinery)
2) Rolling Mill
3) Tube Drawing Mill

The foundry employed 33 people, the rolling mill 27 people, the tube mill 86 people and 35 people as members of staff including the metallurgy department, inspection and despatch. In all, including roll grinders and security, there were about 200 people employed, with a turnover of about £20m. The company was in the red and at the time had made 50 people redundant.

3.2.2 Manufacturing Activities

Here only the activities within the rolling mill department are considered. The sequence of operations and manufacturing processes are illustrated by Figure 3.2.

When sales orders are received, they are accumulated in an order book and processed the next day. The planner, with the help of the production manager tries to marry orders and work back through the stages of manufacturing processes, adding the allowable tolerances and minimum scrap required for the operators to achieve the desired shapes at each stage. Finally, having worked back to the first stage and found the volume of metal required for each batch, the size of the mould is calculated for the refinery.

The combination of mixing and marrying of the orders is a mammoth task in itself and it takes the production manager and planner at least three hours or half a day's work to achieve satisfactory slab sizes. Each time a mould size is calculated, the nearest size must be selected since there is only a limited number of moulds available in the foundry.

Having determined the slab requirements, these are passed on to the casting shop for their planning board. Due to lack of sufficient orders and volume of metal, some melting furnaces had to be switched on only once a week, or sometimes twice a week, for economic reasons. This was one of the main factors of failing to meet the due dates.

The foundry was very old, primitive and hazardous to work in. While this work was being carried out it was reported that one or two furnaces had collapsed into the molten metal. The company was very fortunate not to have lost any lives as result of this accident. However, many thousands of pounds were lost due to rework of molten metal and rebuilding of the furnace.

Once the slabs are cast they are transferred to the rolling mill shop, where the top and bottoms are machined for cleaning purposes and to achieve uniform surface finish. They are then placed into furnaces to bring the temperature above the recrystallisation point for hot working.

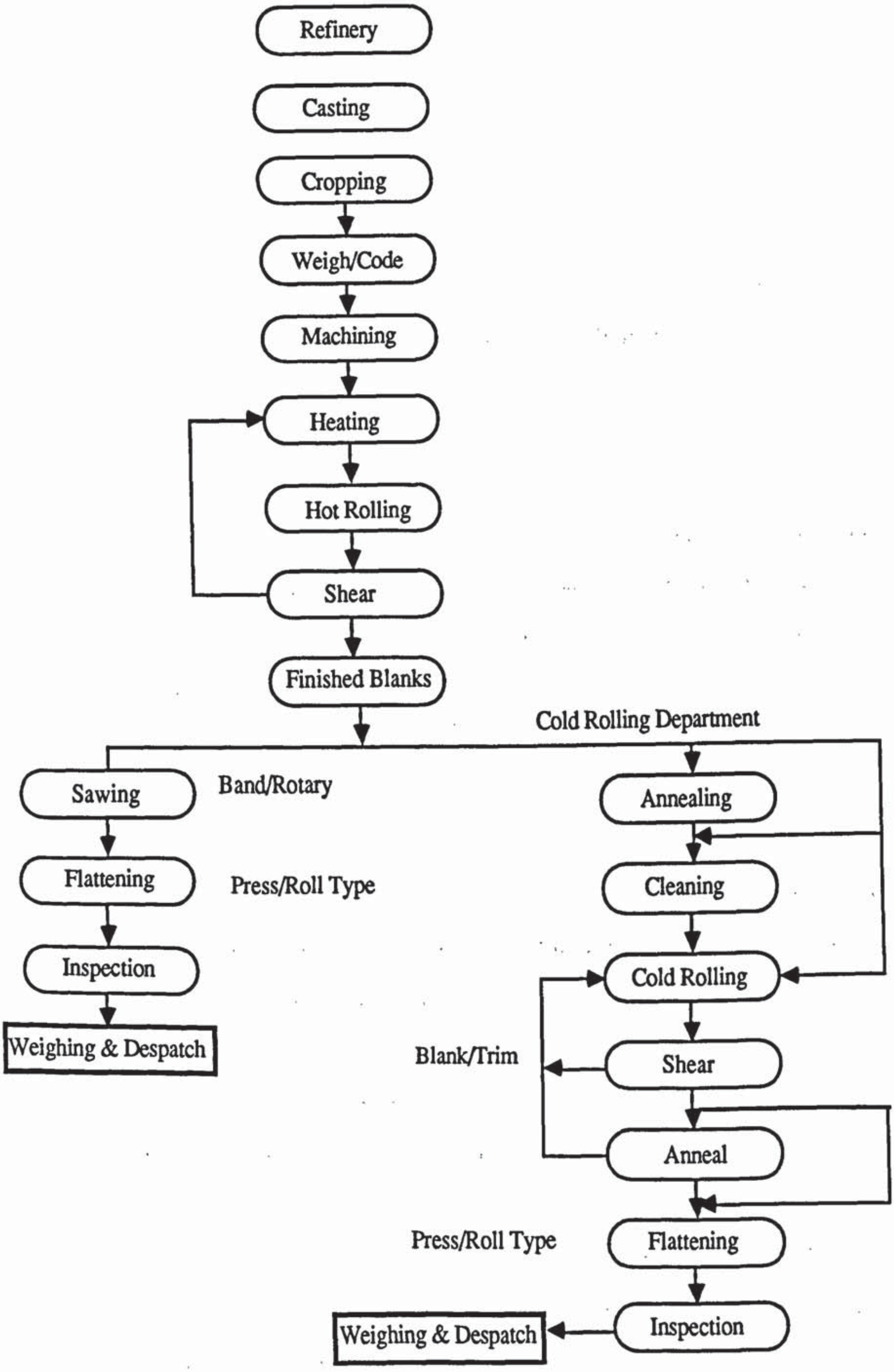


Figure 3.2 An Example of The Process Flow Chart in the Rolling Mill Industry.

The slabs are then rolled and either sent straight to the customers or rolled to a pre-determined gauge for further trimming, cutting to size and pre-heating again for further rolling. In most cases the slabs at this stage these were cut to standard sizes known as "uncut blanks". The uncut blanks may or may not cover more than one customer order, hence they were either re-heated for further hot rolling or transferred to the cold rolling section for further work. Husseini (1989) has described these processes and the manual methods of process planning at this company in greater detail.

3.2.3 Manufacturing Problems

Almost all the aspects of manufacturing, sales, purchasing of raw material and running of the organisation needed to be reviewed. Since the author and his colleague were assigned the task of studying manufacturing activities, the following problems were identified during our stay with the company.

1. Lack of process planning consistency.

2. Reference to previous process plans: although the plans were archived, they were never referred to for future planning. One of the reasons for abandoning the previous process plans was that no mix of orders was ever repeated.

3. No costings were associated with any plans to determine the validity of the judgement of the process planners or production manager.

4. No long, medium or short term plans produced. Although it is difficult to forecast in this environment, it was evident that no analysis was being done in terms of repeated orders.

5. Scheduling was left to the foremen and the planner had very little influence in their judgement. Since the company was dependent on the skill of the mill operators, they could dictate almost anything.

6. Lack of integration, team work and communication between departments. The distance between departments and the conditions of work were not suitable for improving communication.

7. Low material yield: everyone, including the managing director, seemed to appreciate this and show concern. The best yield achieved seemed to be around 60% and to reach this would have made most people proud.

During our study, several plans were examined to determine the true material yield, since this was not being calculated by the planner. In all cases the yield was well short of 60% and values around 35% were not untypical.

With tension between the members of staff, concern from the managing director and worry of further redundancies, although the author and his colleague were welcomed by the top management, they were not treated well by the planning department, foremen and shopfloor operators. Therefore, another company (Company B) was approached and amazingly, they were so enthusiastic about the objectives of the research that in a very short time the research was continuing in full swing at this company.

In late 1987, eighteen months after our departure, Company A closed its rolling mill department and in late 1988 the company went into liquidation.

3.3 Case Study at Company B

The fundamental structure of this company seemed to be sound and although it was a subsidiary of a much larger group, it had its own management structure and a budget to run its own affairs. The order book was full and the main orders were from within the UK with a small portion for abroad. The margins were very tight and the company was subject to fierce competition from abroad. The management expressed their concern in terms of reducing operating costs and improving operator productivity starting from planning down to shop floor documentation, inspection and despatch.

The company was linked to the parent company's IBM System 38 computers and was being billed for their computing requirements. The payroll and accounts were the only areas which were fully computerised. The manufacturing side of the business had a very small share of the computing services.

3.3.1 Manufacturing Operations

The Rolled Metal division had three main sections

1) Foundry (or refinery);
2) Strip Mill Division;
3) Sheet Mill Division.

3.3.1.1 The Foundry

The refinery was mainly for the production of a low quality slabs,where scrap produced from imported slabs was being melted down and recycled. The contribution from the refinery was only 15 to 20% of the company's sheet and strip output. Orders were also being taken from customers requiring specialised castings.

3.3.1.2 The Strip Mill

The strip mill division had about 120 employees with an output of around 7,500 metric tonnes in weight of finished goods a year. The output was all in coil form, with each coil weighing between 350K to 600K. The widths of the strips produced were from 5mm upto 635mm with various gauges from 0.17mm to 7.6mm, although higher gauges were possible on specific requests.

The strip mill had one hot rolling mill with several cold rolling mills, some in tandem. Strips with very low gauge requirements were sent to the rolling mills with tandem arrangements, where the coil would be passed through several mills to achieve the desired gauge. The last mill had shearing blades attached to the mill to trim the edges and produce uniform width.

Some gauges were standardised in the strip mill, where, if there was a shortage of orders at any time or some unknown intermediary operation was to take place, standard gauges were employed so that they would be readily used for further reduction by the strip mill, other customers or the sheet mill.

3.3.1.3 The Sheet Mill

As the name implies, the sheet mill mainly produced sheets, although large quantities of circles were being produced occasionally. This mill was situated about two miles away from the strip mill and was using a large proportion of the output of the strip mill. It employed about 90 people. It consisted of two sections: hot rolling bay and cold rolling bays.

The hot mill can roll slabs upto 1250mm in width, with a minimum gauge of around 5mm. The main bulk of the coils used with in the cold rolling bays are from 5mm gauge down to 0.2mm. With a maximum width of 1250mm, the length can be cut to the customers' requirements.

Each division (i.e. Strip Mill, Sheet Mill, Hot Bays and Cold Bays) has its own planning and scheduling requirements. During the time of manufacture, the customers make enquiries on numerous occasions regarding the progress of their orders. Therefore, it is the planners job to chase up the works orders and report back to the customer on the status of the orders.

Since orders vary in size (see Table 3.1), it is the planner's main task to marry orders together and provide a mix of orders for planning of the hot mill. Once the planner has enough orders, he is able to mix these and justify the operation of the furnaces for the reheating of the slabs. If a particular order is urgent and there is enough material left from the previous orders (known as "the mill stock"), the process planner will try to use them prior to considering the slab requirements.

TABLE 3.1 The Main Parameters of Typical Daily Orders.

MATERIAL	GRADE	QUANTITY	GAUGE mm	WIDTH mm	LENGTH mm	HARDNESS Hv
COPPER	5	400	2	200	400	50-70
COPPER	5	250	2	500	CIRC.	50-60
COPPER	3	300	2.5	450	800	30-50
COPPER	5	50	3	450	1050	50-70
COPPER	5	25	2	300	1580	60-70
COPPER	2	125	2	800	950	50-70
COPPER	5	10	5	750	1500	50-70

Again yield plays a major role; it is very rare that the planners would proceed with any works order if the yield generally falls below 60% although, on occasion, yields of 50 to 55% had been observed. This part of the planning excercise takes the most time if the planners are to achieve a reasonable mix of orders and it also requires full explanation of details to be attached with the route card. Invariably, the planner also has to spend a long time chasing the works orders to make sure that the foremen and operators understand the logic of the works order.

3.3.2 Manufacturing Problems

Again in this company, problems existed at almost all levels of the division, the main problem being lack of information in a short time. It was therefore obvious that more terminals were needed throughout the company to access the central data bank, this would have meant heavier monthly bills in terms of computer services from the parent company and this deterred the management from commiting itself to increasing the share of the computing power and the number of terminals. Although, the actual costs of the computing requirements were not disclosed, we were led to believe by the data processing manager that the bill amounted to hundreds rather than tens of thousands of pounds.

Therefore, the main problems were identified by studying each department and holding extensive interviews with the key people involved for that department. The main problems were:

1. Estimating - The sales department could not respond to customer enquiries in time and needed up-to-date information regarding existing orders and their status so that they could determine reasonable lead times and cost. Therefore they had to resort mainly to standard lead times and costs which were not always strictly true.

2. Coding and classification of products - This was particularly important since there was no formal method of grouping the company's products. Although attempts were made to input the sheet sizes in the data base, these seemed to have no concept behind them. The data base had no product grouping or any characteristics which would group certain items which had the same criteria.

3. Stock control system - There was no record of the stocks available at any time on the shop floor. The only manual system was that when slabs were used a list containing the quantity was sent to the head office, so that orders could be placed again when stocks of slabs fell below a certain limit.

4. Generation of process plans and route cards - A very limited amount of planning was done at the head office by keeping the standard gauge sizes in the computer and generating a process card for each individual order. Orders were not mixed at this stage and were assumed the sourcing of material from a whole slab. A process plan was retrieved from the system for each order, tailored and saved for future reference. The biggest problem here is the mixing of

orders that are for the strip mill and knowing which orders should be mixed to achieve the optimum number of slabs required by both the sheet mill and the strip mill. The calculations were tedious and differed almost daily, since the daily orders were vastly different.

5. Regeneration of route cards - When the planning department receives the daily orders, they then start generating their own plans and mix the orders together where possible. Strip mill had the most problems since the widths of orders were small compared with the optimum width of the coil (635mm).

6. Manual stock control - Since there was no computerised stock control system, the planners had to rely on their hand-written record books and allocate stocks manually. This was time consuming, frustrating and prone to errors.

7. Process planning -Once the orders have been mixed, the works orders were being generated manually, they were hand-written and archived in a room where they would probably never been accessed again. Yields were causing particularly serious problems and any change in plans had to be carried out manually and the process cards then generated again.

8. Communication - The orders and their route cards were being taken daily to the planning department by motor cycle courier. It was reported to us that on occasions some had been lost and the orders were not completed on time. Any urgent information to and from planning was either taken by courier or telephoned.

These problems occurred in both the strip mill and the sheet mill divisions. The head office, situated two miles away in a separate building, was running the sales, purchasing and accounting side of the business.

The above problems were grouped and categorised into three sections:

A) Computer Aided Process Planning and Estimating (CAPPE)

B) Computer Aided Material Optimisation and Cutting Stock Problem (CAMO/CSP)

C) Computer Aided Production Management System (CAPM)

Each section needed a considerable amount of work, that is why the work was divided between two research theses, Part A being investigated by Husseini (1989) and Parts B and C investigated by the present author.

3.4 Computer Aided Process Planning (CAPP)

In this section a review of the CAPP system, provides the background to what process planning is and why it is needed if a CAPM is to function satisfactorily to provide the entire work force with intelligent and meaningful data.

It is frequently recognised that Computer Aided Process Planning (CAPP) is not just one of the fundamental stages of the manufacturing process from design stage to the production stage, but it is the key factor for integration of CAD and CAM systems - Weill et al. (1982), Eversheim and Schulz (1985) and Logan (1985).

While process planning as defined by Udeze (1984) is "...a decision making function which determines the manufacturing processes, including: operations to be performed and their sequence, which machines and work stations to use, and the operations conditions that transform raw material into planned production", it is generally limited to the design of the product transformation process. Groppetti and Semeraro (1985) state that the design task starts from the drawing of the finished part and goes backwards identifying all the transformation intermediate processes up to the raw material selecting the technological processes and operations, and related tools, machine tools, inspection and assembly machines, fixtures, sequencing and grouping operations, determining all the process parameters and variables in order to satisfy the product and production requirements.

Although meeting the specifications given in the drawing is usually the prime objective of process planning, satisfying the times and cost criteria with respect to the constraints set by company facilities, technology and required quantities can be regarded as secondary objectives. Often, some or most of these constraints are variable, hence the optimum process plan developed can only be valid with respect to the particular conditions considered in making the decisions.

3.4.1 Process Planners

Process planners are *knowledge workers* Tulkoff (1985), dealing exclusively with information to establish the orderly and efficient methods and procedures necessary to manufacture the products, starting from the raw material and going through the various fabricating processes to the final assembly.

Process planners require intimate knowledge of manufacturing processes, e.g. (in the rolling mill industry) machining, rolling, shearing, heat treatment techniques, costing, etc. In addition, familiarity with the production capabilities of the specific plant in which a part is being manufactured is vital.

There are essentially three approaches to accomplish the task of process planning Studel (1982): manual, variant and generative.

3.4.2 Manual Process Planning

This involves examining an engineering part drawing, or in the case of the rolling mill industry, the customer's product drawing or dimensions, and developing manufacturing process plans with instructions based upon knowledge of the process and machine capabilities, tooling, materials, materials source, and shopfloor practices. This method relies heavily upon the experience of planner to develop process plans which are feasible, low cost, and consistent with plans for similar parts. However these activities are highly subjective, labour intensive, time consuming, tedious, and often boring. Furthermore, the task requires well trained personnel who are experienced in manufacturing shop floor practices.

Nevertheless, the manual process planning is often the best approach for small companies with few process plans to generate and low product variety. A good process planner can create process plans which are accurate, fairly consistent and cost effective - provided that the number of parts and processing alternatives are small.

3.4.3 Variant Process Planning

The variant or retrieval approach to process planning is a computer-assisted extension of the manual approach, that is, creating a routing for a new part by recalling, identifying, and retrieving an existing plan for a similar part and making the necessary modifications for the new part. The computer assists by providing an efficient system for data management, retrieval, editing, and high speed printing of process plans.

The retrieval process planning system already in use at Company B assumes that all orders are to be manufactured from a common source (for example, the standard sizes of castings) by rolling them to standard sizes of blanks or coils, and therefore it does not have the necessary information about the other sources of material. The process plans for each individual order are printed by the computer on the process route cards. These are then sent down to the area planning clerk on the site for further editing and material allocation.

It is at this point where the process plans are handled manually and alterations are made by hand. At this stage the system effectively reverts to manual planning and misses all the advantages of a variant system.

However, even if the material sources were available to the variant system, it still does not guarantee to produce the optimum or near-optimum process routes, since it lacks the necessary knowledge-base system and the vital technical and manufacturing judgement of how to mix each individual batch of orders and, as a result, how to generate cutting sequences. Therefore, the need for a fully generative process planning system with the help of Artificial Intelligence techniques, and development of various application programs with a comprehensive data base management system becomes essential.

3.4.4 Generative Process Planning

While variant process planning usually provides the user with a process plan that may require editing or assembly before it reaches the shop floor, generative process planning needs little input from the user. Sometimes only a description of the part is needed to generate a plan to produce it.

A generative process planning system essentially consists of two major components Studel (1982) and Harvey (1983). Firstly, the geometry based scheme for translating physical features and engineering drawing specifications into computer interpretable data. In such a system, the geometry of the part is recognised and the system generates the plan based on its understanding of the manufacturing methods needed to produce such a geometry.

The second component is the software, comprised of decision logic, formulae, and technological algorithms, to compare the part geometry requirements to manufacturing capabilities and availabilities. This involves determining the appropriate processing operations, selecting the machine for each operation, determining rolling schedule planning or other operation details subject to available tooling and fixturing, and calculating the set-up and cycle times for each operation.

For this reason, the following sections are mainly devoted to the geometrical aspects of process planning. This is followed by an introduction to the cutting stock problem, and how the geometry of required shapes may be altered at every stage of operation to minimise the waste.

3.4.5 Scope of Process Planning Functions

The three basic requirements of process planning functions are as follows:

(1) Process design;
(2) Operation design;
(3) Time estimating.

3.4.5.1 Process Design

Following Hitomi (1979), we consider process design as "... a macroscopic decision making of an overall process route for converting the raw material into a product".

Essentially, process design consists of :

(i)	Process selection
(ii)	Process sequencing
(iii)	Work-station selection

Process selection involves the decision to adopt a particular process. The selection of an appropriate manufacturing process is not always a simple task. Very often, alternative processes exist for accomplishing a given manufacturing task, and a decision has to be made as to which process to adopt, with respect to some production and economic constraints such as :

Type of material

Batch size

Work size

Machine constraints

Dimensional accuracy

Appearance and surface finish

Cost of raw material, defects and scrap rate

Process complexity

Subsequent processes

In some situations, the constraints imposed by the available facilities on the shopfloor leave the planner with only one or two process options, hence reducing the process selection task to using the only available process.

The process sequencing, as the name implies, mainly consists of arranging the selected processes in a pre-determined sequence, to establish the process route (or work-flow) for converting the raw material into a finished product.

The work-station selection is the process where, from a list of available machines, the planner decides which work-stations to use for the process route in question, while :

Satisfying due dates

Mixing common routes for economic advantages

Not exceeding machine capacities

Balancing work -flow

Improving bottle necks

Utilising man power available

Generally, all the three aspects of process design outlined above are inter-related and are in some cases decided simultaneously.

3.4.5.2 Operation Design

Operation design is defined as microscopic decision making of individual operations contained in the process route Hitomi (1979).

It is mainly concerned with the detailed decisions of production implementation (i.e. the work content of each operation and the method of performing it). The operations work content consists of several steps (loading, machining, unloading) required to accomplish the operation, while the method of operation specifies the man-machine interaction required for each step.

The process design and operation design functions together generate the sequence of operations required to convert raw materials to finished products.

3.4.5.3 Time Estimating

The time estimating function of process planning establishes the standard time for each operation (i.e. total time in which an operation must be completed at standard performance). This is usually a summation of the observed time values of the human elements of an operation. Determination of these standard times is accomplished using techniques of motion and time studies.

The sequence of operation generated by the process and operation design functions, together with the operations standard times generated by the time estimating function, form the process plan for converting the given raw material into the specified finished product. Such process plans are usually listed in documents that are often referred to as *process plan sheets* (or operation process chart). The information can then be used to form the basis for the scheduling methods and cost estimation.

3.4.6 The Proposed Computerised Process Planning at Company B

Having studied the concept of CAPP and the different methods and techniques involved, a summary of the concept of a generative CAPP system required in this environment was developed by Husseini (1989) and is briefly explained below.

Husseini rightly argues that variant CAPP approach does not provide long-term answers to the problems in this industry, since an experienced planner is still required to supervise the generation of new plans, modify and mix the orders on almost daily basis. Therefore, the system falls short of considering the dynamic nature of the problem and providing solutions with the manufacturing constraints and parameters imposed. Therefore he argues "Through the study of the literature of process planning approaches generative CAPP approach was found potentially a better alternative in relation to our particular planning problems" As a result, the generative process planning system developed is based on five main steps.

STEP 1 - Evaluates all process planning routes possible using "the backward search method" to arrive at the possible starting points of the end product. All the routes and the costs associated at each process or stage are stored in a file.

STEP 2 - This function is designed to examine the material availability for each route. At this point the routes which produce low yield or higher costs are eliminated.

STEP 3 - Calculates the overall cost of all possible routes feasible so far. It also determines the planning details such as operations, machines, men, ...etc.

STEP 4 - Searches for the most cost effective routes.

STEP 5 - Produces the process plans, hence it is designed to interact with the input from the planner in terms of minor adjustments prior to the production of hard copies for launching the works order.

The CAPP system that produces the single plans is named SCAPP. When the system was tested and the results produced were compared with the company's own plans 80% of the results were very close to each other, 8% of the results produced by SCAPP were superior, about 10% were different but costwise similar and only 2% of the results were inferior.

However, the main problems lie not only in the production of process plans for single orders but mixing and marrying smaller orders to produce much higher yields and lower operating costs. Therefore, a second CAPP system called "BCAPP" was developed by Husseini.

The concept of BCAPP was based on three phases:

PHASE 1 - As STEP 1 of SCAPP with additional features such as available standard material, i.e. standard gauges, widths, yield and other related data.

PHASE 2 - All orders are grouped together according to the routes, theoretical gauge and the optimisation criteria. The grouping is based on a method known as the *simple linkage technique of cluster analysis.*

PHASE 3 - At this stage the base material which is selected for several orders with various dimensions and this is the basis of grouping orders which fall in the same group. Their routes and operations recorded and plans produced.

When BCAPP was tested on live data from the company, results indicated considerable improvement over their manual or retrieval system. Husseini indicates that "It showed over 5% increase in material yield and improvement of nearly 4% in the route cost for the batch".

However, although the SCAPP and BCAPP were satisfactory, they do not address the problems of nesting or optimising a set of orders from a given inventory. The generative CAPP system developed was mainly aimed at the planning problems of the sheet mill. The strip mill however, requires the optimisation techniques for solving the *Cutting Stock Problem (CSP)* and *nesting* which are the main subject of this thesis.

In the next section, various versions of CSP are considered and the solutions to these problems are discussed in Chapters 7, 8 and 9.

3.5 Computer Aided Material Optimisation (CAMO)

Broadly speaking the Cutting Stock Problem is to use a minimum of material to fill as many orders as possible without generating any scrap. In practice, this is almost impossible and therefore some means of producing ideal shapes and optimum cutting patterns are required to minimise the scrap and hence improve yield, which indirectly contributes towards keeping costs down. This requires development of a computer system capable of generating intelligent cutting patterns according to the manufacturing capacity and production rules. This complex task therefore is called *Computer Aided Material Optimisation (CAMO.)*. The operational logic of this module should be similar to that of MRP II (Material Resource Planning), where once orders are entered in the system and the MRP is activated, the system analyses the existing works orders, material availability and production capacity, it then provides the user with a listing of suggested purchase and works orders with the recommended due dates and quantities. This, of course, is done by exploding down the Bill Of Material (BOM) or the product structure.

CAMO however, would use the techniques for solving CSP to generate a cutting pattern for a batch of orders and keep examining the stock on hand. If the standard stocks which are not available at the time but can be ordered are given the concept of phantom stocks, CAMO like MRP can use these items and suggest the total requirements at the end of the run.

One main advantage that the logic of CAMO has over the MRP logic is that MRP will consider the parts within the BOM and its alternatives, but CAMO will reject stocks that have high costs and searches for stocks with the lowest cost value. Since the solution to

CSP considered in Chapter 9 is based on minimising the cost of the cutting pattern, CAMO will then use the selected stocks for further allocation.

The CSP solution is based on the cost of the stock in use and indirectly, this is determined by the amount of scrap being generated and the value added. With reference to Figure 3.3 the various types of scrap in this environment is shown.

The edge trims or *unavoidable* production losses, as the term implies, cannot be avoided and will invariably occur whenever a blank sheet is broken down into many smaller pieces. These losses are greatest when the stock is being hot rolled.

The smaller pieces which are impractical to handle or not feasible to hold back as mill stock are immediately returned with the edge trims for recycling. A poor plan will result in the generation of this type of scrap.

Blanks produced for mill stock have several implications:

i) They result in a poor order batch yield when the batch is being calculated. In many cases this may be acceptable, since the planner may be expecting other orders which use the same or similar base material. If so, most mill stock materials will generally produce high yields.

ii) If the slabs, from which several batches of orders were satisfied, were to be reconciled and the total yield calculated, a yield lower than the individual order batch may result.

iii) If mill stock is not managed well, after a certain period the holding costs and management of these may prohibit the planner from keeping mill stocks and he starts increasing the sizes of Type (b) scrap in order to reduce the mill stock.

iv) When an order is received for in-house slabs, the first place the company looks for raw material is the mill stock. If the levels are high, the planner will inevitably let good size mill stock go for recycling, hence again converting the mill stock to Type (b) scrap (see figure 3.3).

The main objective of CAMO will therefore be to analyse the orders received from either the CAPP system or the Sales Order Processing (SOP) system where the parameters considered are:

a) Due Date
b) Grade Specification (alloy, surface quality, temper, etc.)
c) Dimensions (gauge, width, length)
d) Quantity

CAMO will identify the cutting operations within the process routes of each individual order and generate cutting patterns which will optimise material yield. The next step is to mix the orders and therefore the cutting patterns to achieve an overall yield which satisfies the requirements of all the orders, manufacturing rules and machine capacity.

Orders may be classified into two groups:

a) Standard Orders
b) Non-standard Orders.

3.5.1 Standard Orders

The standard orders are processed individually since one of the main criteria that requires an order to be standard is the yield. The process routes and the associated cutting pattern for these orders will be generated. They will then be stored in the computer for future retrieval. These orders may include the following characteristics:

- Common Dimensions (gauge, width, length)
- Large Quantities (200 plus)
- High Yield (utilisation of whole slabs)
- High Repeatability (regularity of orders by the same customer)
- Standard Routing

3.5.2 Non-Standard Orders

These orders may be processed as a batch by mixing them together according to the order mix calculations and rules. They may then be compared with the standard orders already held in the data base system. If a match is found, it is retrieved and used as a standard practice. However if a suitable match is not found, CAMO will generate an appropriate cutting pattern and CAPP provides the manufacturing routing for the batch.

The non-standard orders generally will bear the following features:

- Awkward Dimensions
- Low Quantities
- Low Material Yield* (fractions of slabs having to be used)
- Low Repeatability
- Specialised Routings

* Although small orders may be categorised as non-standard, in most cases they may be used to utilise the remains of the larger orders to achieve overall high yield.

Procedures and manufacturing rules will be set up to establish methods by which orders may be mixed and arranged in standard formats. One possible method of establishing these rules is:

<u>Due Dates</u> - Within any problem, highest priority is given to the orders with the earliest dates.

<u>Surface Quality</u> - If two orders have the same due date, highest priority is given to orders with exposed surfaces.

<u>Grade Quality</u> -If two orders have the same due date and surface requirements, highest priority is given to the one that has the highest quality requirements.

The orders with the instructions of the cutting patterns will be sent to the CAPP module for the final detailed process planning.

3.6 Summary

Two companies were thoroughly investigated prior to the development of any systems. Company A had fundamental problems of management and standard methods of production. The recommendation for this company would have been for a larger company involved in the same line of activity to take over and turn around the entire operation of the company, to introduce some off-the-shelf packages to carry out the accounting, sales and purchasing and generally implement good basic manufacturing practice. Once the company is on its feet, then it would be appropriate to introduce research and development and expand the computerisation to the rest of the company.

Company B was far more advanced than Company A and therefore, the manufacturing problems were recognised by both the management and operators. The attitude of staff and operators at Company B was more relaxed. During the course of this work the staff showed great enthusiasm towards computerisation and their cooperation in terms of input and data analysis was invaluable.

From the case studies and numerous visits to other rolling mill companies, it seemed evident that the development of a fully integrated manufacturing system is necessary to overcome the problems existing at all levels of manufacturing. Certainly, the development of a manufacturing management system was vital to cope with most of the problems at Company B.

At the core of the manufacturing management system lies the development of a computerised process planning function. This was successfully carried out by Husseini.

For the process planners to generate complex mixing and batching of smaller orders and for the planning of the sheet mill at Company B, the CAPP system was not sufficient and needed further enhancements. These were the generation of the CSP module and the development of CAMO.

In the following three chapters, a proposal for the development of a Computer Aided Production Management system (CAPM) is presented and Chapters 7, 8 and 9 develop the background and the method of generating cutting patterns for achieving maximum yield using CAMO.

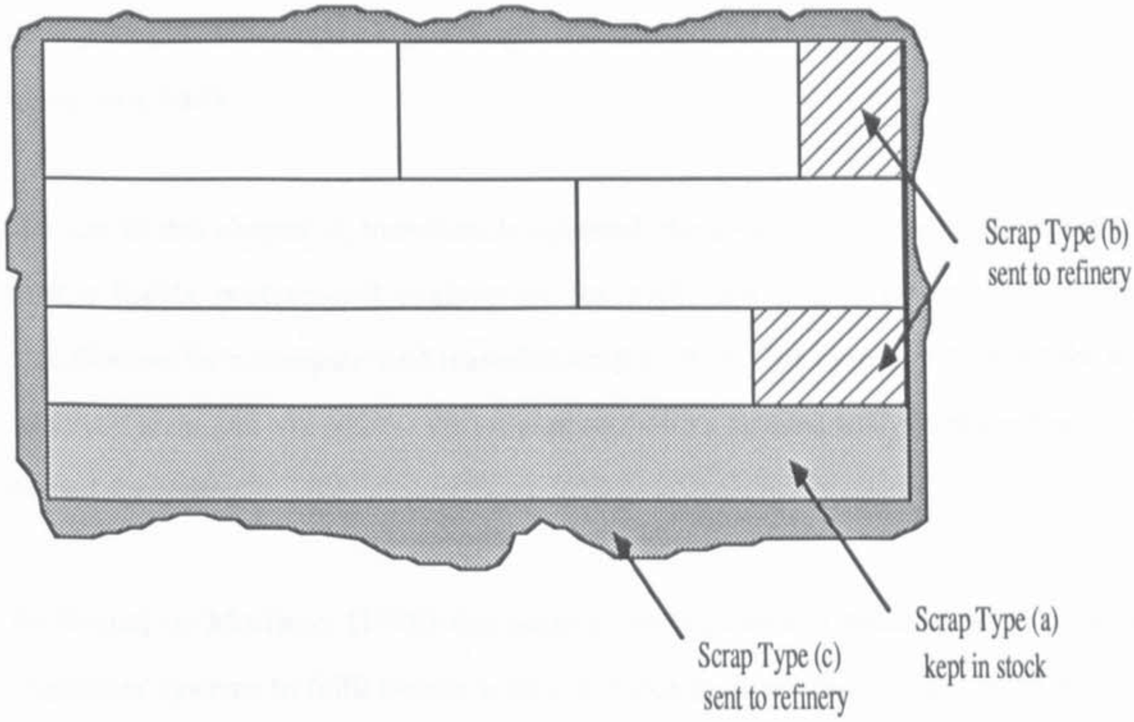


Figure 3.3 Types of Scrap

CHAPTER 4

SYSTEMS ANALYSIS AND CONCEPTION

4.1 Introduction

Having studied the two companies and visited very many others in the rolled metal industry, it became apparent that there were very few systems that could have even satisfied parts of the requirements of this sector. Hence the design and development of a complete CAPM seemed a logical step towards solving the problems in this industry on a long-term basis.

The aim of this chapter is, therefore, to approach the problem in a systematic manner, from a highly professional engineering point of view and to propose an outline specification for a computerised manufacturing system. The system will highlight the important areas and will address the programmer with a detailed analysis of the functions needed for coding.

According to Morrison (1968) the purpose of engineering practice is to provide 'ingenious' systems to fulfil human needs. For this purpose, the engineer must make a selection from all available resources, of what seems to be the most suitable at a glance.

The concept of design and development of the model presented in this thesis was based on the systematic and methodological approaches defined by Dixon (1966), Archer (1965), Groover (1980) and Pahl - Beitz (1984).

4.2 The Design Concept

In the way in which we define it below, design is a very ancient art, applicable to almost any system, and it can be called a science when used in engineering. Archer (1965) calls it "... a goal directed problem solving activity", Farr (1961) defines it as "The conditioning factor for those parts of the product which come into contact with people". Perhaps one of the clearest definitions is given by Reswick (1966) as "It involves bringing into being something new and useful that has not existed previously".

The actual design methodology adopted in this thesis was based on breaking the design process down into six stages, as proposed by Dixon (1966) and Groover (1980). Figure 4.1 shows the inter-relationship between these stages.

i)	Establishment of need
ii)	Definition of the problems (goal recognition)
iii)	Synthesis (the design)
iv)	Analysis and optimisation
v)	Evaluation (functionality and review)
vi)	Presentation (drawings, flowcharts, specifications)

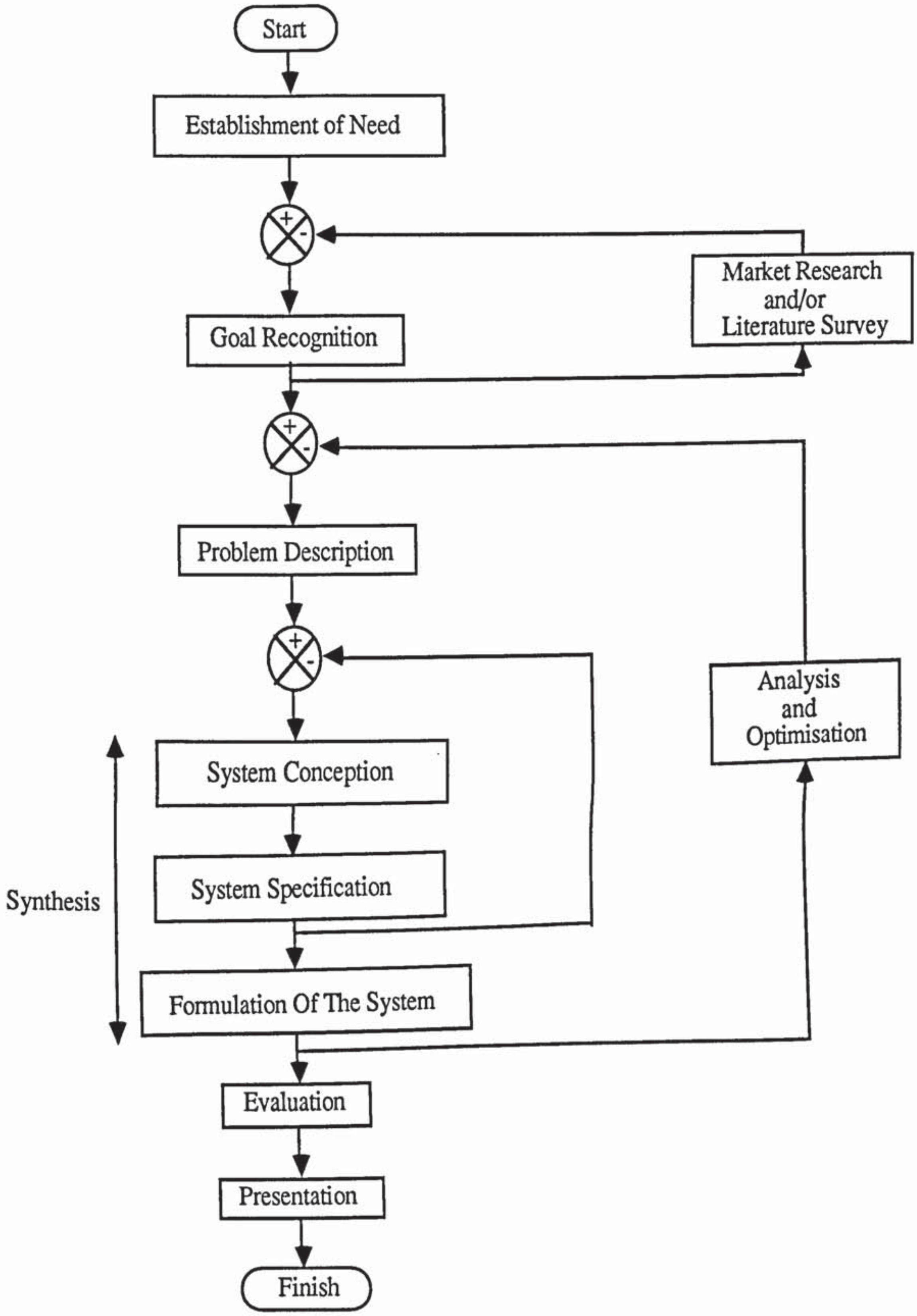


Figure 4.1 The Stages of The Design Process

complicated process, and the designer must not only have a creative mind but also should be very methodical and work in a structured manner. There are a number of techniques available to aid the design process and design is sometimes called "the problem" while the design techniques listed below are known as "problem solving methods".

a)	Brainstorming	b)	Synectics
c)	Delphi technique	d)	Morphological analysis
e)	Attribute listing	f)	Marple's decision trees
g)	Inversion	h)	Fantasy
i)	Empathy	j)	Analogy
k)	Visualisation		

The definitions of these and methodological approaches are explained in depth by Webster (1961, p 266), Norris (1963), French (1971), and Prince (1968). However, it is approriate to provide a brief explanation of Marple's decision trees, since the entire design concept of the proposed system is based on this approach.

4.2.1 Marple's Decision Trees

Any design can be regarded as the outcome of sequence of problems and their solutions. Gregory (1966) argues that the complete problem must be broken down to several smaller sub-problems in a tree structure. Solving each of these problems will solve the main problem. Figure 4.2 shows how this method can work down to the smallest function within any system. This approach was the base for the development of each module within the proposed CAPM system.

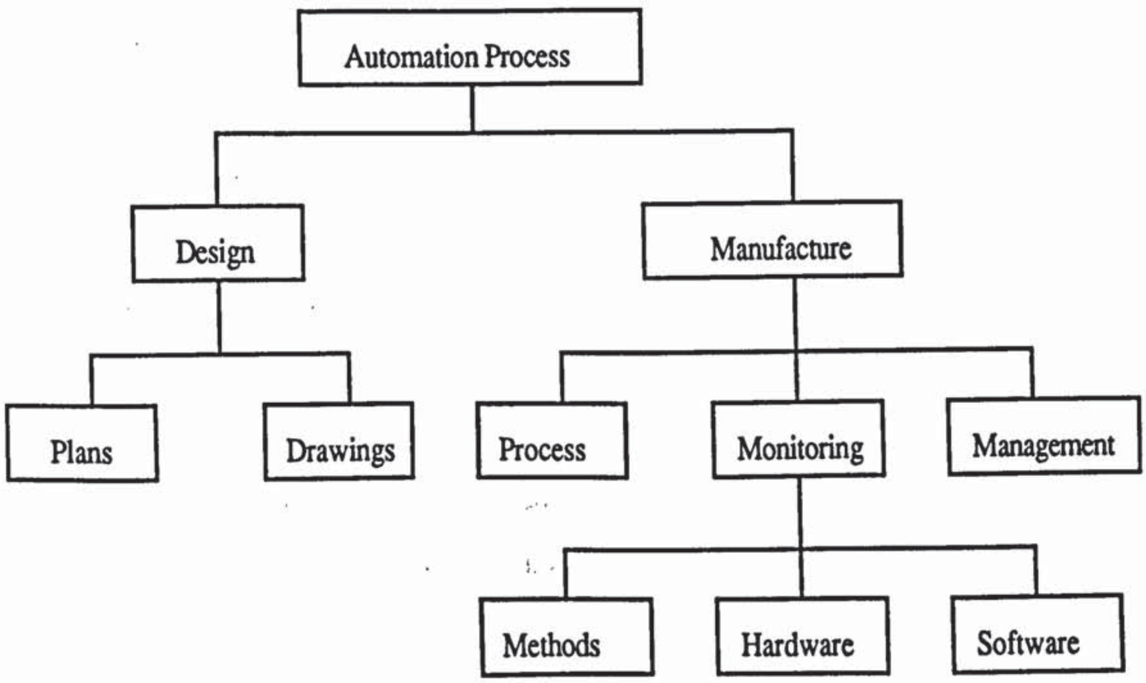


Figure 4.2 Marple's Decision Tree for an Automation Process.

The application of the above approach helped to develop the system thoroughly, very fast and very much to every one's satisfaction. Each function shown above has its own sub-function and all functions have at least one process to achieve the objective of the function.

4.3 The Establishment of Need

After studying the existing manual system at Company B and having discussions and interviews with the personnel, the main problems which needed attention were found to be as follows:

- No part numbering and no formal stock control system.

- No computerised production control system (i.e. Planning, Work to list, Scheduling, Shopfloor feedback and Monitoring).

- No systematic usage of material (i.e. Picking List)

- Inadequate use of works orders and process route cards with sufficient information to point out expected problem areas for the operators.

- No subcontracting procedures except informal routines between the subcontractor and the planning department.

- Limited facilities for quality control and totally manual paperwork for quality control requirements.

- No traceability methods during the course of production.

- No reconciliation at all at any stage - as a result, no formal costing system.

The above problems gave rise to the following effects and caused major upsets within the organisation. Unfortunately, these problems were being neglected and no long term solutions implemented.

1) Inaccuracy of planning details.

2) Inconsistent process sequencing.

3) Poor material yield, complex production rules and routings.

4) Lack of understanding about production rules and routings by the planning department.

5) Lack of knowledge about the status of material availability.

6) Lack of communication between Planning Department and Process Planning office.

7) Lack of communication between inventory control personnel and the process planners.

8) Backlog of paper work and the difficulty in finding the right records.

9) Expensive and time consuming task for training new planners.

10) Company is totally dependent on the skill of the process planners.

11) Customer delivery dates are not met.

12) The need to reduce work in progress and mill stock.

13) Poor overall efficiency.

14) The need to increase profit.

4.4 Goal Recognition

Inaccuracy of planning details gives rise to many industrial problems today and most manufacturers are aware of this. The simplest method to eliminate (or at least reduce) this is to use more manpower by employing more people. Even this would not give the expected levels of accuracy and it is impracticable due to the fact that the profits will be further reduced when the extra wages have to be paid. The alternative to this is to look for some type of computerised solution. This type of a solution will be more difficult to implement and also will be costly in the initial stages, but it will certainly have a higher degree of accuracy in the long term as it will be free of human error and eliminate most of the problems outlined above.

Inconsistent process sequencing results in the reproduction of process plans by the process planners operating on the shopfloor. Sometimes, if these errors are not noticed, the batches of orders go through the wrong sequence of operations, finally being scrapped by the quality control department. Therefore, generation of process plans at shopfloor level becomes important if this problem is to be eliminated.

The computerised process planning system must try almost all possible planning alternatives and generate the corresponding range of material yields for the processing of each order. A computerised system should be developed to mix orders and group these so that optimum base supplies can be produced for each batch.

The existing vague and inconsistent production rules will have to be formally presented on paper in their correct format so that they can be classified in a manner acceptable for use in the computer database. This will force unacceptable routings to be eliminated and the rest to become standardised throughout the company.

A large database must be maintained to eliminate the problem of duplication and out-dated data. Once the system is computerised only authorised personnel will have access to this information and only they will be allowed to update the common files in use throughout the company. The system would have to run on a network cluster distributed through out the company.

The database system would be the main body providing all the relevant information about inventory control, material status, material allocation and all the relevant details on customer records. The system will therefore, eliminate the lack of communication within the departments and would provide excellent traceability features for the quality control section and also for auditing purposes.

There is no need to have a manual paperwork system running in parallel with a computerised system, except perhaps in the early development stages. The availability of fast data access in a computerised solution would eliminate many problems. This would also help to restore confidence of the operators in the accuracy and reliability of data.

The complete computerised process planning system can be used as a simulation package for training new planners and provide them with a tool to test various alternatives and 'what-if' analysis. This invaluable tool will use the same programs and have access to a test environment where live data files can easily be copied across and used for experimentation. Any data corruption or destruction of live data is thus eliminated.

The Computer Aided Process Planning system, which would be the heart of the system, will have the ability to store all production rules and techniques of generating process plans according to the manufacturing rules. The system will then make use of a separate module to solve the cutting stock problem so that slabs can be utilised and material yield maximised.

The computerised system will produce consistent and error-free plans. It would also utilise the process planners' time more effectively and standardise the rules of practice, resulting in a smoother flow of production and eliminating the reworking of order batches.

Generation of process plans from material on hand, by reference to the inventory records, would ensure the utilisation of unused slabs and excess material in the stock before demanding new whole slabs. This would reduce work in progress and keep inventory to a minimum.

Increasing the overall efficiency will undoubtedly increase the profit margin. This might be further increased by computerising other functions which are linked to manufacturing. These might include computerised scheduling and monitoring systems which would report on the status of the work-flow to the management as a feedback, with facilities for corrective actions to be taken.

4.5 The Prototype Design Concept

The design concept is to build a prototype model of the system by studying in detail the required goals already specified and having constant discussions with the end users. This is an iterative procedure between the company personnel and the designer, who can get valuable feedback in response to his initial ideas (see Figure 4.3). Following this technique and the Marple decision trees, the problems listed in Section 4.2 can be classified in six major areas, as shown below.

1) Shopfloor level
2) Management level
3) Communication
4) Security
5) Networking
6) Staff Training

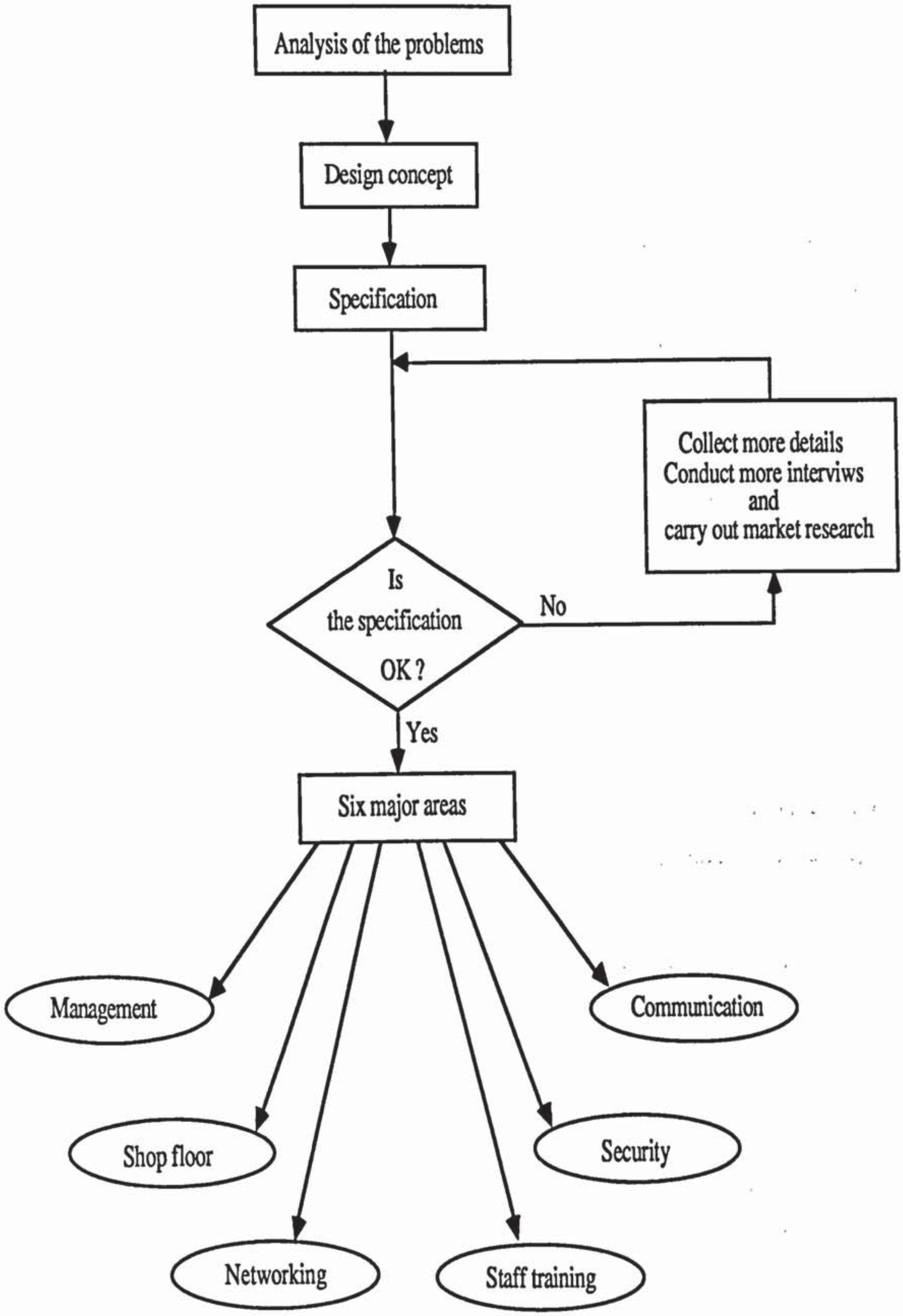


Figure 4.3 The design concept based on the existing problems.

4.5.1 Shopfloor Level

Almost all problems and errors made during planning are uncovered at some stage during production. These problems are either technical (e.g. material composition, surface finish, grain direction, etc.) or planological (e.g. due dates, batch sizes, distorted dimensions, etc.). It is clear that detailed planning and accurate specification will provide the operators with a strict code of practice and eliminate uncertainty resulting from vague and ill-defined instructions. These plans can only be produced if the management has an in-depth knowledge of the exact status of the work in progress, stock, and operator availability at all times.

4.5.2 Management Level

The management level deals with the dynamic behaviour of the shopfloor and should be able to solve all the problems which occur during production. A database management system should be maintained and organised in such a way that information on status of the availability of new material and any scrap worth re-working can be entered and be allocated to orders whenever necessary.

The management should attempt to standardise the codes of practice and process this knowledge to provide the process plans according to standard routings. It is the job of management to make sure that all information necessary for production is provided and no operator should re-route orders, because of lack of details on the process cards, without the consent of the management.

The management level also has to deal with the finished jobs and process all information so as to prepare regular production reports, which consist of profit or loss on each job, efficiency and a host of other information. Hence the management level comes before and after the shopfloor level, which is sandwiched in the middle (see Figure 4.4).

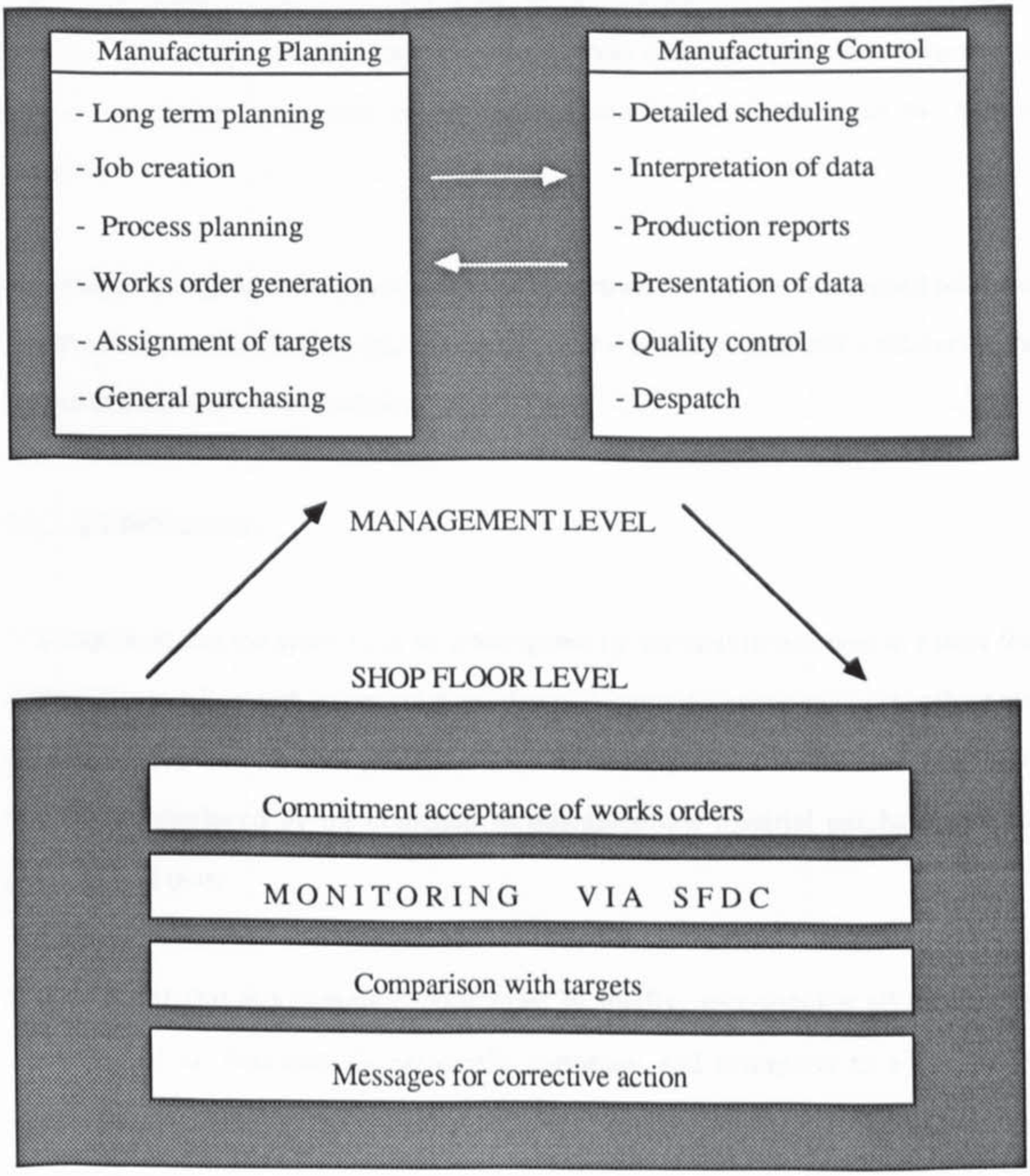


Figure 4.4 Shopfloor and management levels

Finally, the management level can attend to stock control functions for both raw material and the finished products. The relevant information for this section need not be entered again if it is already available in another part of the database but the computer can locate, retrieve and use these automatically. The management area will thus become a very large section which would involve an enormous amount of work to make the system complete.

The system should be able to provide senior executives with a pre-determined selection of key performance indicators and information over a period of time that is relevant to the control of the company as a whole.

4.5.3 Communications

It is important that the system can be interrogated by the sales department to ensure that orders are not taken with promised dates that will upset the status of the shopfloor and any order cancellation is done promptly. Reports on the status of the material availability that can be produced by the computer to guarantee raw material purchase with the required lead time.

It is essential that this communication must be totally successful at all times. The up-dating of all databases is extremely important and discipline to abide by the instructions provided by the system at all levels of management and shopfloor is vital. Therefore, it must be noted that the communication area is one which is very delicate and must be handled with the utmost care. Investment in training is vital and the concept of 'garbage in, garbage out' must be emphasised.

4.5.4 System Security

Security plays a major role in an advanced system such as this, especially when it has to be a multi-user system. The system should be able to exchange information with several other systems and be able to provide facilities to authorised personnel for editing, deleting, copying and creating various files. It must be stressed that the users of such a system may have little computer literacy, and therefore this poses a very complicated problem. It would mean that they cannot be allowed to go into the normal operating system level to achieve their objectives for the simple reason of protecting their own data from possible deletion. The system therefore has to be closed, so that users can only have access to the programs and data is only accessible via application software.

4.5.5 Networking

The system must essentially be a multi-user system. However, the prototype model would be developed for a single user to achieve the required goals. It should be capable of answering various levels of management and be able to be linked to other systems for the purpose of inventory control, scheduling and monitoring systems. Finally the system should be able to be linked to the finance department and provide the necessary information for costing, accounting, payroll and job estimating functions.

4.5.6 Staff Training

With such an advanced system one might assume that staff training is unnecessary, but the value of it will be realised only when the feedback is considered. The accuracy of the system can only be found out by usage and therefore the users must be trained in the proper use of the system. The prototype system could then be debugged and improved to a great extent when the feedback of the users is utilised properly. Therefore, although staff training is a lengthy and often annoying process, it is nevertheless an essential part of designing and building a system such as this.

4.6 Model Development

The development of the model is based on the functional requirements pointed out in Sections 4.2 and 4.3. A comprehensive model is therefore devised from Sales Order Processing to Shop Floor Data Collection and Machine Monitoring. However, due to the nature of this work, the complete system can take up to several man-years of programming and it involves many research areas which need exploring. Nonetheless, one of the most difficult parts of this system, the generation of process plans and intelligent use of stocks through the generation of practical cutting pattern, has been developed. The analysis of the algorithms used in pattern generation and mixing the orders is considered in Chapters 8 and 10.

Figure 4.5 shows the overall design of the model. The concept of the system is one of Data Management. The majority of processes will be relatively simple individual updates of the extensive, integrated data base system. The exception to this will be the processes used to model events, e.g. CAMO (generation of cutting patterns) and production scheduling, which are likely to include sophisticated algorithms and calculations.

The extraction and presentation of data at this stage could be assumed infinite, as it can be handled by Man Machine Interface (MMI) techniques and use of the Fourth Generation Languages.

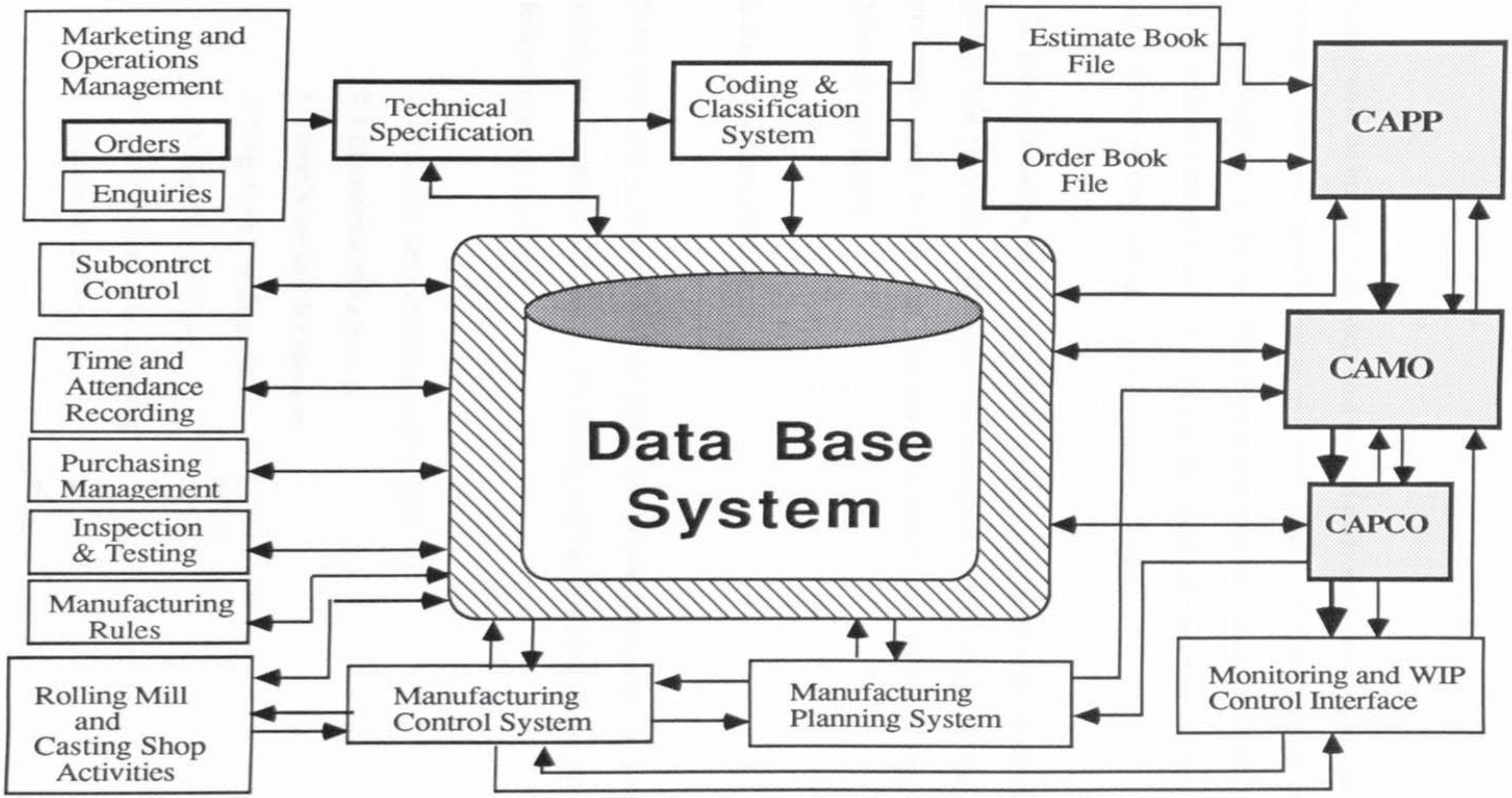


Figure 4.5 The overall model of the proposed production managemant system for the hot rolling mill industry

Data would be imported and exported to third party software such as "the CAPP system" designed and developed by Husseini (1989), also the CAMO module developed in the 'C' programming language. This feature is particularly useful in any integrated system, since it allows the user to output data to the company's existing accounting software or spreadsheet without having to purchase new software.

The model however, has to be split into three sections in order to assign realistic time phases and priorities. Although the functional specifications of most functions are proposed, the author has concentrated mainly on Phase I, the Manufacturing Management option.

4.6.1 Model Specification

With reference to Figure 4.5, the design and development of the model required to satisfy the organisational need of any rolling mill environment can be categorised into the following seven functions.

1. Executive Information System (EIS)
2. Commercial Management
3. Manufacturing Management
4. Manufacturing Support
5. Design Management
6. Administration Management
7. System Wide Functions

As mentioned in Section 4.6, the three stages of development are depicted in Figure 4.6.

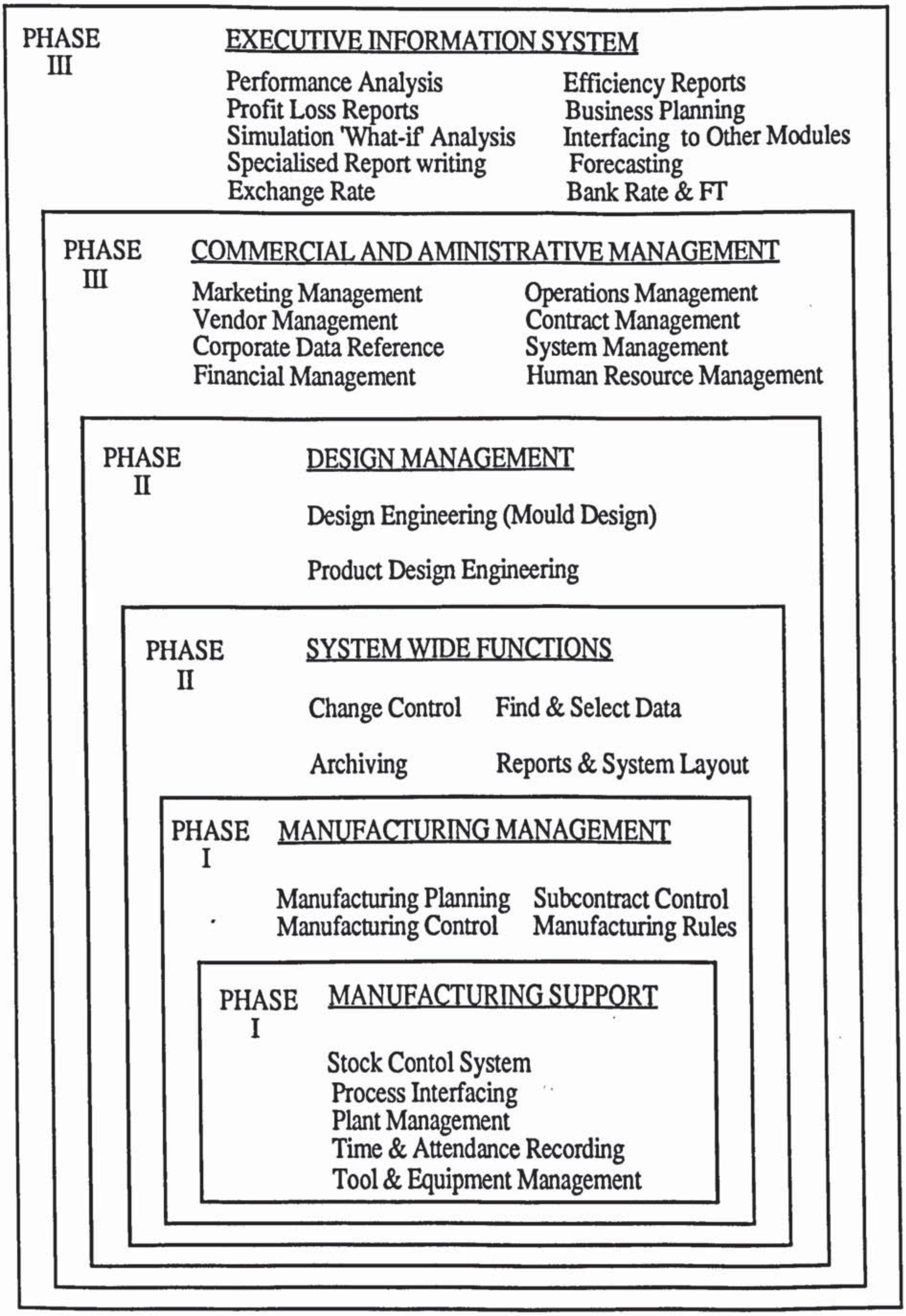


Figure 4.6 The Various Levels of the Proposed System

4.6.1.1 The Executive Information System

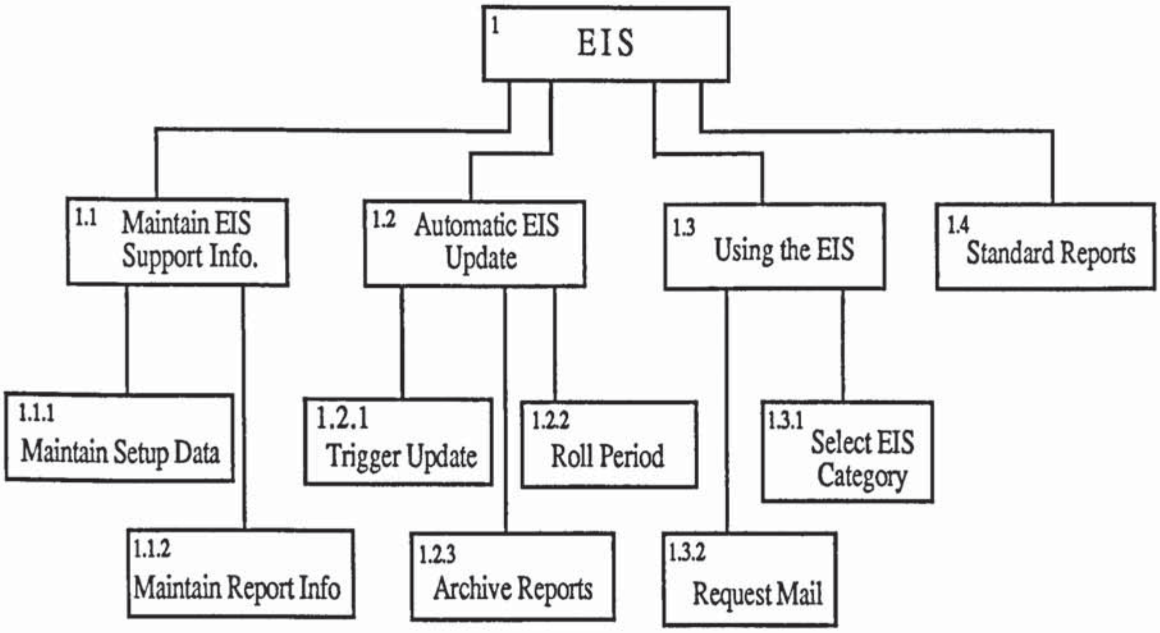


Figure 4.7 The Functional Structure of the EIS.

This part of the functional specification provides the information requirements of the senior executives within the organisation. The objective of the system is to perform the following functions:

i) To provide senior executives with a pre-determined selection of key information which is judged to be of relevance to the control of all manufacturing aspects.

ii) To provide a flexible means of allowing the senior executive to specify particular high level information requirements and to configure the system, either by himself or by the DP manager.

iii) To provide the senior executive with software tools for planning and budgeting purposes.

The system would provide the following categories:

SALES	Orders received, Invoiced, Quotations and Estimates, Visits, Services, Order book, etc.
PRODUCTION	Quantity, Value, Quality, Performance, Utilisation, Efficiency, Utilisation, Lead times, Material Yield, Work in progress.
INVENTORY	Quantity, Value, Turnover, Age, Returns to refinery, etc.
MATERIALS	Purchasing performance, Vendor performance, Energy usage.
BUSINESS PLANNING	Forecasting, Inventory policy, Production, Cost models.
FINANCIAL	Accounting data, Balance sheet, Overheads, Cash flow.
PERSONNEL	Staffing, Attendance, Overtime, Costs, Labour intensiveness.
DESIGN	Product profile, New mould designs, Specification, Performance.

EXTERNAL INFO. Ability to read in any files from standard spreadsheet or any other sources. These may be; metal exchange rates, bank rates, FT index, etc.

4.6.1.2 Presentation of Data

The information provided at this level should have the following options for printing:

* Rows and columns of data
* Graphs
* Bar charts
* Pie charts
* Messages

The EIS should not be updated minute by minute, instead a schedule of updates will be defined by the user (maximum twice daily) and the EIS will update itself accordingly. In addition, the user should be able to request an update as required.

Since this option is only a complementary part of the system and is not the main objective of this thesis, further investigation and research has to be carried out to complete this function.

4.6.1.3 Commercial Management

This area covers those functions involving contact with external organisations. The function is designed to support the activities undertaken by an organisation to generate sales. It includes those activities normally associated with Marketing Operations, but does not include Marketing Development.

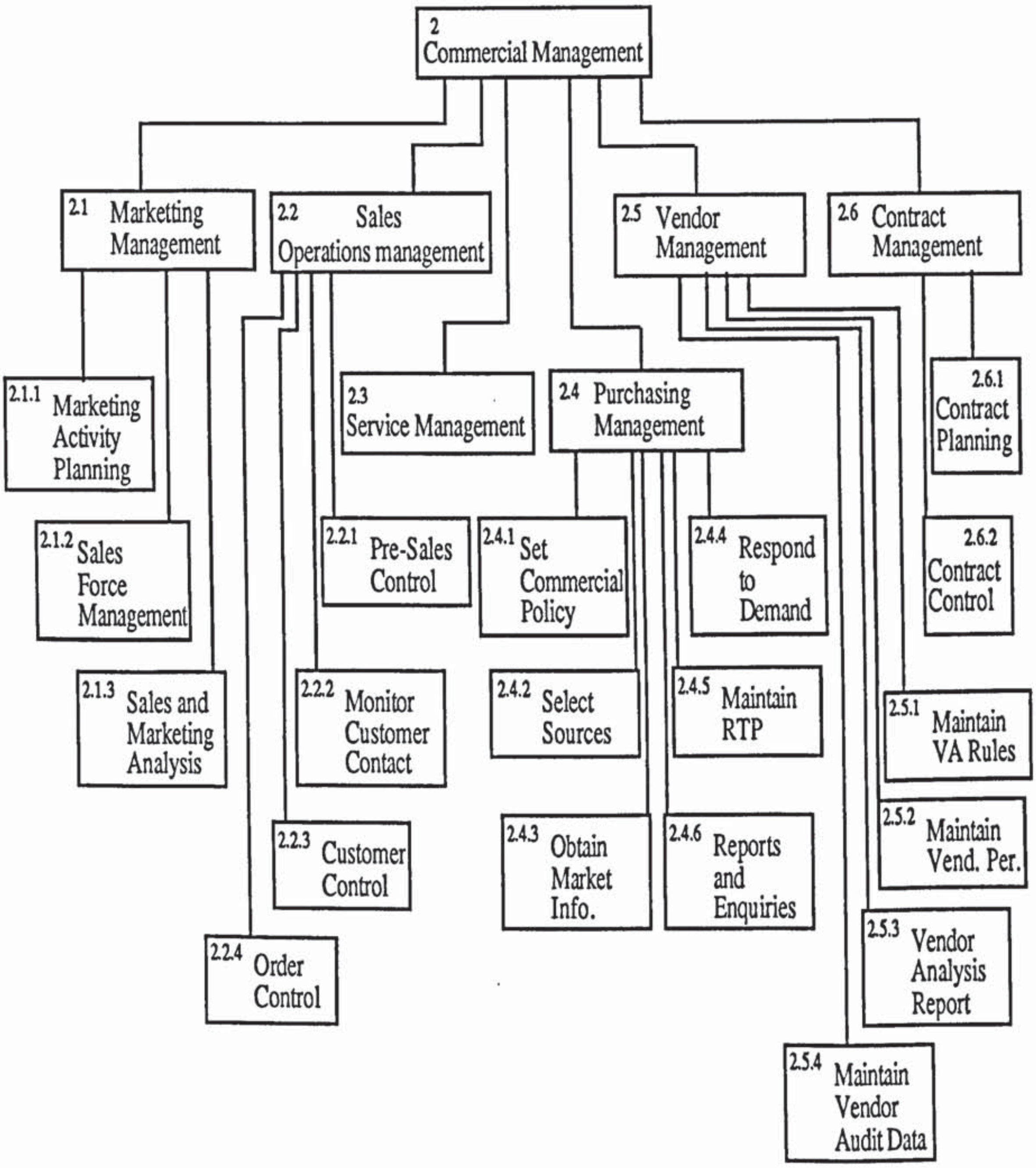


Figure 4.8 The Functional Structure of the Commercial Management.

The objective here is to support Sales Management in planning and controlling the sales and sales support activities, in house, agents and distributors together with the monitoring of their performance against forecasts. Marketing can be defined either by product or market area or customer type, therefore when setting up the various data some thought must be given to the marketing definition.

The sales cycle starts with a lead tracking database. Customer contacts can also be logged and could complement the future marketing function. A commercial estimating function is provided which may precede the sales order processing cycle which follows the sales order through to despatch and can raise financial transactions. These can then be used to compare against the estimate originally given and also for passing the data to sales ledger of the accounting modules.

The purchasing system should follow the buying cycle from enquiry logging through order placement and acknowledgement tracking to authorisation of payment. A range of orders should be supported from simple or multi-line orders to purchasing schedules.

Contract management forms an "umbrella" function to draw together a thread of commonality between jobs and tasks at a lower level in the system. In this way the system makes use of the lower level data to provide information at contract level for deliveries, resourcing and cost control. This is particularly useful in the refinery operations where slabs are being cast for external customers. However, the functionality may be limited to defining tasks, interdependency, non-manufacturing tasks (e.g. roll grinding), and extensive analysis and reporting.

4.6.1.4 Manufacturing Management

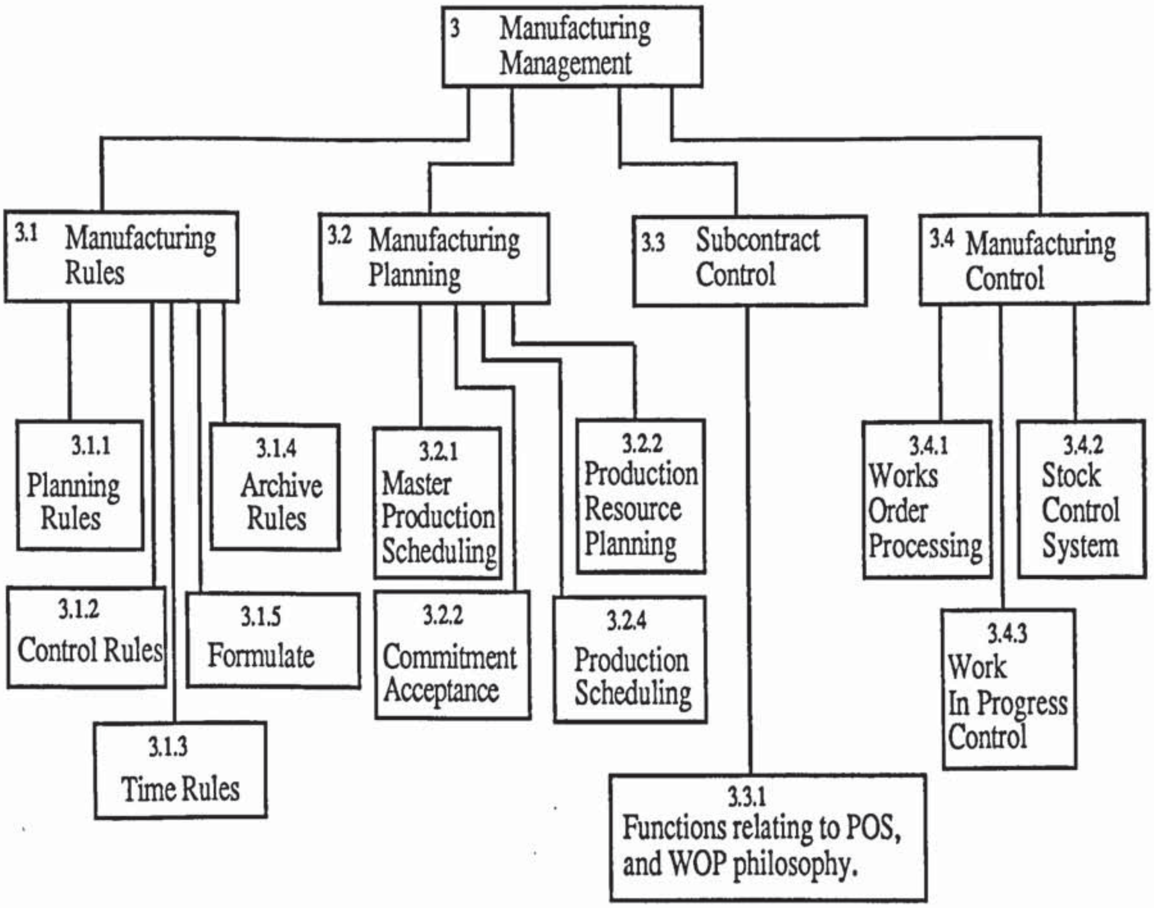


Figure 4.9 The Functional Structure of the Manufacturing Management.

This area would cover those functions involving contact with the shop floor and liason activities with the commercial section. Therefore, the purpose of this option can be defined as:

a) To collate the requirements already defined in the business marketing and sales sections and, using Master Production Scheduling and Rough Cut capacity Planning techniques, establish and maintain plans, agreed between all parties, of what the production department will make and when.

b) To provide the resources required to meet the plan, using material and capacity planning techniques (both infinite and finite).

c) To communicate all external requirements to the purchasing department.

d) To monitor, control and audit the stores and shop floor activity required to achieve the plan.

This is the main area of interest in this project and, although satisfying the total function would take much in-depth research and months of programming, a thorough study of the subject has been analysed systematically in such a way so that any knowledgeable programmer familiar with manufacturing terms should have no difficulty in coding the above functionality. The detailed specification of this function is discussed in Chapter 5.

4.6.1.5 Manufacturing Support

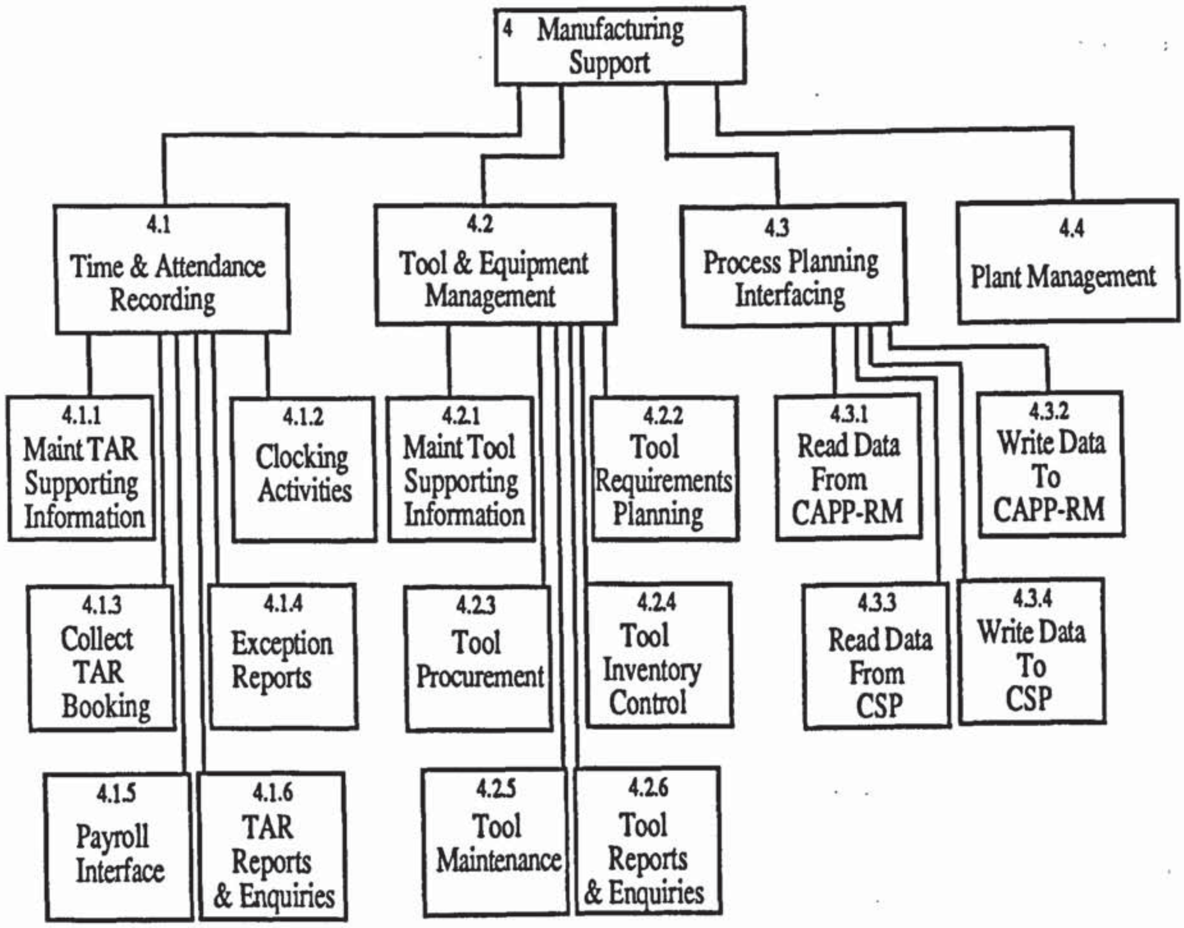


Figure 4.10 The Functional Structure of the Manufacturing Support System.

Time & Attendance Recording is concerned with recording the attendance at work of company personnel, reasons for absence and reconciliation of clocked to booked hours. It is aimed primarily at shop floor operators. It could equally be applied to staff, supervisors and management if desired. It is concerned with keeping a record of who is available and what those people do once they are at work. Once this has been done, production activities can be triggered and scheduling and machine loading commence. It is envisaged that data collection terminals will be used for clocking and reporting job status on the shop floor.

The plant management function controls aspects of physical resources from maintaining database of assets and resources to planning and executing maintenance.

4.6.1.6 Design Management

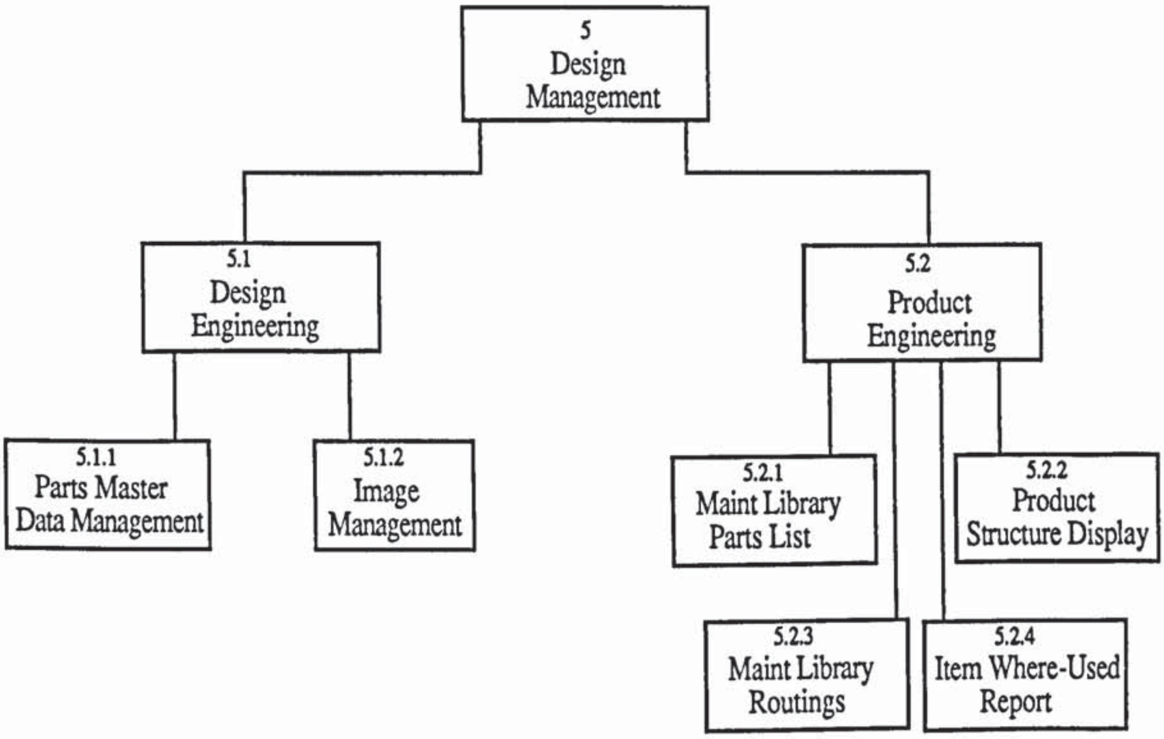


Figure 4.11 The Functional Structure of the Design Module.

This area covers the maintenance of a data base of 'static' information required to support the manufacture of products. It includes all design and drawings of the moulds that produce the slabs, rolls that roll the metal and any other tooling required for slitting, blanking or shot blasting.

Design engineering covers the database of product information. This is the item master file, which contains all items whether stock, current, future or temporary. A drawing register can also be maintained.

Product engineering covers the maintenance of a database of methods. This includes routings and product network relationships. Text fields can facilitate process instructions and change control for engineering designs.

The applicability of this function is best utilised in a typical batch manufacturing environment with hundreds of parts, assemblies and sub-assemblies.

4.6.1.7 Administration Management

There should be a range of functions in the system which would act as a "service" to the rest of the functions. These are divided into system management and control areas, which are needed to look after the system itself and provide data processes used by many other functions in the organisation.

The system management function provides facilities to maintain user access privileges and personalised levels of Man-Machine Interface (MMI) such as menus, paths and help facilities. It would also cover the control of potential areas of conflict within the internal processing, which can be resolved by multi-user control and batch or stream processing. A functional control programme can be set up to configure the system to suit the user.

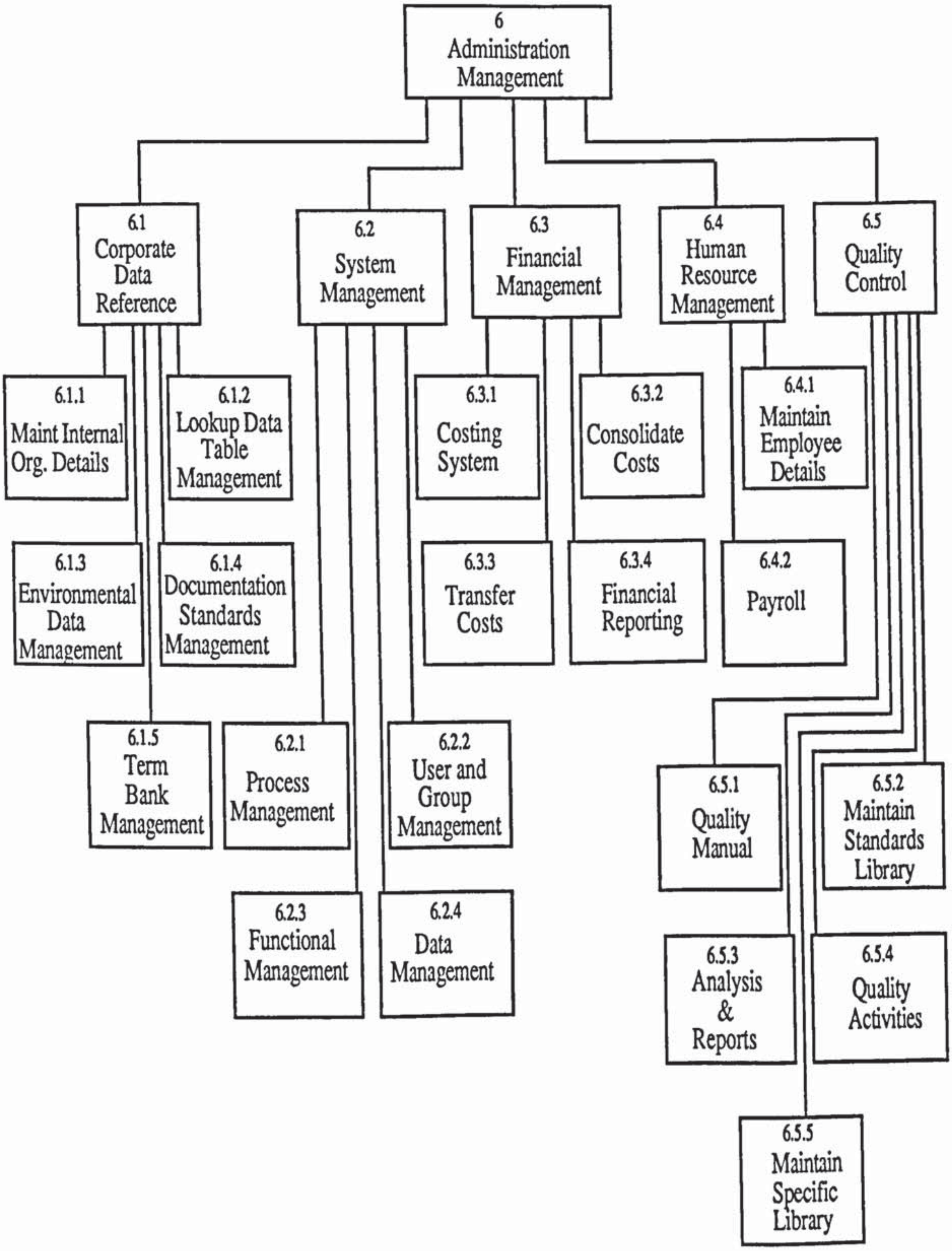


Figure 4.12 The Functional Structure of the Administrative System.

Administrative areas serving other functions in the organisation are identified as follows. A database of corporate information, such as standards, details of the organisation, etc. is maintained. Human resource management includes maintenance of employee and skill details, logging of time and attendance recording.

Quality control function provides the shop floor with the quality manuals available, the specific standards or company standards. The feature to generate reports is particularly useful for the inspection department to carry out tests.

4.6.1.8 System Wide Function

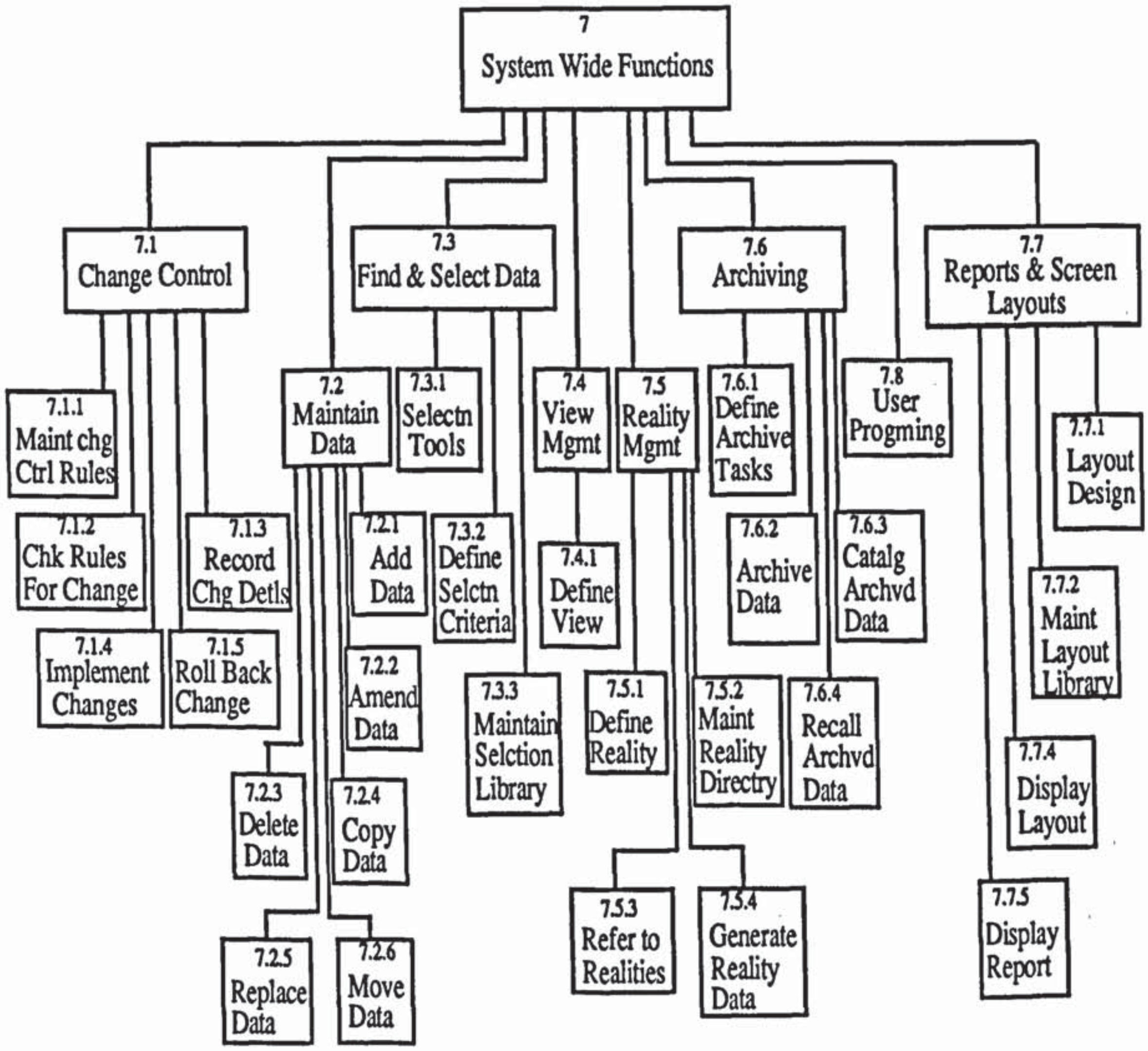


Figure 4.13 The Functional Structure of the System Wide Function.

This section contains functions which are used by, and directly available to a range of application areas. These are system generic functions, in that these functions should be available at any point, and are basically applicable to any data or any function as appropriate. The exact operation may be specific, but the "look and feel" must be consistent throughout the system.

Some of these are basic 'building block' functions and tools (e.g. 'add'), some are specific functions with global domain (e.g. view management), and some are specific functions with global accessability (e.g. help).

4.6.2 Prototyping

Prototyping is an important step towards the development of a well thought-out model. Kendall and Kendall (1988) describe prototyping as a technique for quickly gathering specific information about users' information requirements. The prototyping of the complete system was not possible in the course of this project. However, the essential parts of this such as the solution to the Cutting Stock Problem are thoroughly researched and developed in the later chapters.

4.7 Choice of an Operating System

A survey of operating systems available is carried out in Appendix A. However, the most widely used micro computer operating systems at the moment is the Microsoft's Disc Operating System better known as MS DOS.

The operating system is cheap and it usually comes with the purchase of the micro. It has many limitations as specified in Appendix A. However, most applications are designed to utilise the 640K limitation to its maximum.

It is also cheaper to implement a Local Area Network LAN running DOS such as Novell Netware.

4.8 Selection of a Language

The criteria for a successful CAPM which can operate under a number of computers and be machine independent are as follows:

- The system should be able to be readily (or with little modification) installed on a range of computers and operating systems.

- It should be easily integrated with third-party packages.

- The response time should be quick and it should be capable of operating at machine level without having to write machine code programming. This is particularly useful if one tries to control some device through the machine ports and input or output data.

Both Appendix A and Appendix B give a background to the operating systems and the programming languages available. However this project was restricted to what was available at the time and how it could be best be applied to current problems.

Although procedure oriented languages such as Fortran, Pascal or C offer considerably more flexibility to the user who has a deep knowledge of them, it is still difficult and time consuming to write programs in these languages to accomplish such tasks as simulation, matrix manipulation or linear programming. The languages specifically developed for such tasks are far more easily applied. The usefulness of computer languages and their wide variety poses a problem for both professional and occasional programmers. Higher level languages for business and scientific computing are firmly established and are gaining in popularity every day. The question is no longer, "Shall I use a high level language ?" but, "Which one shall I use ?" There are indeed enough languages available to make the choice a complex, if not difficult, one. Not only are there many languages available, but in some cases, several dialects of one language exist.

However, in developing application packages such as CAPM systems, one looks for efficient programming and bearing in mind the three points mantioned above will lead to the selection of a language rather than a "shell". The programming shells such as MATLAB do not have facilities to compile the task programs and hence they tend to be much slower than equivalent compiled programs.

4.9 Summary

The design concepts developed in this chapter have been based on the methodologies adopted by Dixon (1966), Groover (1980) and Pahl and Beitz (1984). The problems at the two companies studied in the previous chapter with particular emphasis on Company B is summarised in Section 4.3 "The Establishment of Need". The functional requirements are expanded in Section 4.4, "Goal Recognition" and they are categorised in six levels explained in Section 4.5.

A model is developed using the Dixon and Groover technique in Section 4.6 with the functional specification categorised in seven levels in the same section. They are then classified according to the need of the organisation at the time and the bulding blocks are drawn out in details. The functional specifications are discussed in detail in the next two chapters.

Finally, a review of the choice of an operating system and selection of an appropriate language is provided in Appendices A and B.

CHAPTER 5

DEVELOPMENT OF THE MANUFACTURING MANAGEMENT FUNCTION

5.1 Introduction

The term 'Production control' is often used to cover both planning and control activities, although there is an obvious difference in the meaning of the two words. 'Production Planning' deals with 'Pre-Production' activities such as preparation of Production Schedule, acquisition of materials and Production Resource Planning for future production, while 'Production Control' deals with the situation 'during production', such as the implementation of a predetermined production plan and control of all aspects of production according to such plans.

In this project the term Manufacturing Management has been defined to cover both Production Planning and Production Control (see Chapter 4 Fig. 4.3). The function consists of four modules each controlling a number of sub-functions or processes to conform to the standard manufacturing terms.

Various authors Burbridge (1978), King (1972) have provided case studies and implementations of production control systems and each one has formulated their own definitions of what production planning and production control is in their environment. However, The British Standards Institution defines Production Control as "procedures and means by which manufacturing programs and plans are determined, information issued for their execution and data collected and recorded to control manufacture in accordance with the plans".

In most manufacturing organisations, planning and control of production are the responsibility of the same department and this is usually called Production Control. Such a situation is always justified since, although two identifiable aspects exist, they do not exist independently nor in isolation. Production Planning and Production control are very closely linked and entirely interdependent. Decisions during planning will determine the problems and often the nature of control, and experiences during control will influence future planning.

Under the umbrella of Manufacturing Management, Production Rules and Subcontract Control have been included to provide complete functionality of this module. Moreover, a second function, Manufacturing Support, has been included to complement the Manufacturing Management function and provide vital information for the Manufacturing Planning and Control processes (see Figure 5.1).

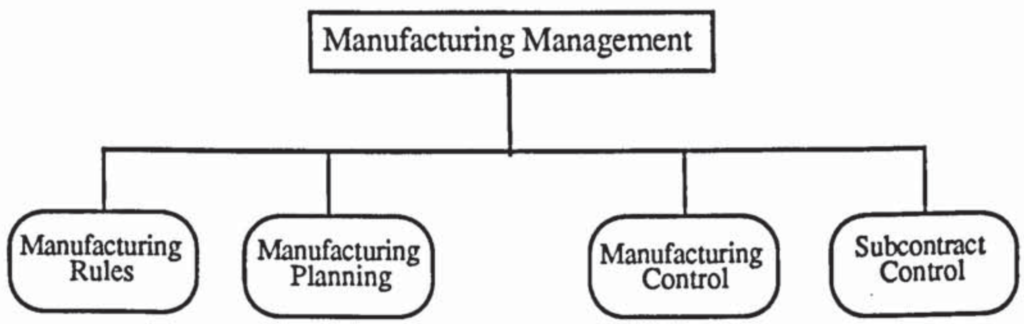
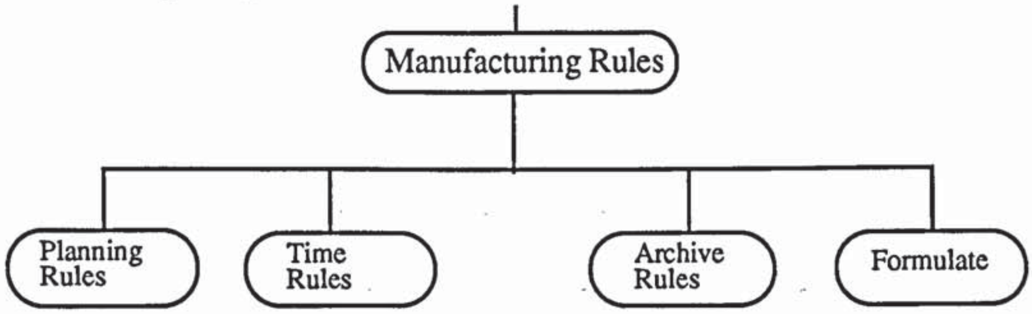


Figure 5.1 The Four Major Functions of the Manufacturing Management System

This chapter and next chapter will attempt to describe the functionality of the proposed computerised Manufacturing Management system and demonstrate the need for the Manufacturing Support function.

5.2 Manufacturing Rules



Figore 5.2 Functions of The Manufacturing Rules

This module allows the user to identify the various rules and procedures that apply to the manufacturing section as a whole. It is one of the most difficult and disciplined requirements of the system. All manufacturing rules will have to be well documented and translated in such a way that they can be entered into the right process under this function. They include:

1) User defined formulae

2) Options that would be controlled by "System Data"

System Data is a set of control functions which are only available to the the systems administrator or the manufacturing manager and can override a set of rules entered at lower levels.

5.2.1 Planning Rules

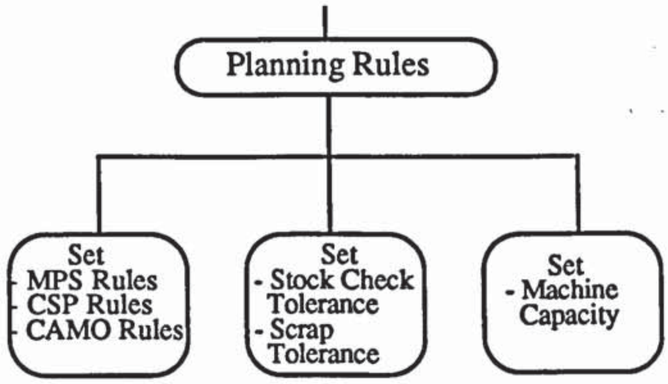


Figure 5.3 Functions of the Planning Rules

This is to allow the user to set rules, procedures and options relating to the Manufacturing Planning section.

At the time of planning the system will frequently refer to these rules and carry out further calculations and planning. The stock check and scrap tolerances are a number of set rules which can be defined (or reset) by a higher level authority. Also the machine capacity is the maximum length, width and reduction ratios set by the planners.

5.2.2 Control Rules

This process allows the user to set rules, procedures and defaults for the Manufacturing Section as a whole. These rules are concerned with:

- Works Order Processing (WOP) parameters
- Stock Control System (SCS) parameters

The parameters which are autogenerated are Works Order No., Works Order prefixes (e.g. if there are a number of sites within an organisation a letter is prefixed for each works order to indicate the site), Stock allocation rounding (e.g. number of decimal places to which the stock should be rounded to), Negative Stock on Hand, location, bin numbers and a whole host of other parameters.

5.2.3 Time Rules

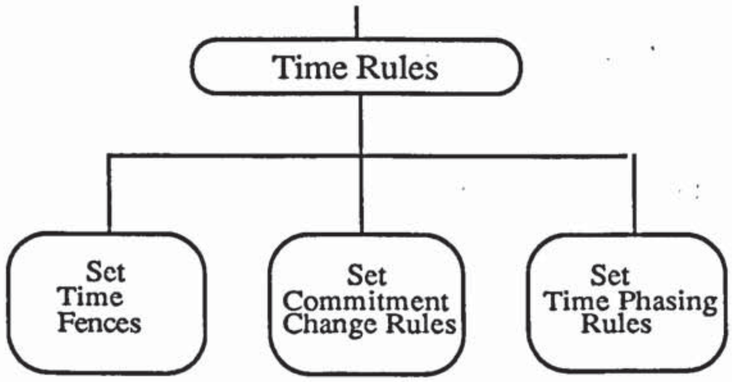


Figure 5.4 Functions of The Time Rules

This process can be used to input time fences for MPS/CAMO/Commitment Acceptance purposes and set the different rules applicable when a change is desired.

This means that when a change is requested the system needs to apply certain rules to check its acceptability. Different rules will be used depending on how far away in time the proposed change is. This identifies those points where the rules change - the "Time Fences." The Commitment Rules require the actions specified to accept orders for each of the time fences specified by the 'Set Time Fences'.

The Time Phasing Rules can be slightly complex and need more attention. This process once set will calculate theoretically on which day each operation of a Production Routing would take place. The following logic will have to be included:

(1) No time phasing - all capacity is assumed to be required in the same period as the demand.

(2) No allowance for queueing - i.e. calculate for each operation the time required:

(setup time+ Re-Order Quantity (ROQ) * unit time) + transit time)/ default hours per day

(3) (2) + planned queue for each operation according to Workcentres

(4) (2) + average queue calculated from:

(Leadtime - total time required) / operations

(5) User defined formula

5.2.4 Archive Rules

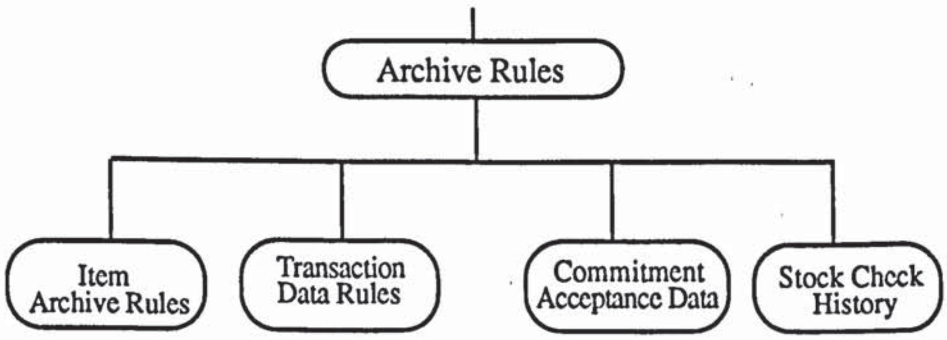


Figure 5.5 Functions of The Archiving Rules

As the name implies, archiving allows the user to define when various items of live data are no longer needed on a regular basis and can be archived; it also decides how long the data should be kept. This is particularly important for the purpose of Quality Control, or for a rolling mill engaged on MoD contracts.

The rules above (see Figure 5.5) determine how long and on what basis the data items should stay on the system once they have become obsolete. Also the transactions against a particular stock item will have to be removed at some stage for which the rule is set. In practice the time rules for some of the above items are set to three periods. Periods are defined by the users as one to three months.

5.2.5 Formulation

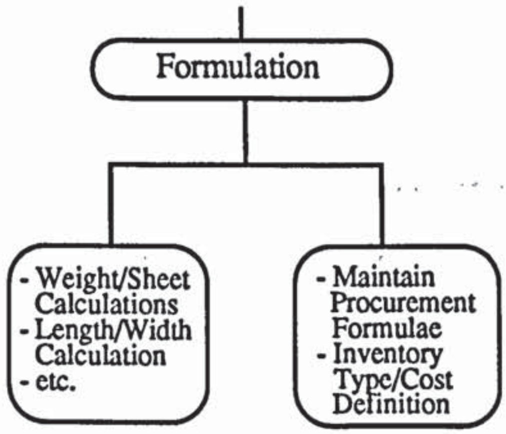


Figure 5.6 The Formulation Algorithms.

A number of formulae may be set by the user by which various parameters can be calculated. This is particularly important in some manufacturing organisations where they do not wish to be dictated by some pre-defined and hard coded analysis of these parameters.

The ability to change or set up user defineable variables based on some calculations also defineable is one of the most important requirements in industry. However, this has been totally neglected in most computerised systems. The procurement will enable the user to define ROQ, Lead Time, Safety Stock, ABC analysis, warning reports, ...etc. Inventory type definitions may be defined as: Redundant items, Obsolete items, Slow moving items, Service items, Repair items, tools and equipment and a whole host of other variables such as roll grinding frequency and coolant change periods.

The scrap allowance formulae is the most needed feature for the Rolling Mill industry where complicated calculations are entered to ensure adequate cover against possible scrap, or overproduction for proofing purposes, of material during production. It should also allow the user to use a proportion factor (i.e. 15% on width or 25% on length, etc.) instead if no formula is defined. The scrap allowance should also be defined for each workcentre.

5.3 Manufacturing Planning

The conception of manufacturing planning signifies an acceptance that production plans are likely to be upset during the manufacturing process. Hence constant monitoring of plans and subsequent modifications to cater for the variations that arise must be accepted. As explained by Nicholson (1978), the first decision to be made in designing a production planning system is whether the production system is to form a major initiative for determining and implementing company policies, or it is simply a 'fire-fighting' service. Top management generally use accounting models to judge the control of the business. These models are often too simple to reflect the operating interactions which take place at detailed shop floor level. This results in the Production Controller being forced to deviate from the intended delivery or stock positions in order to support the accuracy of the financial model (budget). The fire-fighting becomes necessary because the implications of accepting commitments in the form of new orders, changed customer orders and capacity changes have not been worked out in advance and in sufficient detail. The attraction of fire-fighting approach is its flexibility. It may help the management to be in-line with the budgets and to relieve pressure on the production controller from formal plans. In this situation the production controller always promises 'to do his best as soon as possible' for whatever requests are made. This approach requires less effort than establishing in detail, to a sufficient degree of accuracy, what can and what cannot be done. The main weakness of this approach is that it gives no proof that the manufacturing processes are utilised efficiently and/or cannot quote firm delivery dates. Although some form of fire-fighting is always necessary in dynamic jobbing/batch rolling mill production systems, adequate control cannot be exercised without formulating a realistic plan.

In brief, the planner needs to:

- Establish an agreed plan of what the company is going to make and when. This plan must meet business and commercial requirements - the Master Production Schedule (MPS) Benz (1974), Burcher (1985).

- Establish by using Rough Cut Capacity Planning (RCCP) techniques that the MPS is broadly achievable Burcher (1985).

- Establish a formal, disciplined approach to controlling changes to that MPS, and for converting them into live Works and Purchase Orders by marring the sales orders where possible using Cutting Stock Problem (CSP) techniques Wilson et al. (1989).

- Analyse in detail the resources required to meet the MPS via Material Requirements Planning (MRP) and Capacity Requirements Planning (CRP) techniques and to plan shop floor activity on live Works Orders using Finite

Capacity Planning (FCP) and Short Interval Loading (SIL) techniques Burcher (1985), Smolik (1983), Anderson (1981), Wallace (1985) and Looney (1986).

Primarily based upon Master Production Scheduling principles, the MPS is actively managed according to predefined rules using time fences defined earlier so that it remains the best compromise between what Sales and Marketing want and what Production can meet, whilst still meeting broad business targets. If done effectively, this reduces the number of unexpected sales demands that cannot be met by Production and in the case of the rolling mill industry, more efficient plans based on higher yields can be achieved.

To function properly the Scheduler constantly monitors firm and expected Sales Orders and marketing forecasts against the MPS to provide the "Available to Promise" stock.

When the Scheduler judges that a change is required, high level and RCCP checks can be made to assess its acceptability, the former using a spreadsheet approach and incorporating the current business targets.

If the plan looks acceptable, more detailed MRP type and CRP techniques can be used to suggest Works and Purchase Orders and evaluate them using CRP techniques. The Scheduler is bound by the time fences; any of the suggestions that violate these fences will require the change in commitment to be accepted by Production and Purchase. To evaluate them they will be able to use residual finite scheduling Burcher (1985) and stock availability techniques in "what-if" runs. Orders accepted will pass automatically for Works Order Processing and Purchasing having been examined by the CAMO (CSP) functions.

In pursuing the above objectives, the Manufacturing Planning function should facilitate the following activities shown below:

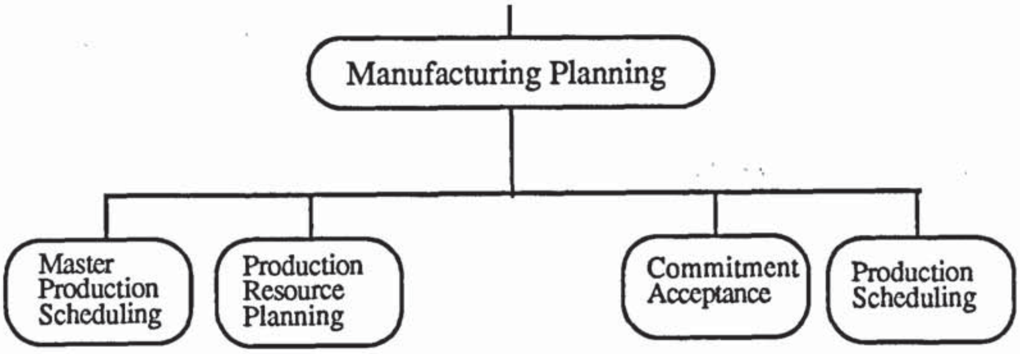


Figure 5.7 The Manufacturing Planning Functions.

5.3.1 Master Production Scheduling

Almost all companies have an MPS, for most however it is relatively informal and not managed. Larger companies tend to be more formal than smaller ones, due mainly to their size. The more formal company will tend to work down from the business level, whilst the others will tend to react primarily to to the existing order book. It is, however, best to plan and not to react. The Master Production Scheduler, if such person exists, Conway (1967) sits between Sales/Marketing and Production, actively managing the MPS in several ways:

- Watering down unrealistic/excessive forcasts
- Liaising with Production so that they do not just make what they want.
- Ensuring that the Business targets are met.

This results in an MPS with "Available to Promise" or 'Unallocated" quantities on various items at various times. Sales can hence:

- utilise current free stock
- utilise "Available To Promise" stock

The MPS function requires an input by the RCCP process which in turn requires:

a) Critical Groupings
b) Critical Process Groups
c) Critical Process Group Capacity

These processes have been designed to allow the user to define which of the Process Groups (workcentres) are critical and group together as necessary for use in RCCP. The critical process groups are lists of processes chosen from amongst the Critical Groupings. The Critical Process Group Capacity is simply the constraints imposed by the user to define their capacity.

The MPS functional requirements and their relationships are shown in Appendix C.

5.3.2 Production Resource Planning

Resources would be the "four M's" - Materials (raw, bought in, tooling, consumables) Men, Machine and Money. Generally the "master" MPS would be used, but it may be necessary to use other MPSs in "what if" runs. The MPS normally extends over a long horizon and can be highly speculative. The MPS already outlined in Section 5.3.1 above analysed this in terms of value, standard hours, etc. and critical process group capacity, and this reflected the lower levels of the items or processes but it took no account of the stocks or existing orders. Our solution to the Cutting Stock Problem (CSP) is used to overcome these shortcomings. It can be carried out over a horizon i.e. consider orders whose due dates fall within a specified period to suggest both works and purchase orders that should be released (input to CAMO). The functionality of CAMO as already been discussed in Chapters 3 and 4. This is rather like an MRP run where all existing orders and forcast are fed into the MRP module and suggested works orders and purchase orders are calculated. However, for the rolling mill industry, the most common practice will be to carry out MATERIAL PLANNING where formal MRP runs with Bill Of Materials BOM is used, are redundant.

Figure 5.8 shows the main functions within the Production Resource Planning.

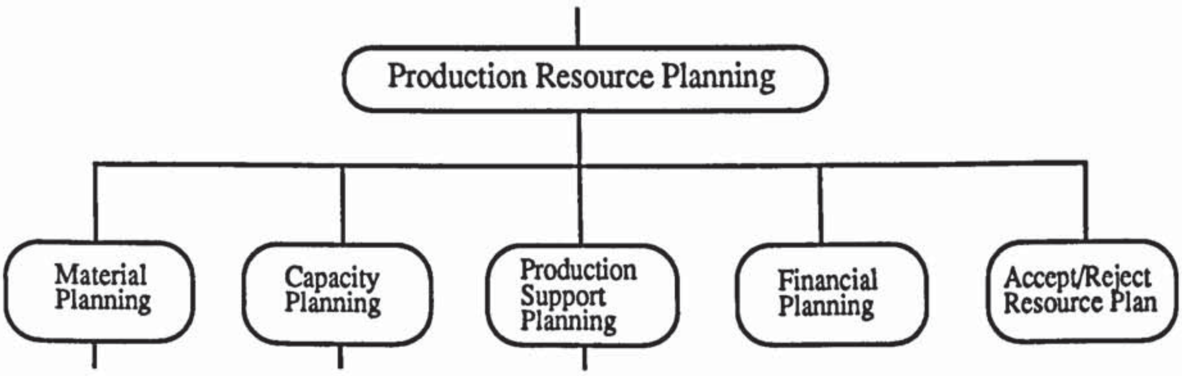


Figure 5.8 The Production Resource Planning Module and its Functions.

The functionality of each module is drawn in Appendix C. The definitions of the Production Support Planning and Financial Planning have not been specified, although it was necessary to include these functions for completion.

5.3.2.1 Material Planning

This function basically simulates the MRP runs. The existing orders that fall within the time fences defined by the Manufacturing Planner are fed into the CSP function. This in turn will use the existing unallocated stocks to utilise the free physical stock for further allocation. However, if the deallocation option is switched on each time this function is run all materials previously allocated are deallocated and the total orders are mixed to calculate the material required together with the suggested works orders.

The system prameters are:

- Material Planning Data
- Identify Items to Review
- Stockturn Analysis Report
- Long Lead Time Items
- Redundant Items

These are some of the system parameters which the Planner wish to change in order to impose various restrictions on the CAMO run. Further discusion of these parameters are provided in Chapter 6. The detailed processes of this function are shown in Appendix C.

5.3.2.2 Capacity Planning

This function is designed to allow the user to maintain the capacity information required by CRP. It should be possible to define the capacity separately from that used by Finite Capacity Planning (FCP) or use that defined in FCP as a base to work from.

The full cycle would involve defining the relationship between Resources and Process Groups, summating the resource capacity accordingly and then amending that as necessary or setting it up afresh. The modules defined in this functions are shown in Appendix C.

5.3.2.3 Accept or Reject Resource Plan

When a plan is drawn, the planner should be able to flag orders associated with the plan for commitment acceptance or processing. The alternative is to reject the plan and return the orders to the SOP or the MPS function.

5.3.3 Commitment Acceptance

In most engineering environments, in particular the rolling mill industry, it is important to establish a formal procedure between the Scheduler (or Sales if necessary) and Production for evaluating, accepting modifying or rejecting changes in commitment. This also enables all such requests and responses between the two parties to be monitored, thus ultimately releasing accepted or modified orders for processing by Production/Purchasing.

If CAMO is used and time fences have been specified, some orders can automatically be released, whilst Production will have the right to reject or modify requests for change that are required which violate those time fences. To do so they will need to be able to check in detail the availability of both materials and capacity. Stock Control System should provide the Planner with comprehensive reporting facilities on the availability and status of stocks. A disscusion of this is given in Chapter 6.

The following parameters should be monitored closely by the system and alarm the planner when violated.

- Production Shortfalls
- Time Fence Violation
- Material Availability
- Capacity Availability

Analysis of Production shortfalls will allow potential shortfalls to be highlighted and a change request made. This also highlights modifications or changes made to Commitment Acceptance. This is particularly useful in that it allows the user to highlight the potential shortfalls in production output and request a change. The time fence, material and capacity will analyse in details the material and capacity requirements and check against the time fence for the production scheduled. If violated, then the system will accept, modify or reject the analysis.

5.3.4 Production Scheduling

This is the application of manufacturing resources to the right operations at the right time Benz (1974). Here, Capacity planning, Sales order processing and the firm programme procedure are all aimed at making the right resources available, and avoiding chaos if they cannot be made available. Requirements planning and stock control are aimed at having the materials available for all stages of manufacture. Hence the objective of Production Scheduling, hand in hand with shop load analysis, is to specify how the available resources are to be used to meet due dates as far as possible, and to get maximum output consistent with it.

The proposed scheduling system is primarily based on the creation of works orders. These can be categorised into two main headings:

- Define Works Order Details
- Maintain Works Order Details

The above two functions create and maintain data describing the work to be included within the production schedule. These are commonly known as operations sequencing and they are determined by the process planning functions. The Planning functions, which are described fully by Husseini (1989) who carried out the process planning and operations sequencing of this project, are based on a generative type system. In addition to this, a Production Structure file can be maintained so that the planners can call a standard structure unique to any repeat sales order and assign that to a Works Order. This structure and routing can be modified to suit the requirements of the particular order. This technique however, is commonly known as *retrieval type process planning*, already discussed in Chapter 3, and is widely used in computerised systems.

The input and tables required by the scheduling function in this type of industry need to contain the following control. Some of these however, should provide the planner with the facility to activate them or switch them off depending on how complex the scheduling operations need to be.

- Due Dates Calculations
- Resource Definition
- Resource Availability
- Resource Capability
- Works Order Priority
- Resource Loading

5.3.4.1 Due Date Calculations

Due date calculations are based on the due date rules. These rules are set by the process planners and also can be read from a file which defines the shift patterns and shift changes. The due dates of the works orders can be established based on loading them to an existing production schedule generated by Finite Capacity Planning FCP.

Two more dates necessary by the scheduling system are Earliest Start Date ESD and Latest Start Date LSD for the Works Orders, the manufactured items within the Works Orders (stages in the routing) and for the operations for those items. This of course will depend on the availability of material or the base supply for the stages of the routing.

5.3.4.2 Resource Definition

In a matrix format the resources should be defined as to what they are, when they are available, what they can do and which production support resources can be used together. One resource may not be capable of doing anything byitself but it may need another resource to support it while doing the job. In other words, each resource would have a number of input nodes and output nodes which would relate to the preceding or subsequent resources. This is sometimes referred to as primary (routing) resources or secondary (supporting) resources. The data will depend on the type of resource which will usually be machines and men respectively.

The above definitions are known as the static data. These can also be partial processes on the routing which are referred to as block routing and therfore a complete process plan of any works order can be a number of these blocks and individual resources. Of course, to the process planning function, the definition of any block consisting of a number of resources is only one resource capable of doing a specific task defined in the RESOURCE MATRIX FILE. Also, the data on employees considered as resources may be taken from the EMPLOYEE MATRIX FILE.

5.3.4.3 Resource Availability

The scheduling system should be capable of planning within two levels of resources e.g. machines and men. The definition of operator availability may be maintained in a function called Time and Attendance Recording TAR.

A resource calendar is needed to maintain a yearly shift pattern to a resource and modify it as necessary to suit the exact working pattern for the resource. Operators should also be considered as resources within scheduling, and so this process may maintain EMPLOYEE CALENDAR DATA.

The calendars may be held as one per resource or as a standard with only modifications noted in a separate file.

5.3.4.4 Resource Capability

This is the definition of the processes that each primary resource can perform and the efficiency of each resource/process combination. There should also be a field within this matrix defining the preference "Picking Order" of which resource will be chosen by the scheduling system.

Each primary resource may be capable of performing more than one process. The RESOURCE CAPABILITY MATRIX describes the Resources process capabilities. Combined with the logic in the scheduling system it forms an economical way of describing Alternative Route data and allows the scheduling system to cope with loading to flexible resources. This is a "Preference Order" which allows the user to define the suitability of the resources to each task. The resource may be the first choice for more than one process.

In this matrix the RESOURCE DEPENDENCY DATA are also defined as which primary and secondary resources are used together e.g. machines and men. For example, if the primary resources (called by the routings) are machines and the secondary resources are men, the RESOURCE DEPENDANCY DATA is effectively data on employee skills.

5.3.4.5 Works Order Priority

The priorities are based on algorithmic processes which define the relative priority of each Works Order on the basis of some of its characteristics. Several prioritisation algorithms can be defined so that the planner may use one of these to prioritise the Works Order before a scheduling run. It should be possible for the planner to "freeze" the priorities on a range of Works Orders before re-prioritising the remaining Orders, perhaps with a different algorithm. The algorithms to prioritise the Works Orders should be independent of Orders. These are:

1. First In - First Out (FIFO). This will prioritise on either the Works Order number (if the planner maintains a system for sequentially numbering WOs) or on the "data first scheduled" recorded on the WORKS ORDER DETAILS. Several Works Orders may share a date first scheduled so that the logic would take the lowest Works Order number as the highest priority.

2. Critical Ratio. Each works order will be given a priority. These can vary say between 1 and 99. If two priority constants A and B are defined, the following relationship may be used to generate a suggested priority.

$$P = ((\text{Days Required}+A)/(\text{Days Available}+B)) \times 100$$

   This method will have the option of including Purchased Leadtimes for consumed materials in the calculation of "Time Required". Option to calculate Least Float per remaining operation on critical path should also be available. Days Required is the difference, in days, between the due date and the latest start date.

Days Available is the difference, in days between the due date and the earliest possible start date.

3. Least Float with Short Works Order Preference. This is a second method of determining the priorities as shown below;

P = (A - B)/(Days available - Days required) - C/10(Days required)

if P <= 0 then P = 1

if P > 99 then P = 99

A, B and C are the priority constants. Using this method can give overriding priorities to shorter jobs by use of constant C.

This method will also have the option of including Purchased Leadtimes for consumed materials in the calculation of "Time Required". Option to calculate Least Float per remaining operation on critical path should also be available.

4. WIP Holding Cost. The relative WIP holding cost will be calculated using the cost of the consumed materials, the work that is required to make each item in the Works Order and the hourly rate of the top preference Resource for each operation. The logic is similar to calculating the cost of a particular product using a BOM "Cost Roll-up" process, but the costing from the process planning function which calculates the cost and time of each operation in the routing can be used. This will lead to orders with expensive materials or processes with high priorities.

5. User Defined. This will require the planner to set up a file and define a number of parameters associated with the Works Order and establish a relationship that suits his purpose. This file can contain a number of combinations between these parameters and pick up one for each Works Order.

6. Option to pick up priority from user defined priority on Sales Orders.

The facility to freeze priorities may be based on two reasons:

I The Planner wants those Works Orders to be held as the most important orders and so all subsequent prioritisation of other orders must yield lower priorities than those in the frozen group.

II The Planner wants the relative priorities of those orders to be retained, although it does not matter if other orders are prioritised into that group.

The frozen priorities should remain until the Planner runs a separate program to unfreeze all or some of the frozen group. The range of orders to be frozen may be:

a) Works Order numbers range
b) Priority range
c) Range of values for any of the data fields within the Works Order header details.

There should be facilities for manual review and editting of priorities.

5.3.4.6 Resource Loading

The method of loading each resource is to take each Works Order in priority sequence and load the operations of its items to the available resource. Here, three methods of loading are considered.

- Backward Loading
- Forward Loading
- Short Interval Loading

A device (Changeover Logic) is needed so that when a method is violated according to some rules the system is switched over to the next method.

Changeover Logic

A table or file is created with a set of decision rules and processes that are common to both the Backward and Forward Loading system. These rules are then used in conjunction with a changeover logic, which means that when a validation stage within the Backward loading logic identifies that an operation is infeasible the system changes over to that of Forward Loading (see Fig. 5.9).

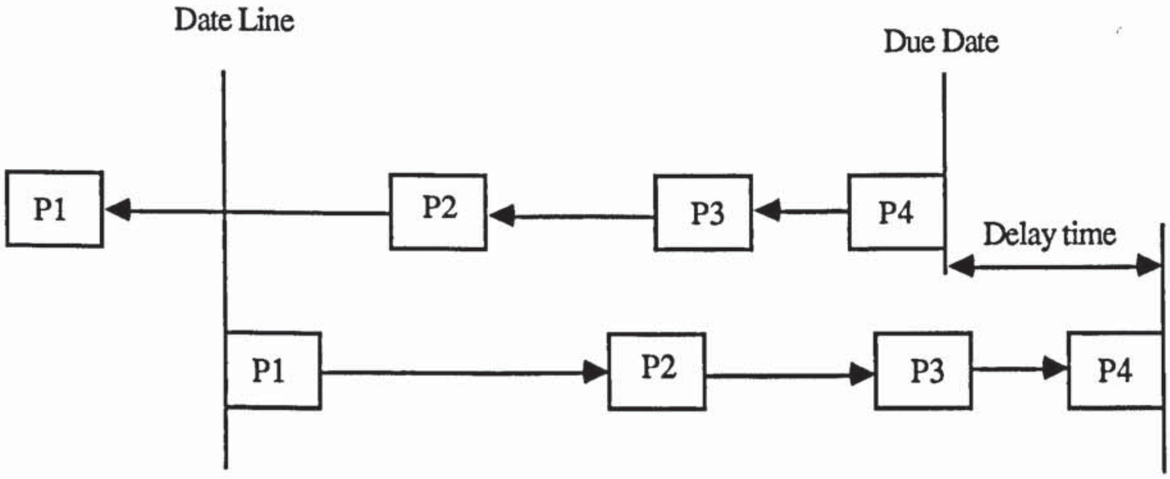


Figure 5.9 The Backward and Forward Loading Diagram.

The Backward Loading logic tries to load as many items on the Works Order as possible. When ever part of the loading becomes infeasible, the changeover logic should:

1. Note the loading changeover point and all items and operations that should have been loaded before that point in the routing or product structure. This information can be subsequently reported.

2. Prevent further loading of the operations and items before the infeasible loading.

3. Allow the Backloading system to run on for items in other parts of the routing to see if any other branches of the Works Order routing will be loaded infeasibly (see Fig. 5.9).

4. Within the scheduling rules and parameter file if the scheduling indicator is set to "Backward and Forward" the following occurs: when the whole Works Order has been dealt with, the Changeover Logic will clear the Works Order from the schedule so that the Forward logic can be used to reaload the Works Order feasibly. This involves unloading the operations for manufactured items within

a Works Order from both the WORKS ORDER DETAILS and RESOURCE LOADING CALENDARS.

5. If the scheduling indicator is set to "Backward" the works order is left loaded to the extent that has been feasible, and the scheduling logic then tries to load the next Works Order using the Backward loading logic.

Backward Loading Logic

The Backward loading Logic is used first in the scheduling system. It attempts to load each Works Order backward from its Due Date (see Fig. 5.9). The Logic will have the following stages:

1. Choose the highest priority.

2. Determine its Due Date and the last operation on the final operation.

3. Determine which Resources that Operation can be loaded on.

4. Choose a Resource Timeslot to load the Operation into.

5. "Load" the operation by noting the Scheduled Start and End Dates (SSD,SED) and Times and the Resource code against the item on the Works Order.

6. Check that the loading is feasible within the following criteria that the Scheduled Start Date (SSD) should not be before the date of scheduling. If the loading is infeasible use function CHANGEOVER LOGIC.

7. See if any stocked materials are planned to be consumed at the operation. If so, check for that material allocated to the Works Order by some other (CSP or Stock Control) system. If material is allocated adjust the allocation date to SSD for the operation. If material is not already allocated, allocate material from the stock system.

8. Both of the actions in 7. (modify the allocation date or allocate material) should be validated using the following criteria:

   a) Is the material (allocation) available at the time required ?
   b) If the material is not available, is there sufficient time between the date of scheduling and the required date to cover the leadtime of the required material ?

   If the loading proves to be infeasible because of material supply constraints use function CHANGEOVER LOGIC.

   If however, the loading is correct, repeat this process from 3. for the next (previous) operation on the routing.

9. When the whole final item has been loaded then chose the next manufactured item to be loaded from the items described on the final item's PARTS LIST or PROCESS PLAN.

10. Determine the Required Date and Time for the next item from the final items's loading. This may be the SSD for the first operation of the final item. However, it might be the SSD of some other operation on the final item if the parts list specifies that the supporting item is consumed at a later operation.

11. Repeat the loading logic for the next item starting with the final operation on the item's routing. Start this loading process from 3. above.

12. Repeat the loading process for all items on the Works Order. When the whole Works Order has been loaded, select the next most important Works Order for loading. If there are two Orders with the same priority, choose them in Works Order Number sequence. Repeat this process from 2. above.

<u>Forward Loading Logic</u>

The Forward Loading Logic will load the operations of manufactured items within Works Orders as early as possible. It will be used to load items that cannot be loaded feasibly using the Backward Loading Logic. After removing the Works Order using the CHANGEOVER LOGIC, the system should reload the items as early as possible.

The way of choosing the items to load will incorporate elements of Backward Loading Logic.

1. The system will load the Works Orders starting at the lowest level items and gradually loading higher levels of operations.

2. Each item should be loaded starting at the first operation.

3. It is assumed that the main priority will be to load items as early as possible. Consequently, the system will not attempt to delay the loading of items with respect to the depth of the routing. So, if two "branches" of the routing go to different levels of complexity, the system will not treat them differently. For

example, one branch might have six levels of assembly but the other only has three. The lowest level items of both branches would be loaded at roughly the same time. This may mean that some items on the shorter arm of the routing will be loaded too early and will be held at WIP untill the other work catches up. The alternative is a very complex Loading Logic which is not discussed in this project.

4. Determine the earliest date that the item can be loaded within the available material - in the form of allocations from the Stock Control system.

5. Nett the item's requirement against stock: i.e. see if any of that item can be supplied from manufactured items held in stock or allocated to the Works Order at the time for ideal loading. Adjust the requirements if some or all of the requirement can be satisfied from stock.

6. Start by loading the first operation on the item's routing. Select a primary Resource timeslot as early as possible.

7. Check that there will be a secondary resource available to support the chosen primary loading.

8. Load the operation by noting Scheduled Start and End Dates and Times on the primary and secondary Resource Calendars and on the operation details on the Works Order.

9. If material is allocated to the item at that operation, adjust the date on those allocations to the SSD for the consumuing operations.

10. Choose the next operation to be loaded and calculate the start date using the logic described in 5. and 6. above.

11. Continue the loading process for all operations on the item.

12. Check that there are no free stock items available at the Schedule Finish Date (SFD) for the item - if there is - allocate and reload the item.

13. When the item has been loaded, repeat the loading process for all other items.

14. Repeat the above processes until the complete Works Order has been loaded.

Short Interval Loading

Short Interval Loading (SIL) will allow the Planner to manipulate a small section of the automatically generated schedule to primary resources. This can be represented graphically in the form of a "Gantt Chart" and embody all of the "parent-operations" and operation sequencing with dependencies described within the routing. It can be possible, therefore, for the Planner to see the loading visually (see Figure 5.10) and over-ride such dependencies after suitable warning messages.

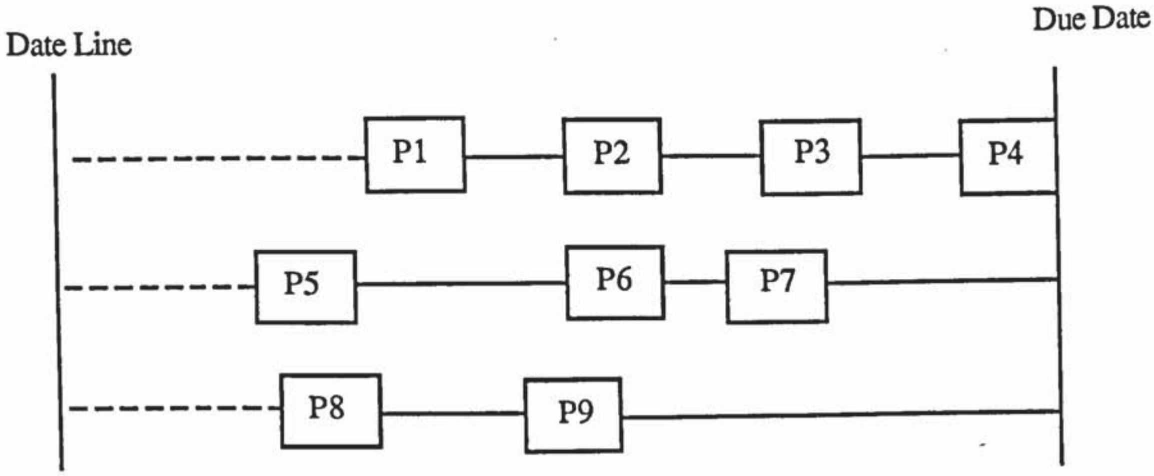


Figure 5. 10 The Gant Chart Schedule Diagram.

This will allow the Planner to move, combine, split and reroute operations. The Planner will also be capable of freezing the supervised schedule before subsequent automatic scheduling of the unsupervised work with the normal logic; hence a perfect tool for "what if" analysis.

5.4 Subcontract Control

Some of the reasons for sub-contracting are:

- The overloading of available machines
- Creation of bottle necks
- Requirements for operations for which no machines are available
- Infeasible operations

Subcontracting can be carried out at any stage of operation and in general at the top level assembly or subassembly or component level. However, the subcontracting should be treated exactly the same way as purchasing functions. However, in some cases the raw material supplied to the subcontractor may be free issue and the company just pays for the labour work. The Works Order Processing function should be capable of providing functions similar to the Purchase Order System and it should also have functions where the same part may be subcontracted from one subcontractor to another.

5.5 Manufacturing Control

The function of Manufacturing Control, as the name implies, is to control the manufacture of goods to meet the agreed MPS, to input and maintain live Works Orders that satisfy the MPS, to initiate and record the movement of stock to meet those orders, to decide upon the physical sequencing of work on the shopfloor to meet those orders, to monitor and record the actual work done on those orders and any other shop floor activity.

As shown in Fig. 4.3 this function has been split into three sub-functions:

- Works Order Processing (WOP)
- Work In Progress (WIP)
- Stock Control System (SCS)

The first two processes are discussed below while the third process SCS is explained fully in conjunction with the functionality of CSP in Chapter 6.

5.5.1 Works Order Processing

It is intended that orders will come through the full Sales, MPS, Process Planning and the CSP or MRP and Commitment Acceptance procedures. If so, then all the data would be there for processing. However, provision must be made for manual entry. This section does not include physically issuing materials or receiving goods, these are the province of SCS functionality. WOP should include the allocating of materials, the creation of WIP, layout of process routes (routings) and costing of processes and finally printing of shop floor documentation.

The Planners should be able to place Works Orders, delete Works Orders, amend and accept Works Orders and process them. Any Works Order may contain several Sales Orders as a result of the CSP procedures where orders are mixed together to provide material optimisation. Sub-Works Orders must therefore be raised where necessary to account for the batch spliting of the orders and eventually individual orders at the packing level.

5.5.2 Work In Progress Control

This function covers the monitoring of shop floor activity. It starts once materials have been issued to the shop floor, and ends when goods are ready for receipt back into stores or have been scrapped off. Both boundaries are therefore with Stock Control. The function essentially breaks down into three areas:

1) Machine Monitoring (MM) - recording the status and usage of shop floor machines and processes.

2) Shop Floor Data Capture (SFDC) - recording the status and usage of shop floor employees.

3) WIP tracking - recording the status and progress of works orders.

The full system should record the time of data entry (and hence the time taken) from its own clock as it happens, the quantities from regularly polled machine sensors, and additional information from bar coded documents. It will validate the data as it is entered and provide two way communication between the SFDC terminals and the host computer. It will, however, be possible to poll data on a less frequent basis (i.e. in batch mode) at the user's discretion. In addition the system will be capable of working without validation or two way communication should the host computer be unavailable for some reason. This is to ensure that a computer hardware problem will have the least chance of stopping the recording or data. The system will be capable of being "detuned" in several ways to allow:

a) WIP tracing with SFDC but without MM, the only difference bing that the user keys in the quantities produced and scrapped,

b) The full system without monitoring what each employee does (to cater for process industries).

c) WIP tracking with neither SFDC nor MM, the user keying in all the data either as it happens or after the event (e.g. from a time sheet).

To allow the planners to maintain processing rules, library information and system utilities, the following information must be maintained.

For machine monitoring - machine stoppage codes
- inactivity time before stoppage assumed.
- clearing machine registers.

For SFDC - employee diversion codes
- transaction types for SFDC terminals

The general processing rules and options relating to WIP Control are maintained in a file which can be displayed, changed or printed as follows.

- print MM transaction log
- print SFDC transaction log
- MM polling frequency
- SFDC polling frequency
- SFDC processing frequency
- approval required before processing
- operation late finish tolerance
- action time tolerance
- WIP time and quantity down date basis

Where real time monitoring of machines is involved, any stoppage or down time needs recording against valid reason codes. This is to allow the planners to maintain a table of such reasons. They would be used when a machine or process is stopped for longer than a time specified by the planners.

Since the application of Machine Monitoring requires an indepth study of machine monitoring techniques and customised software in a highly automated environment, the discussion of this subject is out of the scope of this project.

The use of Shop Floor Data Capture in the rolling mill environment is essential in controlling the flow of material and WIP tracking and control. Therefore, several transaction types are necessary to provide the planners with enough data to be able to reconcile the status of any works order.

Where real time data entry of shop floor activity is required, this will be via transactions entered by the operators through SFDC terminals using a sequence of questions and answers. This function is to allow the planners to define those transaction types.

Full sequence would be to define the transaction, select the SFDC terminal(s) it is relevant to, repeat as necessary and download the information to those terminals. In order to save time and effort, the system should come with a library of transactions covering standard use and common or expected variations already created.

The system should provide facilities allowing planners to configure terminals by providing an editing facility that defines the required parameters and then downloads a suitably formatted file to the terminal.

1) Split the transaction into three parts - activity, employee, machine and allow entry in any sequence.

2) Allow repeated entry of any part of 1) during a transaction (e.g. several employees booking onto one works order).

3) Define a set of default SFDC transactions, but also keep several different versions of each transaction with some of the parts in 1) inferred from previous transactions or pre-defined (e.g. machine number for a terminal dedicated to a particular machine or process).

4) Validate the data entered on-line to reduce the chance of errors being rejected later.

Some of the transaction types necessary in this industry are: Start, End, Cancel, Shift Change, Scrap, Enquire and Block Transactions. For each transaction the operator has to enter his employee number, product code and the reason for this transaction. The terminal should work the operator through a sequence of questions and be able to provide a help facility if required.

For the system to make use of the data entered via the SFDC terminals, it is necessary to poll SFDC terminals for transactions, validate them, check for irregularities like operations taking too long, print a log of transactions and an exception report if necessary, to allow manual correction of bookings and create data to update WIP.

This facility therefore can be used to update the WIP data file for work carried out on the shop floor. The works order and operation number are flagged as complete with the operation quantity, time and all other parameters as necessary.

The two following methods can be used to calculate the time and quantity to be posted to a part-completed operation.

Method 1 - The transaction having the actual time and quantity completed by the operator. The amount of planned time consumed by the part-completed operation can therefore be calculated as:

Planned Time consumed = (Qty to date/Total Batch Qty) Total Planned Time

This method is used when the "Time and Quantity Reduction Basis" in the WIP Rules or on the Works Order or on the Item is set as "Quantity Made".

Method 2 - This is aimed at jobs with small batch quantities and long operation lead times, where the quantity made to date is difficult or impossible to estimate. The operator does not enter a quantity completed and the planned time consumed to date is assumed to be the same as the actual time consumed. If the actual time reaches or exceeds the total planned time, the operation is assumed to be almost complete untill the operation is finally closed. If the quantity completed exceeds that planned then a manual transaction will be needed to incraese the quantity for tha next operation.

5.6 Summary

The functional specification of the manufacturing management system required to satisfy most of the problems specified in Chapter 3 are discussed in depth. Although the teminologies used in this chapter are of general form, the applicability of these terms are compared in the later chapters.

Section 5.2 identifies the manufacturing rules applicable in any manufacturing industry and the need for the establishment of a central data base system to accomodate for these rules and facilities to modify them when necessary. There are two important functions which are global for each function; a) user definable formulae (an example of this is given in Chapter 10, Section 10.3) and b) the system data file (this is the setting of dates, tollerances, acceptance limits, global changing of some values, part numbers, etc.). The overall functionality of this module is to set up and update a number of data base files which contain the company policies and manufacturing rules of practice. Whilst, Section 5.3 discusses the working operation of manufacturing planning and processes and establishes the MPS, RCCP and the scheduling policies.

A number of working logics have been devised to assist the scheduler in generating a feasible loading for the machines according to the manufacturing rules, resources and constraints already defined. A module called the Changeover Logic is defined so that once the system has scheduled a job, it examines the scheduling policies implemented (i.e. forward loading, backward loading, short interval loading), order parameters and loading rules, if any of the defined constraints are violated the Changeover Logic will switch over to the next preferred logic.

Section 5.5 controls the manufacturing activities of the shop floor to ensure that what has been planned is being manufactured and the works orders raised on the system conform to the shop activities. Hence control is maintained over the progress of works orders, work in progress and the stock control system.

The implementation of shop floor data capture is vital in the control and of the manufacturing activities of the operators. However, this is only achieved if the operators stick to the codes of practice by entering the correct product bar codes and quantities produced at each work center. As discussed, this effort can be eliminated and be automated by the implementation of a machine monitoring system so that counters are installed at the entry and exit of each work centre to determine the count and report to the shop floor data capture the status of every works order.

CHAPTER 6

DEVELOPMENT OF THE MANUFACTURING SUPPORT FUNCTIONS AND STOCK CONTROL SYSTEM

6.1 Introduction

Having developed the functionality and requirements of a Manufacturing Management system, mainly for the rolling mill industry, the function would not be complete without the development of a stock control system and the inclusion of the Computer Aided Material Optimisation (CAMO) within a frame of Manufacturing Support. The idea is that the functions within CAMO would utilise and mix Sales Orders. Techniques of Cutting Stock Problem are used to select the most cost effective stocks to satisfy these orders. Once the stocks are selected, the stock control system would be updated to include the selected stocks and mark them as allocated for a given batch of orders. In the next batch of trial orders the allocated stocks will not be included in the search methods of CSP, unless the planner decides to mix all order batches. In this case, all allocations will be deleted and CAMO will start afresh.

The results generated by CAMO would be similar to that of MRP, Orlicky (1975).It suggests how much material is needed and suggests purchase orders and works orders, applicable to the Rolling Mill environment.

Manufacturing Support, (see Fig. 4.4) is intended to provide the Manufacturing Management Function with necessary information. This requires a large amount of calculation and specialisation which is not readily available through the normal conventional systems. These include the Time & Attendance Recording, Tool and equipment Management, Process Planning, CSP and third party Interfacing and finally

Plant Management. In all these areas specialised skills are required to update the system and maintain its functionality so that the Manufacturing Management system can readily use the data provided by this system for its purpose.

In this Chapter Manufacturing Support functions are briefly discussed and then the development of a well maintained Stock Control System SCS is established. Thereafter, the concept of CAMO is integrated within the SCS and Manufacturing Planning concept to give the system the full functionality of an MRP system within the Sheet Metal Industry.

6.2 Manufacturing Support

Most computerised systems are based on a closed loop environment Show (1984). This means that once committed to a manufacturing system then it is very difficult to interface with other third party software where the functionality of those specialised systems in most cases outweighs that of the implemented system. For example, MAPICS, a Computer Aided Production Management System (CAPM) developed by IBM mainly for the mainframe machines system 3Xs, contains its own Sales Order Processing, Purchase Order System, Works Order Processing and a whole host of other manufacturing functions designed and developed by IBM for general use by the users. However, if a user in, say, an export marketing environment tries to use the SOP module and generate the export documentation required by the Customs and Excise and the Shipping Liners, the company is forced to pay a considerable amount of money and spend much effort in developing the requirements inhouse with IBM approval. This obviously can be very expensive, as most bespoke packages are.

However, the British government has developed a computerised system called SITPRO SPEX which satisfies all the requirements of such bodies and provides the user with a fully detailed documentation of such matters. Hence, it relieves companies of spending hours in generating these documents and avoids confusion over the print format of these very complex forms.

Unfortunately, most computerised manufacturing systems do not provide the facilities for interfacing with third party systems and the data transfer between these packages is extremely complex and expensive.

However, the standard functional modules proposed in this system would be as follows:

6.2.1 Time and Attendance Recording (TAR)

This function is concerned with recording the attendance at work of company personnel, reasons for absence and reconciliation of clocked to booked hours. It is aimed primarily at shop floor operators. It could equally be applied to staff, supervison and management if desired. It is concerned with keeping track of who is available and what those people do once they are at work, so that production activities can be triggered and scheduling and machine loading commence. It is envisaged that data collection terminals would be used for clocking and reporting job status on the shop floor.

The rules regarding TAR vary considerably from company to company, in general the following facilities should be available:

- Overtime Authorisation
- Automation, Semi automation or Manual clock-off
- No. of days consecutive lateness for reporting employee to supervisor

The case studies at the two rolling mill companies showed that they authorise overtime in advance. This allows them to keep control of how much is being worked and why. Therefore, a list of such overtimes within the system should be catered for and should be used to warn the user of employees who are working unauthorised overtime.

Companies also have different rules regarding clocking on and off. Many insist on employees clocking on and off at each break. Others, only at lunch, and some only insist on clocking on. Automatic clocking will allow the company to define which rules are applicable to them according to shift patterns. The system will then use them to automatically clock employees on and off where relevant.

Many companies categorise time spent by employees, so that different pay rates can be applied in Payroll. Hence a table is required to allow them to maintain such payment categories. It is not to define the payment calculation itself (that is normally done in Payroll). A typical reason for this would be to define premium factors e.g. Overtime, or a different rate for diversion work etc.

A file is maintained to update the shift patterns used by the company. The full sequence would be to create a number of daily patterns, which are then used to build weekly patterns, which in turn build into yearly patterns. These can then be used later in conjunction with absence codes to build up expected attendance pattern for each employee.

The absence code provides a full picture of each employee's time requirements, non attendance or absence to be recorded as well as attendance. This is to allow the accountants to set up and maintain a table of absence codes to describe the reasons why employees are not or will not be at work. Typical reasons would be holiday, sick leave etc.

Finally, after the development of a whole host of other supporting functions which should be organised at the time of the development (such as clock on/off record, undefined clocking, clocking correction, etc.) the information should be booked against the jobs designated. The SFDC data should only be passed to the relevant function if it has been checked, and agreed by the production controller. Once, processed correctly, if desired, a payment code can be added and the information may be read into the Payroll.

6.2.2 Tool and Equipment Management

This section is concerned with allowing the Planners and Setters to carry out the following:

a) Maintain a database of tooling records to identify descriptive details (i.e. jigs, fixtures, tapes, cutters, rollers, inspection equipment etc.), stock and cost information together with their type (e.g. consumable, finite life, permanent etc.) and where applicable the expected life and refurbishment cycle.

b) Maintain a link between tooling and product.

c) Calculate tooling requirements - This should have provision in the CAMO or Scheduling system.

d) Maintain a tooling inventory and location record.

e) Maintain a tool usage record and calculate refurbishment requirements.

f) Record tool refurbishment and calibration activity

g) Produce tooling reports and enquiries.

A tool supporting information file is maintained to allow the Setters to update rules and library information relating to Tool Management. Typical libraries would contain a register of tooling, products where they are used, pre-setting instructions and usage calculations. The Setter can define various formulae so that they would take into account the Works Order quantity and unit time, severity factor etc. and generate a tool register which specifies which tooling the calculation is applicable to. Some companies, such as Sandvik rely on calculations based on Taylor's therories of cutting tools.

Finally there should be a link to the Process Planning BCAPP or SCAPP to allow the planners to define the relationship between tooling and Production Routing Operations and/or Parts Sequencing plus any preparation work required before the tooling can be used on the shop floor. This information can be used for TOOL REQUIREMENTS ANALYSIS and prepare tools for shop use. This function is particularly useful in some of the orders received by the jobbing shop industry, particularly at both companies A and B, where tooling is issued operation by operation.

6.2.2.1 Tool Requirements Analysis

This should be similar to MRP principles which analyses the tooling requirements of Works Orders. The Planner has the choice of analysing live, simulated or both sorts of Works Orders using MRP techniques or PRODUCTION SCHEDULING dates. This function will also highlight any new or consumable tooling that will need to be bought or made and any non consumable tooling that will need refurbishing or is a potential bottle neck.

The system should categorise and report on the following information:

i) Tool allocations required to meet Works Orders

ii) Suggested/Simulated orders for new tooling

iii) Predicted bottle necks brought about by insufficient tooling

iv) Tool life predictions

In addition, the system should highlight tooling stock profiles that would result from carrying out the above.

6.2.2.2 Tool Inventory Contol

This is to monitor Planned and actual stocks and usage of tooling. Esentially this should be the same as the Stock Control System function. Tool allocation and usage should be traceable and must be linked to the relevant Works Order. The allocations should be carried out at the time of placing the Works Order. The non-consumable tooling should have a number of transaction codes placed so that the tooling can be returned to stores at the end of production.

The return of tooling should include return to stores, hirer, machining centre, setter, regrind or calibration.

The allocation of non-consumable tooling for any particular Tool Register Number could be designed as follows:

PART NUMBER: 1234X2389 22

Trans Code	Doc. No.	Qty	O/SQty	Trans Date	Due Date etc.
ALD	WO100	2	2	010189	050189
RTN	WO100	2	2	010189	050189
ALI	WO100	2		050189	050189
RTD	WO100	2		050189	100189

The AL (ALocateD) transaction is placed when there is a requirement for that tool to be used on Works Order number WO100. At the same time an RTN (ReTurNable) transaction is placed indicating the tool is returnable back to its departure place. At the time of raising a Kitting List for the tooling and issuing the tool to the shop floor, the transaction ALI (ALlocation Issue) is placed against the AL transaction. Hence each ALD should have a matching ALI. When the job is finished and the tool becomes redundant it should be returned to the stores and on receipt of the tool the transaction RTD (ReTuRned) is entered to cancell the RTN transaction.

6.3 Process Planning Interfacing

This function should be developed primarily for linking third party packages. However, since the Process Planning aspect of this system was developed by Husseini (1989) as a separate project using the ProLog language, it seemed appropriate to provide this link for communication with the software. The Cutting Stock Problem is, however, coded in C and the database system for the whole project is assumed to be ORACLE or INFORMIX with a 4GL tool driving the user definable reports and analysis.

As already defined in Chapter 3 Section 3.4.6, the process planning system SCAPP and BCAPP cover the following areas:

1) Grouping of Sales Orders by Base Supply.

2) The detailed planning of orders.

3) Material availability for each route.

4) Calculation of overall costs.

The application of the Process Planning function should be transparent to the Planners and will require the following types of links.

a) SCAPP or BCAPP to have access to read certain parts of the CAPM database.

b) To be able to update and provide necessary information regarding routing for Works Orders.

c) To provide the necessary sequence of operations data for CAMO.

There are two types of data which SCAPP and BCAPP need to have access to.

1) Data that will be used directly to build up the Process Plan. The Process Plan should be capable of accessing any part of the CAPM system. Some of these include:

   a) Item Stock Data
   b) Item QA Data
   c) Item Master Data
   d) Item Cost Data
   e) Item Leadtime
   f) Quality Standard Data
   g) Text Library

2) Data that is read to validate the Works Order Header. This Provides raw data if any, from the previous Process Plan in the library if the job is strictly standard. The system checks for a Production Routing match. If this exists, the system displays details for the Planner to proceed or exit. It also provides the Process Planning function with all the necessary information regarding the cutting pattern by which a piece may be cut and the tooling requirements.

6.4 Stock Control System

This function is the heart of any computerised manufacturing system, since without components and parts nothing can be made. In most engineering environments almost every part is given a number or code, hence coding and classification methods have been developed and been researched for years Love (1986). Furthermore, the introduction of this function brings about the control for accuracy of physical stocks, movements to and from those stocks and it provides an audit trail to those movements. It also serves to advise stores personnel of future stock requirements and highlights any problems. Many papers and publications and have been developed, Thomas (1980) and Morrison (1970); but very few address the problems of stock control systems at printing shops or rolling mill industries, where unique components, assemblies and sub-assemblies do not exist.

However, it is rather difficult to introduce part numbers in the Rolling Mill environment since the life of each part is very short and the dimensions can vary infinitely. Therefore, in this section, a method is proposed by which parts are grouped together according to their gauges and metal codes and techniques of providing facilities to take advantage of the above statements are introduced.

Some of the most fundamental requirements of a comprehensive stock control system in the rolling mill or similar industries are:

1) Control Stock Movements
2) Control Planned Movements
3) Highlight Planned Movements
4) Carry Out Planned Issues
5) Receive Goods
6) Action and Record Unplanned Movements

7) Barcode Input

8) Manage Stock Records

9) Audit Stock Accuracy

10) Carry out Cyclic Checks

6.4.1 Control Stock Movements

This function initiates, controls and monitors activities relating to planned and unplanned stock movements. All movements should be associated to a process which changes the stock data, even scrap would be recorded since it is a result of a process which would have been documented, relating to how and when it was produced.

Since any type of stock item can be kept in several parts of the shop floor a Multi Location facility is required to keep track of where the parts are.

The following apply to all transactions within the Stock Control System.

a) Issues against Multi Location items must force entry of a valid Location. Receipts must warn if the location does not exist and allow its creation. The creation must be authorised to prevent creation of too many locations by everyone.

2) Series of transaction codes are defined as follows:

i) ALD - ALocated quantity.

ii) ALI - ALlocation Issue.

iii) RTN - ReTurNable to store.

iv) RTD - ReTurneD to store.

v) ALC - ALocation Cancelled.

vi) ALA - ALocation Amended.

vii) ISS - ISSues to shop floor.

viii) ORD - ORDer on shop floor, subcontractor or purchase.

ix) REC - RECeived ordered item.

x) TRN - TRaNsfer to store unplanned "free" items.

xi) ORA - ORder Amendment.

xii) ORC - ORder Cancelled.

xiii) SHE - SHortagE quantity

xiv) SHI - SHortage Issue

xv) ADJ - ADJustment

When a Works Order or Purchase Order is raised for a particular stock item i.e. slabs from a refinery, an ORD transaction is placed against that item with ALD transactions against any other item that goes to make the Slab or finished good. On receipt of such item/s an REC is automatically used to indicate the transaction type. Hence ORD is always satisfied with an REC.

When a Works Order is raised for the lower level items or components an ALD is placed to indicate the quantity required of that item for the Works Order. Therefore, when the items are issued from the stores, ALIs are placed aginst these items with the quantity issued. Hence an ALI can only satisfy an ALD.

If the shop floor require any quantity over and above the quantity that was planned to be used, an ISS is used to indicate excess quantity (which is normally referred to as scrap).

If the quantity of a Works Order is changed the, ALA transaction code is used to indicate this change at the lower levels and ORA at the top level. If the Works Order is cancelled ORC transaction is used at the top level with ALC at the lower level items.

RTN and RTD are mainly used when dealing with tooling equipments i.e. a non consumable tool will be issued (RTN) and is expected to be returned (RTD) back to stores.

TRN transaction is used when an item is either transferred to stores or is purchased without an official order. Although this very rarely happens it is required such a transaction should exist.

The SHE and SHI are used to show the quantity short of what the Works Order asked for. In most cases the work is done on the Works Order up to the point where no further work can be done as a result of these shortages. When the items become available then the SHI transaction is used to satisfy the shortage.

3) Traceable item movements must record the Issue number as well as the item number.

6.4.2 Control Planned Movements

This is to control and monitor stores activity relating to planned stock movements. The planned movements would include:

a) Receipt of items on a purchase order i.e. raw materials, items bought out complete, subcontract work. Transaction type REC to satisfy previously placed ORD.

b) Issuing of kits of items (ALI) on a Works Order to the shop floor.

c) Receipt of goods from completed works orders (ALI and REC).

d) Despatching of finished goods to the customer.

6.4.3 Highlight Planned Movements

This is to check what stock movements are planned within a specified time period and highlight for further action those currently possible and impossible. i.e. to highlight those items with an outstanding kit shortage transaction.

The planned stock movements could contain:

a) Orders Due Report - To show orders (works and/or purchase) due within a specified time period. This needs to show Orders Number, Supplier, Item, Due Date, Scheduled Due Date, Quantity, Unit Price, Total Value, Comments, etc. The system would provide reports on overdue orders, summary or detailed reports with deliveries already made, complex sort and totalling required, suppression of cost fields if required.

b) Goods In Inspection Report - To show items received for inspection but not released with the required dates. This should have selection criteria via a security code so that only authorised staff can get reports, e.g. waiting inspection, waiting laboratory tests, waiting certificate of conformance, etc.(see Item g below).

c) Trial Kitting List - To identify the works orders that need kitting and issuing within a certain time period, the current stock/order position on each item in the kit, highlight potential shortages. Reports are generated to show: Works Order Number, Item Number, Description, Quantity and Issue Date Required, Stock On Hand, On Order, In Inspection, Allocated, Commited, Batch No., Etc. Calculate potential kit shortages and number of items.

d) Trial Despatch Report - To allow the Planner to identify saleable goods due to be despatched to customers. Similar in concept to Trial Kitting List only restricted to outstanding Sales Orders and will also be able to show those with a QA release status. If carried out as a screen display the planner needs to be able to flag orders accepted and automatically produce a Kitting List using the information.

6.4.4 Carry Out Planned Issues

This function is to carry out planned issues from stores to the shop floor, subcontractors or to the customers. It is required to produce paperwork to enable someone to find the items and pick them off the stores shelves and to record issue numbers, batch numbers and quantity. The quantity column should have an additional column to allow for the excess issue of components such as bandoliers, gas, sheets, bars, etc. which have an expected return quantity.

The system should be able to print material requisition for each individual works order or sales order, or picking list in item number or location sequence for a range of works and/or sales orders with common items summated.

All the planned issues should be recorded either to works orders or sales orders, converting units of measure as necessary. It should also produce the necessary paperwork to accompany the items issued, both for the shop floor, subcontractors and despatches to customers.

6.4.5 Receive Goods

This is mainly the raw material in the form of scrap metal or slabs in the Rolling Mill industry; and in the form of parts or sub-assemblies in others. The receipt of goods must be checked against a valid order to make sure that the details are correct. For purchase orders, paperwork should be produced (i.e. a Goods Received Note (GRN) or inspection card) to monitor and control the goods through inspection (if necessary) and into stores. The actions required in this function are:

- Check Order Details
- Record Planned Receipts
- Over Delivery Actions
- Produce Receipt Paperwork
- Quarantine and inspection
- Update Inspection and QA Records
- Release Items to Stores
- Reject Goods to Supplier

6.4.6 Action and Record Unplanned Movements

This function is to action and record unplanned issues and receipts and requires a great deal of dicipline within the stores. It needs to be particularly easy and staightforward, otherwise operators may be tempted to neglect carrying out this function. The requirements of this function could be as follows:

Check Stock and Location - This is when a request is made to receive or issue items only from a particular location and check frequently if the items and locations are up todate. It should be very easy for the operators to pick and place items quite easily from their location.

Record Stock Movements - This will allow the operator to update item records and audit trail these for unplanned receipts or issues. This will include block or individual transfers between stock locations or contracts.

6.4.7 Barcode Input

This is a separate set of duplicate functionality to enable the input of selected data via a bar code reader. The capability of this function will depend on whether the company is prepared to install the data collection hardware capable of having bar code facilities. The uses for this facility in a fairly modern environment are endless and below are a number of areas in which this facility can be used for control.

Barcodes Stock Movements - Movement of stocks in stores where the number of stock transactions are high. These include:

> Stores Office Transaction - Input of issues and receipts data at the stores office via a wedge data terminal.
>
> Portable Transactions - Material issues recorded during picking in the stores.

Barcodes in Stock Checks - Portable terminals to be used for collecting stockcheck data.

6.4.8 Manage Stock Records

This is to audit all transactions and ensure the accuracy of the stock data. The system should highlight items that are at the end of their shelf life and bring out the items which were redundant. The housekeeping and correction of transactions are dealt with through this option.

6.4.9 Audit Stock Accuracy

This again is to ensure that the stock transactions and stock levels match that of the real physical stock levels in the store. Stock verification, cyclic checks and stocktakes are means of acheiving this objective. Therfore, the system should be capable of producing the relevant paperwork and documentation to cope with this requirements.

6.4.10 Carry Out Cyclic Checks

The cyclic check is extremely important in achieving reliable stock figures and for the system to generate realistic material allocation and material planning. This function selects items that need stock checking, records stock on hand/location details, changes the status of the items (where necessary) and produces a listing according to the operator's selection criteria. Stock rules must be set up so that during normal stock checks forced stock items are also included e.g. negative stock on hand (highly undesirable). The selection criteria may include:

1) Range of items including WIP in store.
2) Range of locations.
3) Limit the number of items chosen.
4) Range of suppliers.

These selections must be compared with the physical stock on hand, and corrected via an ADJ transaction. Various report must be available in order to compare the stock counts against the system stock and highlight the following reports:

1) Exception Report - i.e. item numbers where stock count exceeds the tolerance and therfore requires a recount.

2) Variance Report -By value and quantity.

3) Discrepancy - Quantity x Cost.

There should be annual stocktake similar to the function above except selection is not necessary. The operator should specify the date of this event. Shop floor WIP would also need stocktaking.

6. 5 Design of Stock Master System

The Item Stock Number would contain two files, one which contains all the header details and another which contains the transaction details. The header details of the item data is designed in two tiers:

1) The common entity header
2) The header attributes

6.5.1 The Common Entity Header

This is specifically designed to cater for the requirements of stock control Raw Materials and mill stock with direct interfaces to purchase order, works order processing and CAMO. It should incorporate features that allow the stock transaction codes to be accessed by the following varying criteria:

- Transaction code
- Material type
- Material size
- Gauge

The full functionality of search routine would have to be specified in detail at the time of configuration.

The following header card is proposed as the common entity.

STOCK NUMBER: 123X456AA 789 DESCRIPTION: Half Hard 450k Cu Slabs

---

DIMENSIONS:

UNIT OF MEASURE .. mm
GAUGE 150
WIDTH 1000
LENGTH 1500

QUANTITIES:

STOCK AVBLE 152
STOCK ON ORDER 98
ALLOCATED STOCK.. 12
SHORTAGES 0
FREE STOCK 140
STOCK IN QC 0
RE-ORDER LEVEL 100
RE-ORDER QTY........ 500
OTHER STOCKS 46

MISCELLANEOUS:

PURCHASED ITEM .... Y
BULK ISSUE ITEM .... N
EXCLUDE FROM CAMO .. N
LEAD TIME (WEEKS)....... 4
SUPPLIER CODE .... ALCOA
NO. OTHER SUPPLIERS... 8

HISTORY:

ISSUES LAST YEAR 58765
ISSUES THIS YEAR 45674
ISSUES LAST PERIOD 23456
ISSUES THIS PERIOD 12658

COSTS:

LATEST COST 450.00
LMATERIAL COST 0
OVERHEAD COST 0
LABOUR COST 0
AVERAGE COST 455.00
SELLING PRICE 650.00
VAT CODE 1

PRODUCT DETAILS:

UNIT WEIGHT .... Kg
PRODUCT GROUP CODE 4000
LOCATION L23DY
BIN NUMBER DD21
MAX. DEC. PLACES 2
LAST STOCK CHECK DATE 22.03.89
ABC CODE .... A
FREE ALLOCATION .... Y
STANDARD STOCK ITEM ... Y

The above would be the screen enquiry, with the information held on each stock record.

The description of some attributes are given below:

STOCK AVBLE	This is the complete physical stocks available in the stores.
STOCK ON ORDER	This field is updated whenever a purchase order is place via the purchase order system. As goods are recieved, this field is reduced to reflect what is outstanding and the STOCK AVBLE is increased by the same quantity.
ALLOCATED STOCK	This field shows the quantity of stock allocated to existing orders. This quantity is not deducted from STOCK AVBLE field since the FREE ALLOCATION field is set to Y. The transaction report shows the detailed analysis of this field.
SHORTAGES	This is the quantity that CAMO assumed available, but in reality it did not exist. This can only happen if STANDARD STOCK ITEM is set to Y which means tells CAMO "include this item in the calculation of CSP, even if you run out of this item.", and at the end of the run this quantity is reported in the SHORTAGE column.
FREE STOCK	This field shows the quantity of stock items after all the allocations, whether the allocations are set to free or not. Therefore: FREE STOCK = STOCK AVBLE - ALLOCATED STOCK.

RE-ORDER LEVEL RE-ORDER QTY	Both these fields are used by CAMO. When the shortages exceed the RE-ORDER LEVEL CAMO will check the RE-ORDER QTY field and suggests a purchase order in multiples of the quantity specified by this field.
OTHER STOCKS	This field is particularly useful since it is very rare that in a rolling mill environment, the planners would plan specific dimensions of a stock. If there are any other stocks with the same gauge, but different widths or lengths those stocks are also considered as a viable raw material. Therefore, this field shows how many other stocks with varying dimensions have the same gauge.

Other fields of the master header are self explanatory, and therefore, will not be discussed.

6.5.2 The Header Attributes

The transactions associated with any stock records should be displayed in the following format.

PART NUMBER: 123X456AA 789 DESCRIPTION: Half Hard 450K Cu Slabs

----

Trans Code	Doc. No.	Qty	O/S Qty	Trans Date	Due Date
ORD	AL1234	1000	98	12.01.89	12.02.89
REC	AL1234	500		10.02.89	
ALD	WO321	18	12	19.02.89	22.03.89
ISS	WO012	100		09.09.89	
ALI	WO321	6		20.02.89	
REC	AL1234	402		21.02.89	

*** End Of Transaction Report ***

Above is a listing of some of the transactions that could take place. The fields specified would have to be designed more comprehensively.

6.6 Summary

The manufacturing support system is designed to provide the manufacturing management function with additional features which may be optional to the operation of the system. This module also provides the user with the facilities to interface with any third party packages to import data to the system.

The standard modules discussed in Sections 6.2.1 to 6.2.2 provide the means of monitoring the availability of labour force and allocating them to vailable resources. The sections also explained the integration of a comprehensive tool management system and tool inventory to allow the planners to issue the necessary tooling required to satisfy the works orders.

Section 6.3 explains the reason for the development of the interfacing functions to the third party packages. The CAPP system developed by Husseini (1989) and the CSP system developed in C in this thesis would be communication to each other through this function.

Section 6.4 provides a detailed analysis of the implementation of a Stock Control System SCS. The system is designed to be split into two sections, the stock master header and the stock transaction file. The latter being the attributes detailing the general status of any particular stock and the former the log of all the issues and returns.

The chapter provides all the necessary information required for the Computer Aided Material Optimisation (CAMO) to make full use of the SCS and input the correct amount of stocks to the Cutting Stock Problem module for the generation of the cutting patterns. Using the lead times and the latest cost fields CAMO would be able to report suggested orders and plan material according to the demand.

CHAPTER 7

DEVELOPMENT OF COMPUTER AIDED MATERIAL OPTIMISATION (CAMO)

7.1 Introduction

Chapters 4, 5 and 6 provided a detailed description of a modern manufacturing management system. One of the most important components of the proposed system is the Computer Aided Material Optimisation module. CAMO, together with the Stock Control System (SCS), sit at the heart of the CAPM system in a rolling mill environment. This system is thought of as the Material Requirements Planning MRP function of a conventional CAPM system. In this chapter, the underlying concepts of MRP are discussed and comparisons are made to the functionality of CAMO, how it works and its benefits.

Entire books have been written on the subject of MRP, Orlicky (1975). However, it is not the objective of this chapter to explore the many facets of MRP, details of which are beyond the scope of this thesis, but to draw comparisons of fundamental points with that of CAMO, so reference is made to the main functions of MRP concept.

MRP has developed over the years to become more than just material requirements planning. Wight (1981), uses the term 'MRP II' to represent manufacturing resource planning, a system for planning and controlling the operational, engineering and financial resources of a manufacturing company. Many authors, Anderson (1981), Wallace (1985), Fisher (1981, 1986), Kochhar (1986), Orlicky (1975) and Fox (1983) have tried to explain the concepts of such a system and the implementation philosophy adopted in various research work. One such philosophy is the approach by Love and Barekat (1988). They argue that although MRP II is an appropriate tool to carry out the production planning and control functions of CIM (Computer Integrated Manufacturing), the current hierachical 'TOP DOWN" approaches are not appropriate. A method of "Distributed 'BOTTOM UP' approach" to both the system's implementation and operation methodology is presented.

Furthermore, as explained in Chapter 6, MRP is normally implemented in an area where product structures exist and systems to generate the Bill Of Materials (BOM) is one of the fundamental elements to the working of MRP systems. However, Bill Of Material is non-existent in the rolling mill industry where products are changing shape at all levels of operation. In a typical example, one may start with two copper or steel castings and roll these down to a predetermined gauge and break these down for further rolling. Therefore, by the time the customer orders are satisfied, the products are separated into hundreds of small pieces which is totally the opposite of the situation in normal engineering companies (e.g. automotive industry or any assembly plant).

However, with the development of the process planning system SCAPP and Cutting Stock Problem CSP techniques, a method of using the process plans and the generated cutting patterns to simulate the activities of an MRP system is presented.

The process planning routes, once generated for each order, are used as the BOM and the decentralisation methods introduced by Love and Barekat (1988) are used to evaluate the material requirements at aech levels of operation.

7.2 Material Requirements Planning

MRP is a computational technique that converts the MPS and existing sales orders for end products into a detailed schedule, identifies the quantities of each raw material and component item and determines when each item must be ordered so as to meet the due dates specified for the final product.

Groover (1987) discusses the working of MRP "... The master schedule provides the overall production plan for final products in terms of month-by-month or week-by-week delivery requirements. Each of the products may contain hundreds of individual components some of which may be common. For example, several parts may be produced out of the same sheet steel. The components are assembled into simple subassemblies. Then the subassemblies are put together into more complex assemblies, and so forth, untill the final product is assembled together".

7.2.1 The Inputs to MRP

There are generally three sources of input data on which MRP relies:

1. The master production schedule and the current order book.
2. The bill-of-materials file, which defines the product structure.
3. The inventory record file.

In the case of MRP II an additional input is the Rough Cut Capacity Planning RCCP, which is a file defining the machine capacity and shift patterns.

Figure 7.1 represents a diagram showing the flow of data into the MRP processor and its conversion into useful output reports. Figure 7.2 shows a typical bill of material for a component. The discussion of the above three inputs are presented by Groover (1987) and Orlicky (1975).

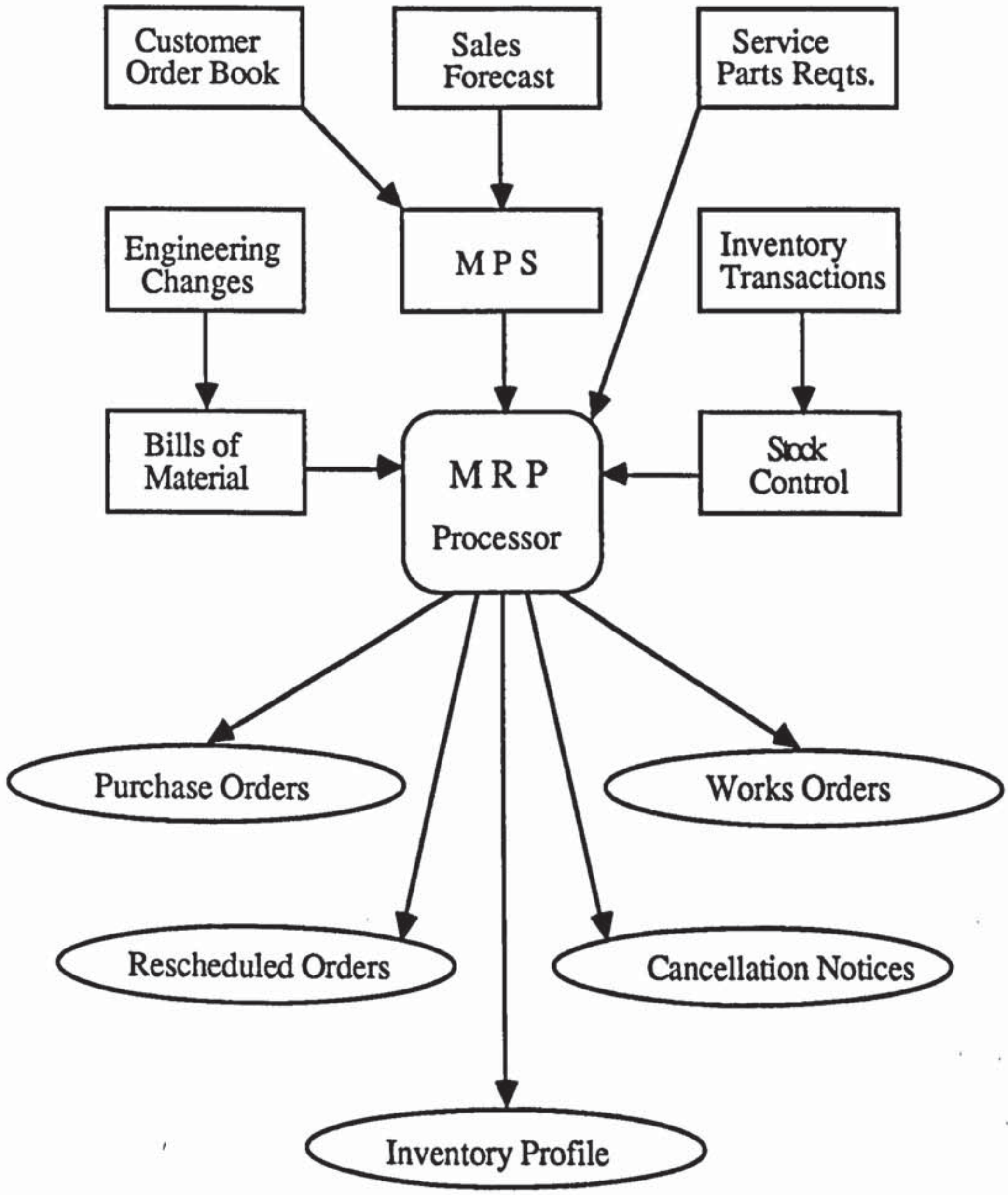


Figure 7.1 Structure of a Material Requirement Planning (MRP) system

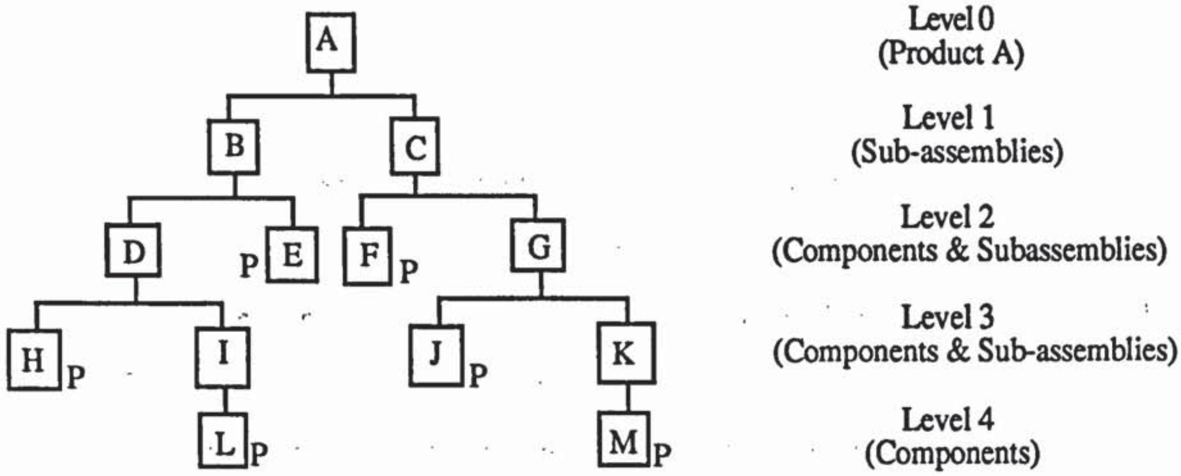


Figure 7.2 The Structure of the Bill Of Material (BOM)

7.3 Centralised MRP System Approach

Conventional manufacturing control systems adopt a centralised approach to the planning process, Love and Barekat (1988). This is especially evident in MRP II systems in which a single system is used to plan and control every manufacturing related activity within a plant. All the relevant data is held within one complex system. The system must answer the needs of all manufacturing activities within the plant and therefore, offer a broad range of facilities and options which serve to continually increase the complexity of the system.

Love and Barekat also argue that the centralised (and global) nature of the system removes much of the key decision-making from the shop floor but is nevertheless critically dependent upon timely and accurate feedback of shop performance data. System performance rests upon good communications between the shop floor, where knowledge of plant status and capability is greatest, and a high level management.

7.4 Concept of Decentralised MRP Systems

Love and Barekat (1988) have studied the Kanban philosophy, and consider it as "... a decentralised system". The system is designed to detect and solve problems locally, with all the key resource control mechanisms operating at shop floor level. They then combine MRP with Kanban to produce a partially decentralised solution. These systems employ MRP to control the procurement of bought-out parts whilst Kanban is used to organise material flow inside the factory.

With the concept of Group Technology (GT) approach, Love and Barekat have considered the cellular systems and have appointed the cell manager responsible for all activities and objectives of the cell performance. These may include budget for the cell, planning, scheduling and control of the cell and reporting requiremnts to the top management.

The cellular structure therefore, makes possible the development of a distributed form of MRP II. This concept offers a capacity sensitive alternative to conventional planning systems. Each cell is provided with its own small MRP II system hosted on a local computer and linked to all the other cells and company functions via a local area network (see Figure 7.3).

The information held at each cell is:

1. The relevant product bills of material.

2. The cell only holds the levels between the input and output stages of the item produced by the cell.

3. Inventory records relevant to the cell.

4. All routing data concerning the manufacture of the part while in the cell.

The individual cell plans its own requirements using essentially the same tools found in a conventional MRP system.

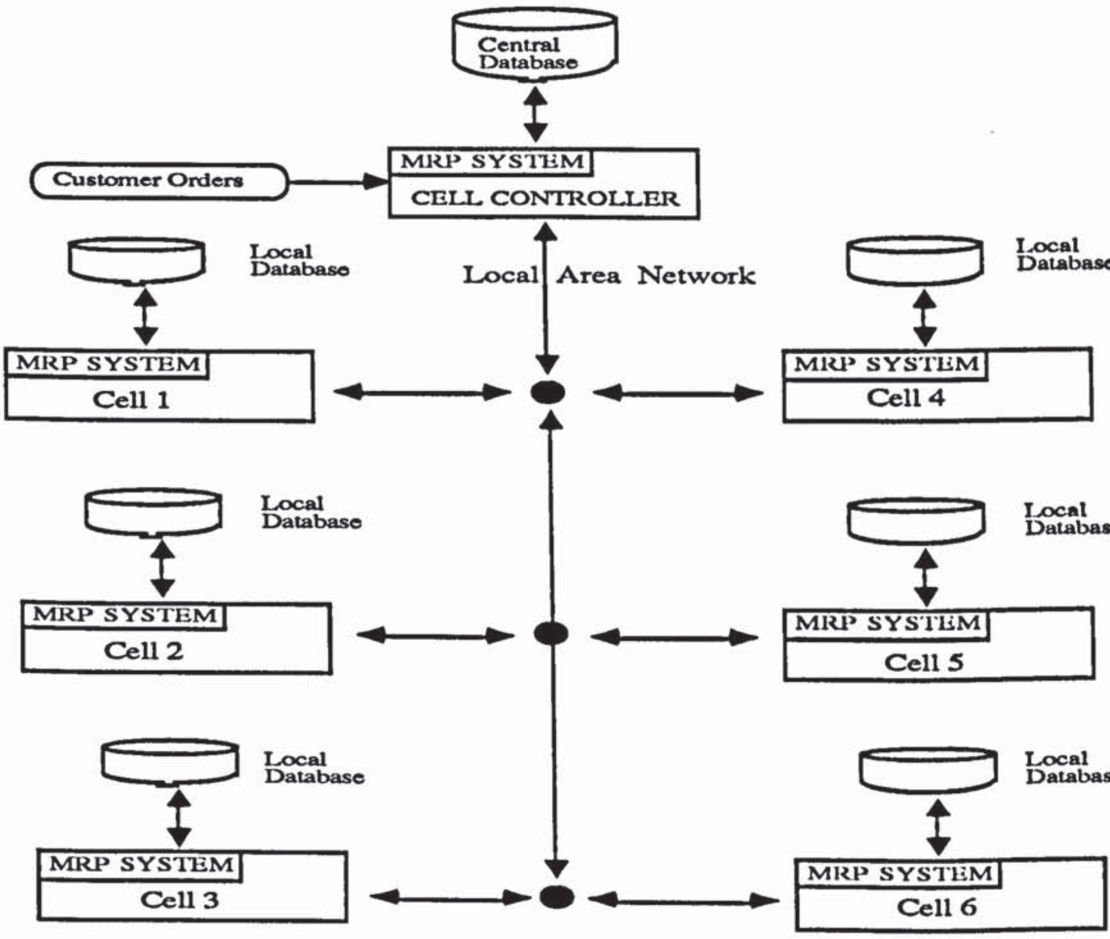


Figure 7.3 Data communication in a distributed MRP system Love & Barekat (1988)

7.5 Functional Specification of CAMO

Having presented the overall features of both centralised and decentralised MRP systems, the approach is used to propose a system for the rolling mill or paper industry very similar in concept to that of decentralised MRP systems.

The three inputs explained in Section 7.2.1 are redesigned to bear practically the same functionality and input to CAMO as they would in an MRP system. These are explained below.

7.5.1 The MPS and the Order Book

Since there is very little forecasting in the rolling mill industry, (or at least at the two companies studied in this thesis) the main effort is concentrated in the use of the confirmed orders. There are also a large number of enquiries (or estimates) given each day which usually result in confirmed orders. These together with the confirmed orders may be used as one of the input to the Master Production Schedule (see Figure 7.4).

The MPS as explained in Chapter 5 Section 5.3.1, receives the RCCP details from the process planning module SCAPP. The RCCP would have to group the three processes of Section 5.3.1 form the process plans generated by SCAPP.

The estimate file would have a weighting factor field, so that regular customers who are likely to return with their estimates as confirmed orders may be given a wighting factor from 0 to one by the Sales Department. The MPS in turn would give higher priorities to those estimates with high weighting factor. In all, there are three inputs to the MPS function.

1. The order book file
2. The Estimate file
3. Rough cut capacity planning file

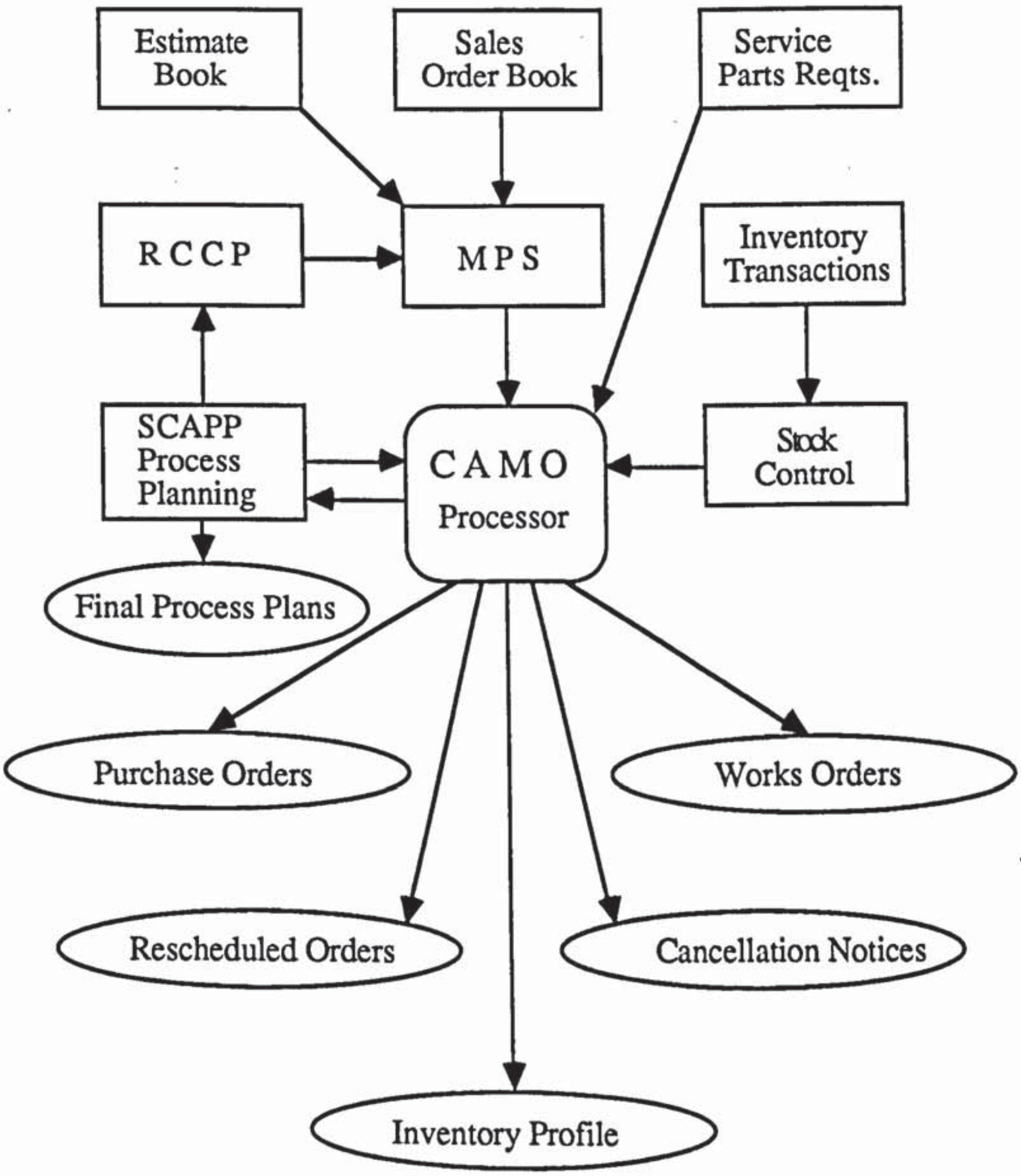


Figure 7.4 The structure of Computer Aided Material Optimisation (CAMO) system

7.5.2 The Bill Of Material Equivalent

The bill of material is considered the most important input to the MRP system.. Here, the bill of material equivalent is the most important input to the CAMO system. As shown in Figure 7.4, the SCAPP process planning collects the orders from the common data base shown in Figure 4.5. SCAPP generates the process plans using the five stages as described by Husseini (1989) also (see Chapter 3). Using the planning logic described in Figure 7.5, SCAPP generates the process plans shown in Figure 7.6 (a-b).

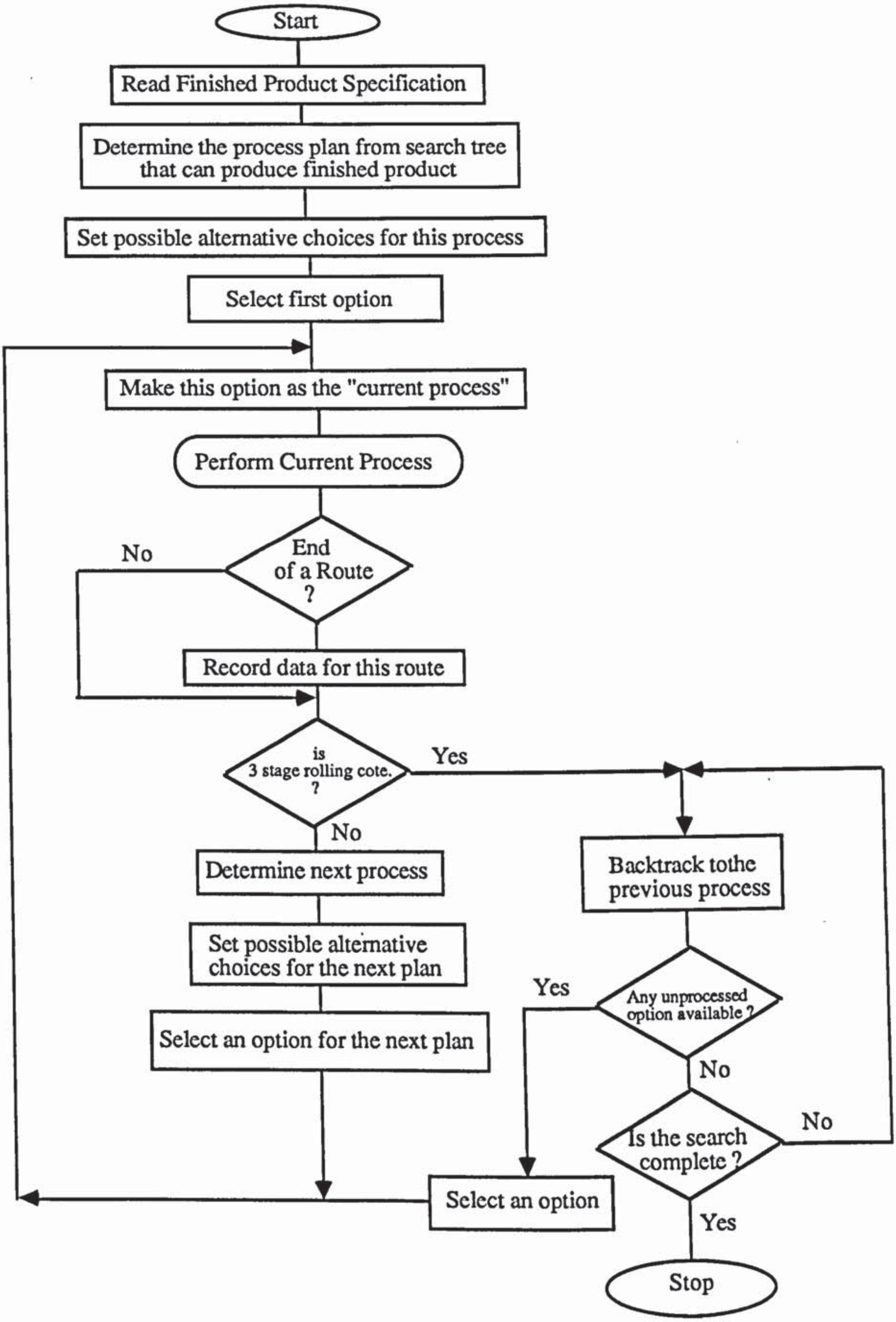


Figure 7.5 The Process Planning Logic Used by SCAPP.

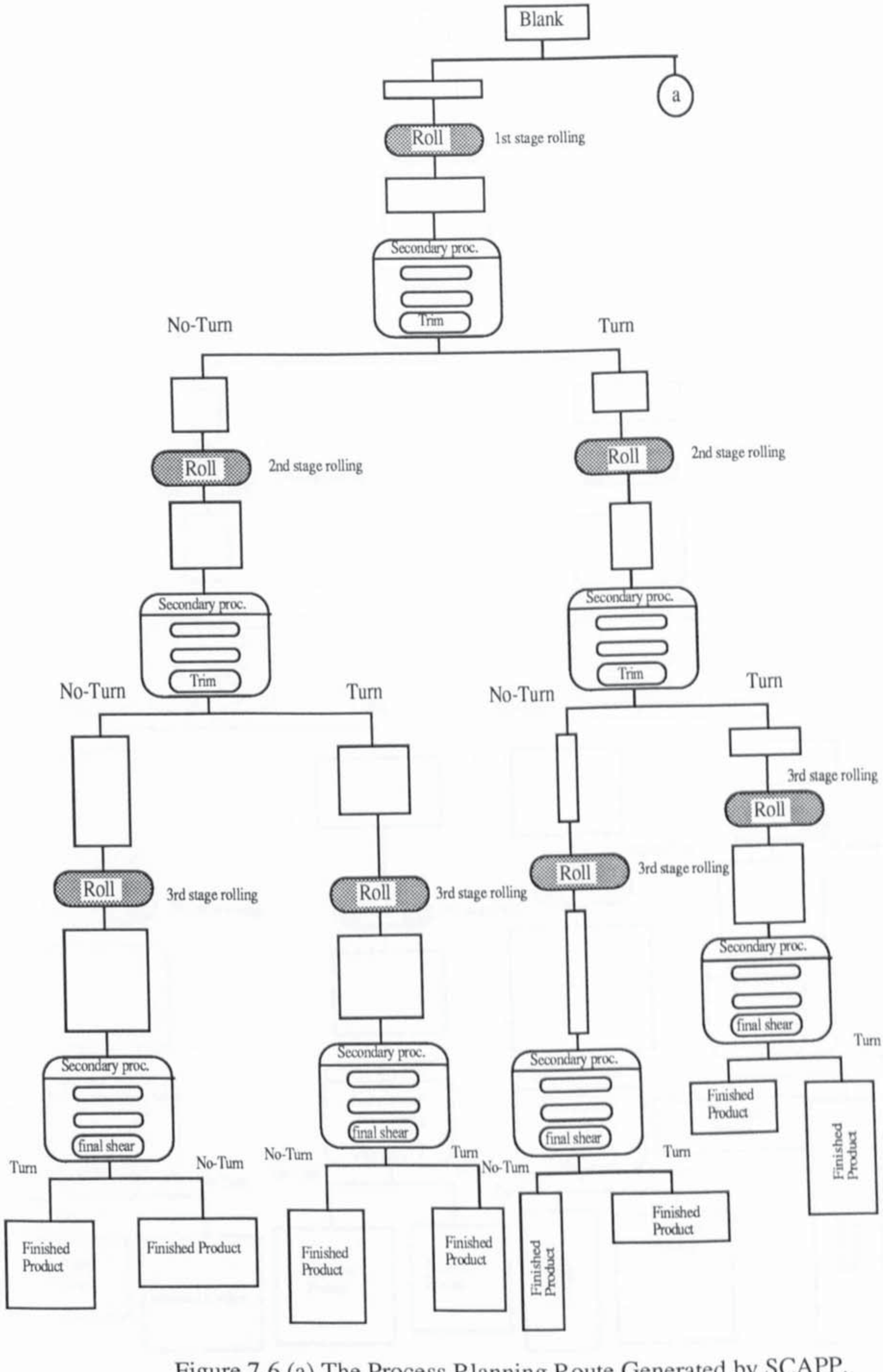


Figure 7.6 (a) The Process Rlanning Route Generated by SCAPP.

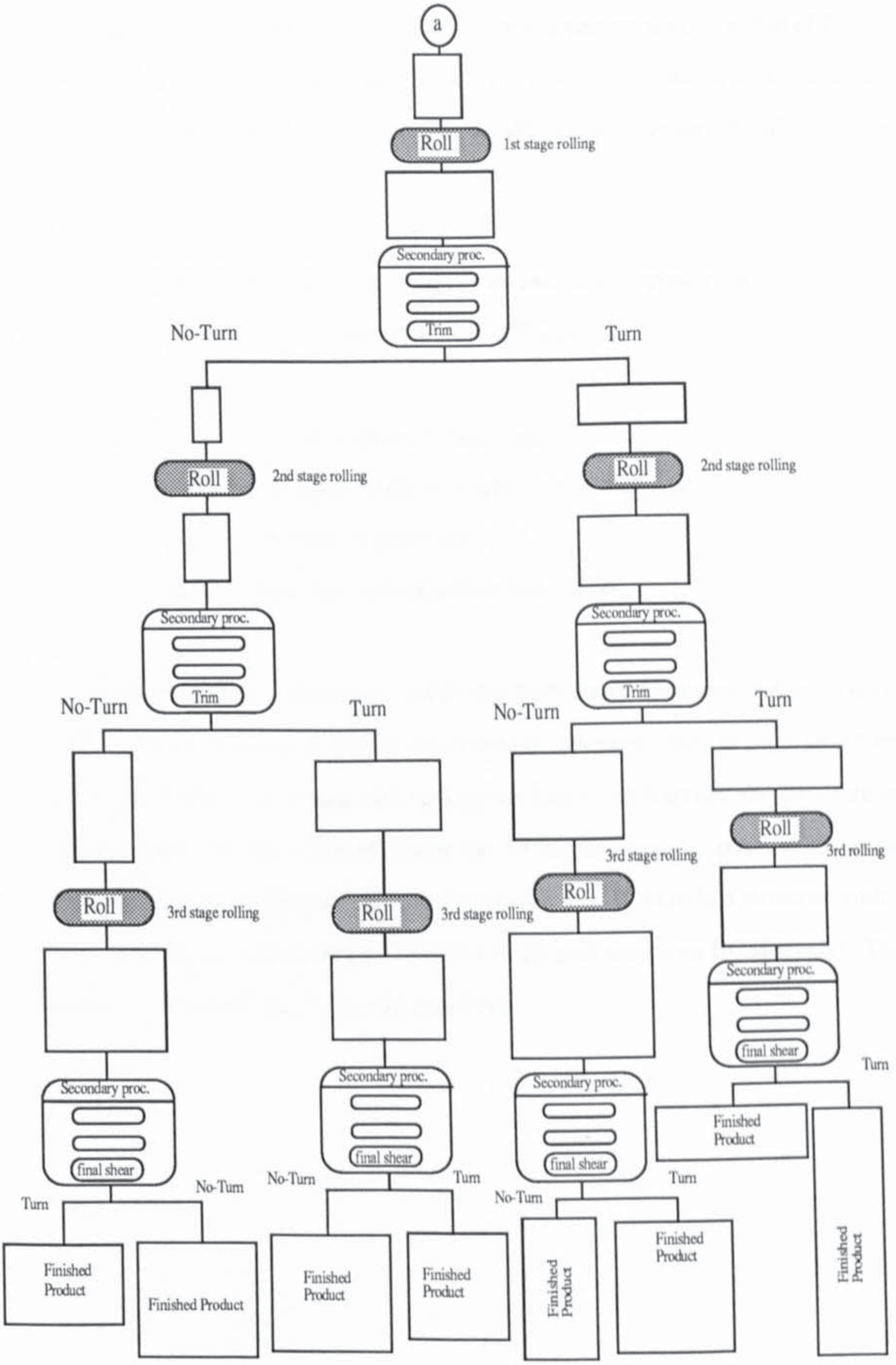


Figure 7.6 (b) The Process Planning Route Generated by SCAPP.

Analysing Figure 7.6 (a-b) further, it can be seen that a pattern similar to that of Figure 7.2 can be drawn. This is depicted by Figure 7.7. Therefore, it can be seen that every time an order is entered in the order book file, SCAPP would generate a feasible process plan and input this to CAMO.

CAMO in turn would collate all process sequences and 'superimpose' these to generate a table which summates the following and shown in Figure 7.7.

a) Common gauge at each stage
b) Common width or length at each stage
c) Number of pieces required
d) Due date (or Latest Start Date LSD)

It can be seen by Figure 7.7 that many orders may have the same process routes. This is equivalent to bills of material having the common sub-assemblies in their structure. Using the distributed bill of material analogy by Love and Barekat, the structure of process sequences for any batch of orders can be broken down to produce the same number of structures as there are levels of explosion (i.e. if there is a structure with 4 levels of assembly, then there should also be 4 items with their own BOM levels). This is depicted by Figures 7.8a, 7.8b, 7.8c and 7.8d.

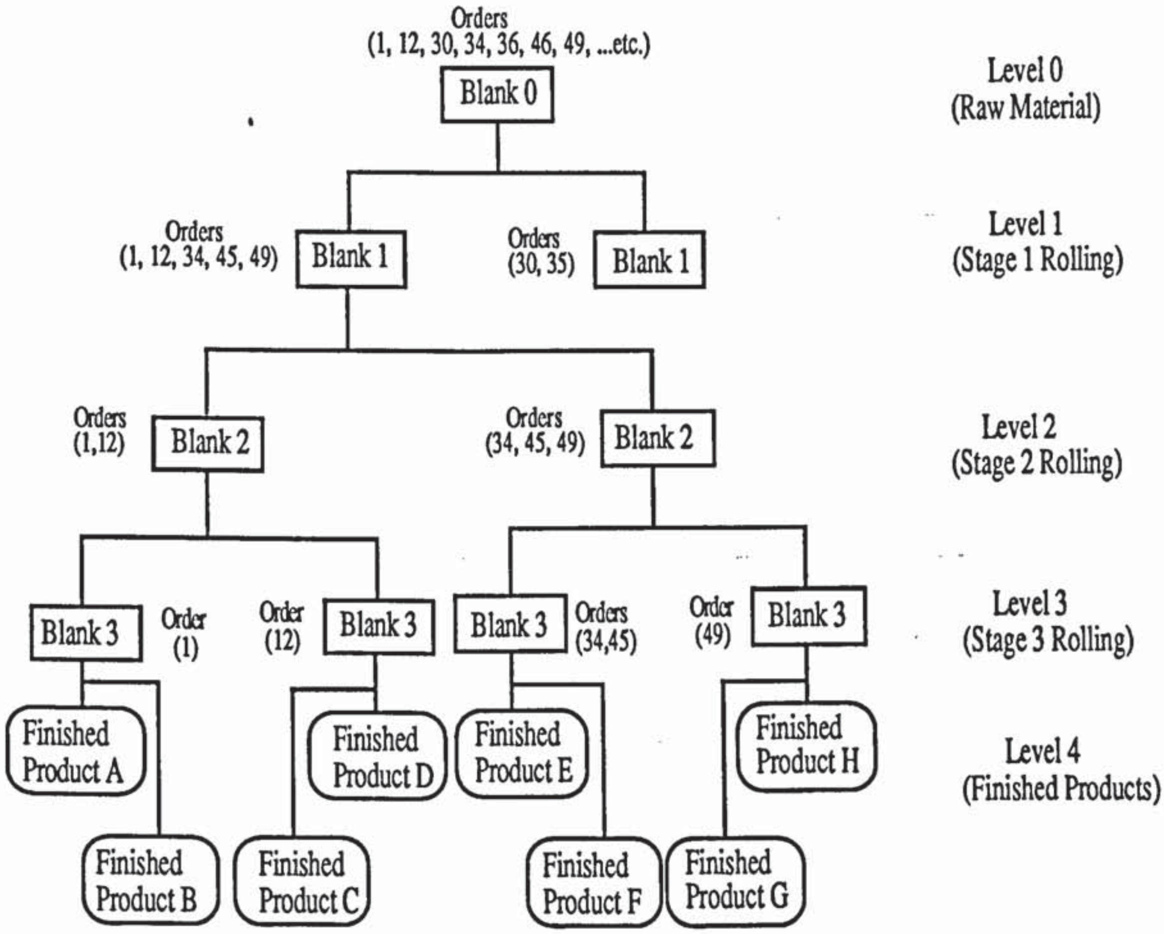


Figure 7.7 Product Structure Similar to the bill of material

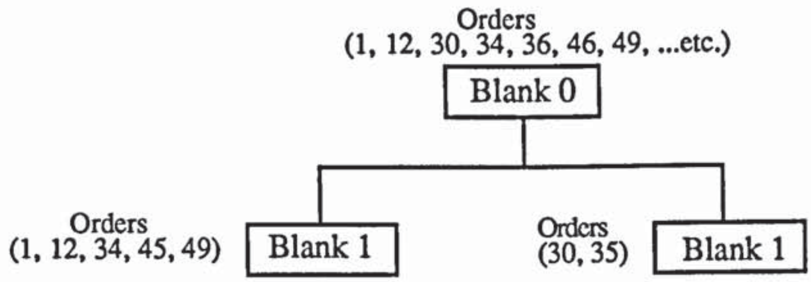


Figure 7.8a Stage 1 Rolling Structure

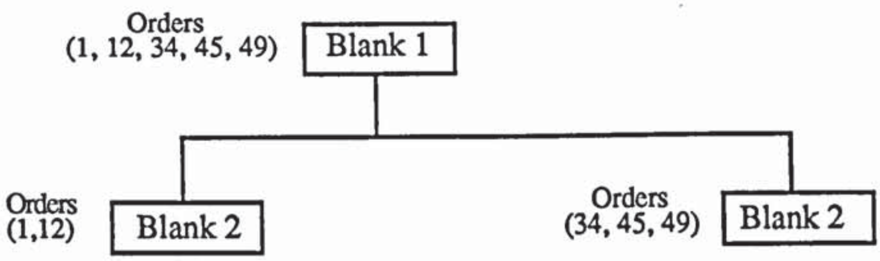


Figure 7.8b Stage 2 Rolling Structure

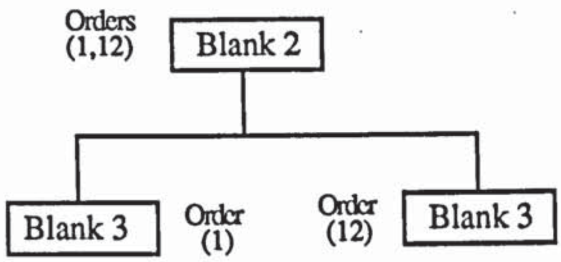


Figure 7.8c Stage 3 Rolling Structure

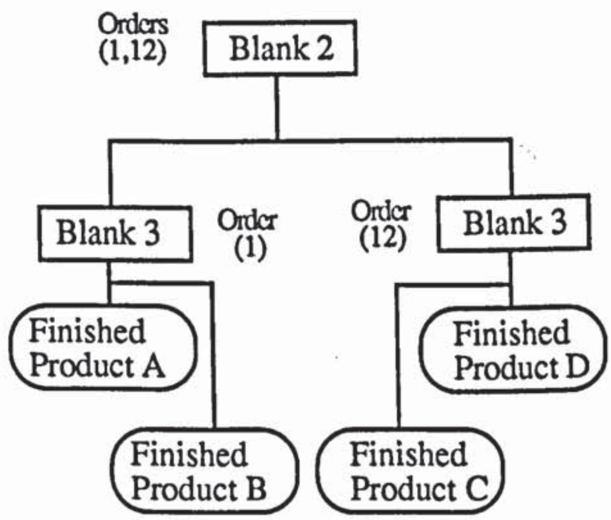


Figure 7.8d Stage 4 Roll to Finish Structure

7.5.3 The Inventory Record File

A comprehensive stock control system has already been designed to provide the CAMO system with the necessary information regarding:

- Cost of the stock or slab being considered.
- Gauge of the stock and other stocks that have similar gauges.
- The current allocated stocks.
- The current free stock for further allocation.
- Stock availability.
- Stocks that are allowed to have shortage status.

The intermediate rolling stages may or may not have stocks available. If there are no stocks on hand, then there would also be no lead time to consider since the middle stage rollings are not bought products.

When the final, calculation of material requirement is complete the system would out put the resulting stock profile to SCAPP and make allocations within the stock control system if the allocation field is set to Y.

The total allocated stock is then offset against the stocks on hand. The balance is stored in the suggested order file with the lead time taken from the stock master record.

The manufacturing lead time is determined from the process route sheets already established by SCAPP. Suggested works orders together with their suggested due dates are stored in the works order file.

7.6 How CAMO Works

The material optimisation processor (see Figure 7.4) operates on the basis of data contained in the MPS, SCAPP and the Stock Control File. The MPS specifies the list of all current, estimated and orders forcast in either a weekly or monthly period. The SCAPP defines what process routes or sequence of operation each order is required to go through to achieve the final product. SCAPP also defines each stage at which rolling takes place, hence suggestion for converting the state of the product from one dimension to another, the common gauge, width or length required for that stage is also provided. The stock control system contains details of the current and future status of each item. It also contains details on the stock items already allocated to other orders.

CAMO computes the optimum quantity of raw material required at each stage prior to rolling by 'exploding' to the lower levels on the process route structure (see Fgures 7.6, 7.7 and 7.8 (a-d). Similar to MRP, There are several fctors that must be considered in the CAMO process route and material explosion.

1. *Quantities of orders and inventory* - Each order has demand for several hundred pieces of the given dimension. When SCAPP provides the common gauge at each rolling stage it calculates a common gauge based only on one feasible item (if the dimensions of an item fall outside the manufacturing constraints two or more items are considered and a factor by which the quantity must be divided by is calculated). CAMO would take the net quantity at the top level structure and by exploding down the product route, it calculates the gross requirements at each level. It would then check the quantities in stock against the gross requirements and subtracted this to determine net requirements for meeting the master schedule.

2. *Dimensional commonality* - Similar to common items in MRP (BOM), in this environment, the common items would be the orders that have common gauges at any stage of rolling. Therefore, CAMO would collate there common orders and separate them where necessary. The most important function of CAMO would be to collect orders with similar gauage and width or length, calculate the net raw material items to be used and store this information for CSP module so that optimisation and cutting patterns generated by CSP are based on only one dimensional analysis.

3. *Lead times and time-phased scheduling* - All parameters regarding the manufacturing lead times would be stored in the machine capacity and parameter file as discussed in Chapter 5. From this file SCAPP would be able to determine setup times and operation time and hence the total manufacturing lead time can be determined. Purchased items lead times are already given by the stock master record described in the previous chapter. The order lead times are stored in the customer order file inputed to CAMO by the MPS. Therefore, the CAMO processor must determine when to start the operations by offsetting the due dates for these items and their respective manufacturing lead times.

Like MRP, a feature of CAMO should be the master production schedule's time-phased delivery requirements for end products, this time phasing must be carried through the calculations of the individual rolling stages and raw material requirements.

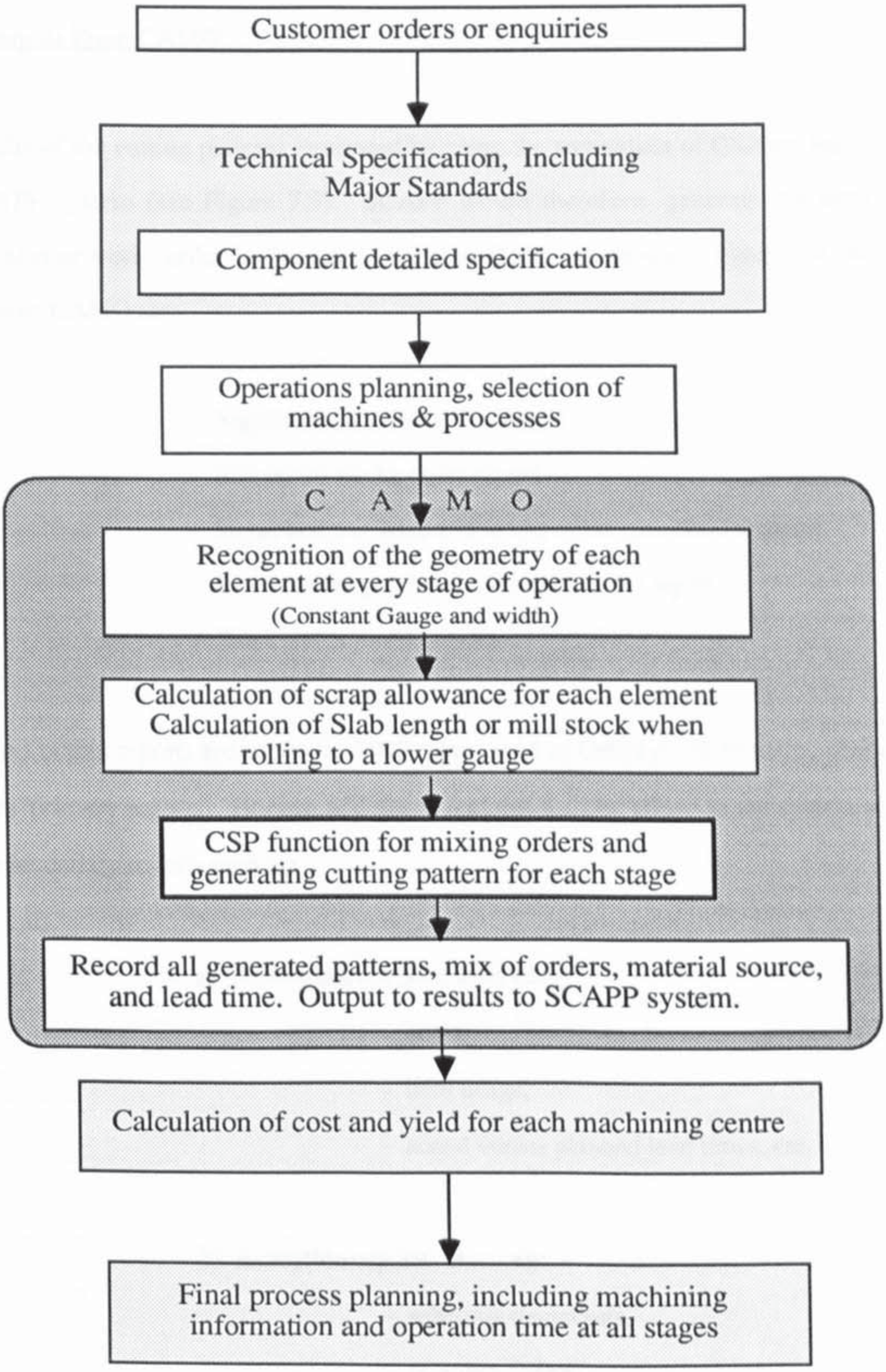


Figure 7.9 The information flow chart from customer orders to final process plans.

7.6.1 Outputs From CAMO

The results of the cutting patterns generated by using the techniques of CSP are fed to the SCAPP system (see Figure 7.9). SCAPP would therefore, generate the main process plan or works order for launch on the shop floor. As shown in Figure 7.4, the ouputs from CAMO are:

- Suggested purchase order report
- Suggested works order report
- Suggested purchase and works order cancellation report
- Rescheduled purchase and works order report
- Stock status profile report

The above output reports are similar to MRP reports and as Orlicky (1975) points about these are 'primary reports'. Hoever, additional features may be added to the system to provide secondary reports such as:

a) Performance report, indicating:
   - costs,
   - item usage,
   - actual versus planned lead times, etc.

b) Exception reports, showing:
   - schedule deviations,
   - overdue orders,
   - scrap rate and material yield.

c) Inventory forecasts, indicating projected inventory levels (both aggregate inventory as well as item inventory) in future periods.

7.7 Summary

In this chapter a comparison between the working of MRP and the computerised material optimisation CAMO in the rolling mill environment was presented. In doing that, the concept of distributed MRP and distributed bill of material presented by Love & Barekat (1988) was used.

The process structure produced by SCAPP system Husseini (1989), was used to convert the process routes where rolling takes place (see Figure 7.6), to a structure similar to a parts list or bill of material (see Figure 7.7). The structure was then broken down as shown in Figure 7.8a-d, and sub-structures produced. This concept is again similar to distributed bill of materials.

At each sub-structure where one stage rolling takes place, the length of the material expected after rolling is calculated. Having calculated the length, with the common gauge and width being constant and specified by the process route parameters, the CSP function is used to determine an optimum cutting pattern at that stage. Three major factors are considered at each sub-structure, order quantities and inventory, dimensional commonality and order mix and lead times and time-phased scheduling methods. This process is repeated for each sub-structure until the finished product is achieved.

Furthermore, the information regarding the cutting sequence at each stage of rolling is passed to SCAPP for the generation of the final process routs. The CSP objective philosophy is based on using the stock that has the least standard cost. The solutions of this therefore, produce the cutting pattern with minimum material cost. Hence, together with the operation costs, SCAPP would be capable of determining the cost not only for each stage of operation but for the entire process plan.

The yield is also calculated at each stage, providing the process planner with a better understanding of where the highest scrap rate lies. This on occasions, has been observed to be at the hot rolling stage where the reduction ratios are the greatest.

Finally, it can be concluded that the introduction of a computer aided material optimisation philosophy has enabled the rolling mill industry to achieve the benefits of what many companies claim to achieve from MRP in other industries.

CHAPTER 8

DEVELOPMENT OF TECHNIQUES OF ONE-DIMENSIONAL CUTTING STOCK PROBLEM

8.1 Introduction

The concept of this chapter is to outline the methods available in the art of the cutting stock problem. Of all the studies, none were found to provide solutions to cater for the practical situations where the stocks being cut from a large number of sheets could home in to provide the optimum number of stock sheets required. Techniques based on the Gilmore and Gomory's methods (1961) for the one-dimensional and the two-dimensional cutting stock problem are developed. Gilmore and Gomory acknowledged that their methods will not, in general, give integer solutions and their recommendation of simple rounding to meet integer constraint proved to be highly unsatisfactory in the rolling-mill context. It was, therefore, necessary to develop algorithms to find practical integer solutions to this type of problem.

8.2 Literature Survey

The cutting stock problem (CSP) is the problem of filling as many orders as possible without generating any scrap loss. This is a deterministic problem since slabs can be spread and rolled within known limits at the rolling mill to obtain pieces of the desired shape.

When a material is produced in bulk, there are constraints on the dimensions of the units of production. Two related problems arise. Firstly, the *assortment* problem, Hinxman (1979) what should be the stock sizes, i.e. the dimensions of the production units. Secondly, the *trim loss* problem Hinxman (1979), given the stock sizes, how should the stock be cut into pieces of the dimensions required by the customers, i.e. the order sizes. Trim loss and assortment problems have been categorised by dimension. Hence, a one-dimensional problem is one in which only one dimension of the stock and order pieces is significant, whilst a two-dimensional problem is one in which the stock and order pieces are rectangular and the dimensions in the two orthogonal directions are significant in the determination of a solution.

Various papers, Gilmore and Gomory (1963), Gilmore and Gomory (1966) and Haessler and Vonderembse (1978)) have been published to solve the problem of one-dimensional CSP objects, such as length of some steel bars, rolls of paper, or coils into smaller parts, each part having a given length and value as to maximise the total value of parts cut. A number of authors, Gilmore and Gomory (1961), Haesseler (1974) and Dyckhoff (1981) have made successful attempts to solve these problems by various methods for cutting specific types of material. In general the methods may be divided into two groups, *heuristic* and *algorithmic*.

8.2.1 Heuristic Method

A heuristic method cannot be guaranteed to find the optimal solution and often will not. A heuristic method is judged to be acceptable in use if the solutions it produces are satisfactory for the specific problem defined. That is, the solutions are known, or believed to be within a tolerable range of deviance from the optimal solution.

Heuristic methods are usually adopted where it is not feasible to employ algorithmic methods. An algorithm may not be available, or the computational cost of using the best available algorithm may be prohibitive.

Consider a plumber who has been asked to install the plumbing of a number of large sky-scrapers. His plans call for a large number of various length pieces of tubes, but the local stock holder of tubes only sells standard 5 metre tube lengths. He must figure out how few of the tubes he can purchase and still be able to cut all the required lengths from them. This problem arises frequently in industrial situations such as sheet metals, glass and paper industry, the garment and timber industry, and has been referred to as "the cutting stock problem", "the bin packing problem", and the "loading problem".

In order to determine the minimum number of tubes required, the plumber might first consider all possible patterns of cutting one tube into required lengths. Then these patterns must be combined in an optimal way so that demands are satisfied. The difficulty lies in the very large number of cutting patterns that can be generated. For example, with a parent roll of paper of 5000 mm and demands for 40 different lengths ranging from 500 mm to 2000 mm, the number of cutting patterns can exceed 100 million Gilmore and Gomory (1963).

8.2.1.1 One-dimensional Heuristic Problem

Haessler (1974) and Haessler and Vonderembse (1978) described a typical problem in the paper industry and a heuristic procedure for obtaining satisfactory results.

However, due to the nature of the problem in the flat rolling industry, the one-dimensional cutting stock problem is considered prior to the two-dimensional cutting stock problem.

8.2.2 Algorithmic Method

An algorithmic method for a problem guarantees to find the optimal solution for that problem as specified. The algorithmic methods used for CSP fall mainly into the well known categories of *linear programming* , and *dynamic programming* . These concepts also arise in the heuristic methods, either because they are used to solve sub-problems arising in a heuristic method, or because the structure of a heuristic method is based on them.

8.2.3 One-Dimentional Cutting Stock Problem

The one dimentional CSP is best illustrated by an example as shown below:

A large slab is rolled to produce smaller master slabs for further rolling and coiling ordered by the mill's customers. A typical slab is capable of producing master slabs that are 2500 mm wide and up to 20 mm thick, the length is obviously dependent on the exact size of the slab and conditions of the rollers at the time when the slab was rolled.

However, the average length is around 10 metres long. Assuming the set of orders to be filled requires cut slabs that are 12 mm thick and 10 metres long the quantities and widths required are as below:

Quantity	Widths (mm)
1	1500
4	1250
2	1000
2	750

The problem is to combine the required lengths into cutting patterns and determine how many times each cutting pattern is to be used to produce the requirements listed above. Ignoring material lost in the cutting process, a 2500 mm slab could be cut in half to produce two 1250 mm slabs. If two master slabs are cut this way, all the requirements for 1250 mm slabs will be met. Similarly, for the example given, one master slab can be cut into 1000 mm and 1500 mm pieces and a second can be cut to obtain two 750 mm slabs and one 1000 mm slab. This completes the solution to this simple example.

The example illustrated above occurs in a variety of industrial situations. It was first discussed in the literature by Eiseman (1967) and Paull and Walter (1954) in the context of the roll trim problem in the paper industry. The problem above therefore, can be formulated as follows:

$$\text{Min } \Sigma_j X_j \qquad \text{.......... (1)}$$

Subject to: $$\Sigma_j A_{ij} X_j = R_i \text{ for all } i \qquad \text{.......... (2)}$$

$$X_j \geq 0, \text{ integer} \qquad \text{...........(3)}$$

Where

X_i is the number of master slabs to be cut according to pattern j.

R_i is the number of cut slabs of width, W_i, required to fill order i for i=1,2,....,m.

A_{ij} is the number of cut slabs of width w_i obtained by cutting one master slab according to pattern j.

In the conventional formulation of the one-dimensional cutting stock problem, if the master slabs are of width w, a possible cutting pattern is any set of A_{ij} for i=1,...m that satisfy the following conditions:

$$\Sigma_i A_{ij} W_i <= W \qquad \text{..............(4)}$$

$$A_{ij} >= 0, \text{ integer for all } i \qquad \text{.............(5)}$$

The definition of a cutting pattern given in (4) ignores for the moment some considerations that are unique to the master slab problem in the steel industry. The necessary adjustments are discussed later in this chapter.

There are two aspects of the formulation in (1) to (3) that require special attention. The first is that the number of cutting patterns can be very large. Pierce (1964) has shown that for roll trim problems of moderate size the number of cutting patterns can easily exceed 100,000. The second is the integer restriction on the number of master slabs that must be cut according to each pattern. It should be clear that a value of $X_j = 1.6$, for example has no meaning. Furthermore it is not sufficient to simply round to the nearest integer number because then the restrictions in (2) may be violated. The large number of integer variables make it impractical to find optimal solutions.

Solution Methods

There are two ways to solve the problem formulated in (1) to (3) to obtain good, although not necessarily optimal, solutions. The first is to sequentially generate patterns that meet some criterion for acceptability until all the requirements are satisfied. This approach has been discussed by Haessler (1974 and 1978) and is useful in that it yields integer answers and permits direct control over certain types of solution characteristics such as the number of patterns used. The disadvantage of this approach is that in some situations it may give a solution with substantially more trim loss than is necessary.

The second approach is to find a linear programming solution and then adjust the results to obtain integer answers that meet all the restrictions in the problem. The pioneering work by Gilmore and Gomory (1961 and 1963) made it possible to obtain linear programming solutions to large scale cutting stock problems without first generating all possible cutting patterns. In their delayed pattern generation approach, Gilmore and Gomory find the next pattern to enter the solution if one exists by solving an associated problem. The solution to the associated problem is the pattern that has the largest degree of infeasibility in the dual problem.

The advantage of solving the problem using a linear programming algorithm is that it is possible to find the absolute minimum possible trim loss. The disadvantage relative to the sequential approach is the solution variables will generally be noninteger and substantial effort may be required to find an integer solution that satisfies all the restrictions. The most effective solution procedures are those that use both the sequential and linear programming approches to solve the problem. This might work as follows:

1. Find an initial solution using a sequential pattern generation approach.

2. Using the sequentially generated solution as the initial basis, find a minimum trim loss solution using a Gilmore and Gomory type linear programming algorithm.

3. Round the linear programming solution to integer values by rounding down and then up in such a way that the quantity produced of any size does not exceed the quantity orderd.

4. Solve any residual problem, consisting of those sizes for which too few have been produced in the linear programming solution, using the sequential approach.

There are a number of factors that are important to the Cutting Stock Problem which call for a generation of solutions that have no trim loss in the conventional sense. There are a number of reasons why this is practical.

1. Required slab sizes that cannot be cut from master slabs can be produced from ingots or held until the next group of orders is ready to be combined.

2. It is possible to inventory cut slabs until an order comes in that the cut slab can be applied to. If the master slab width is 800 mm and there is an order requiring a 500 mm width slab, but none for a 300 mm slab at present, the 500 mm can be cut and processed and the remaining 300 mm slab can be held in inventory.

3. It is possible to vary the width of the master slabs that are produced in the mill. Hence increasing the width reducing the gauge and increasing the length. Slabs can also be rotated through by 90 degrees so that the length becomes the width and the width becomes the length.

Generally, when slabs are rotated and rolling commences only two dimensions change, the gauge and the length.

As such, the definition of a feasible cutting pattern must be modified from what is given in (4) above. Let W_k be the width of master slab size k and let M be the material lost each time a slab is cut along the length. If W_i is the coil width to be produced at the rolling mill and P_i and Q_i are the amount coil size W_i can be spread and squeezed, respectively, during the rolling process, the restrictions that a pattern must meet to have no trim loss with master slab size k are as follows:

$$LL <= \Sigma_i A_{ijk} W_i <= UL \qquad \text{........(6)}$$

$$A_{ijk} =>, 0, \text{integer} \qquad \text{........(7)}$$

where

$$LL = W_k + M - \Sigma_i A_{ijk} (Q_i + M) \qquad \text{.......(8)}$$

and

$$UL = W_k + M + \Sigma_i A_{ijk} (P_i + M) \qquad \text{........(9)}$$

The above analysis is best illustrated by use of an example; suppose a 2000mm slab is to be cut once to obtain two cut slabs. If the material lost in cutting, M, is 10 mm, the usable amount of slab is 1990 mm. If each of the two cut slabs can be spread 20mm or squeezed 30 mm, any two coil sizes that sum to a value in the interval from 1930 to 2030 mm can be made from this master slab size with no scrap loss beyond the unavoidable 10 mm lost in the cutting process. If two cuts are to be made to obtain 3 cut slabs, the sum of the three coil sizes would have to be in the interval from 1890 to 2040 mm .

Obviously, spreading or squeezing a coil as it is rolled has an impact on the length of the coil and therefore on the weight of the finished coil, A 800 mm coil rolled from 820mm cut slab will be 2.5% heavier than an 800 mm coil rolled from an 800 mm cut slab. These variations are considered to be within the allowable industry tolerances for filling an order and no adjustment in order quantity is made because of this difference.

A second factor to consider is the question of which orders can be cut from the same master slab. In addition to size and quantity, the customer also specifies:

- Due dates
- Detailed grade specifications regarding metallurgical contents.
- Surface quality required depending on whether the metal will be used in an exposed or unexposed application.

In general metallurgical specifications for the same family of grades can be put in a quality hierarchy. Orders for this family of grades can be cut from the same master slab, provided the master slab meets the grade specifications of the highest quality order included in the pattern. When master slabs are produced for this family of grades, an attempt is made to produce only master slabs that meet the highest quality specifications and can be used for exposed surfaces. To this point, however, it has not always been possible to attain this level of quality consistently. Therefore the solution procedure must be developed in such a way that orders with similar requirements for:

- due date
- grade specifications
- surface quality

be combined into patterns whenever possible. This is important because at any time there may be a limited number of the highest quality master slabs available. Any pattern that requires the highest quality master slab but which also contains orders with lower quality requirements in some sense wastes a scarce resource and makes it more difficult to fill the orders on time.

It should be kept in mind that the scheduler has ultimate control over this through the definition of the problem to be solved. If only orders for the highest quality grade, exposed surface, and earliest due date are considered, then all the patterns found will contain orders with similar requirements. The difficulty here is that there may be relatively few cutting patterns available that have zero trim loss, as this is defined in (6) to (9). It may be necessary to include orders with other due dates or quality specifications to find patterns which utilise the whole master slab.

The two extremes in terms of problem formulation as it relates to the differences in due dates and order quality can to summarised below.

1. Only permit orders with similar due date and quality restrictions to be combined by solving each sub-group independently of one another.

2. Permit all orders in the same great family to be combined by solving one big problem and attempting to limit the intermixing of orders through the pattern generation process.

A third option is to start out as in 1. above with a limited group of orders but permit all orders that are not satisfied in patterns that use the whole master slab to carry over to the next problem. By sequencing the probems based on some priority scheme that is used in the solution procedure, it is possible to allow intercombining of quality and date requirements on a controlled and as needed basis.

Golden (1975) deals with the problems of combinatorial optimisation where problems are indeed very difficult to solve.

Various approaches by a number of authors Gilmore and Gomory (1961), Johnson (1974), Held et.al. (1974) and Eilon and Christofides (1971) are considered for the rest of this section.

The different approaches studied will include column generation, zero-one programming, combinatorial heuristics, and subgradient optimisation.

Again a number of brief examples will help to demonstrate the ideas behind each theory.

1. Paper Trim Problem. Consider the problem of cutting an unlimited number of pieces of stock of various lengths (for example, rolls of paper or rolls of copper coils) so that at least n_i pieces of lengths l_i are furnished, $i=1,....., I$. The objective is to meet the demands, n_i, while minimising the total number of rolls that must be cut. Since cutting each roll involves some waste, this keeps total waste at a low level, Lasdon (1970)

2. Vehicle Loading Problem. Allocate n objects or items of given magnitude Q_i $(i=1, 2, ..., n)$ to boxes, each box having a capacity C, in such a way that the capacity constraints are not violated and the number of boxes required is a minimum, Eilon and Christophides (1971).

3. Table Formatting. Given a list $L = (a_1, a_2,......, a_n)$ of real numbers in (0, 1), place the element of L into a minimum number, No, of "bins" so that no bin contains numbers whose sum exceed 1. Let the bins be computer words of length k bits and suppose there are items of data requiring $k_{a1}, k_{a2}, ..., k_{an}$ bits respectively, Johnson (1974).

8.2.3.1 Column Generation

Gilmore and Gomory (1961 and 1963) presented a linear programming formulation for the cutting stock problem involving column generation and efficient solution of knapsack problems. Their later papers (1964 and 1966) develop a knapsack algorithm five times faster than the standard dynamic programming recursion applied in earlier work.

In the paper trim problem, we assume for simplicity, that all pieces available for cutting have length l and that max $l_i <= L$. A cutting pattern, represented by a vector (a_1, a_2, ...a_l), consists of a1 pieces of length l_1, a_2 pieces of length l_2, etc. The number of cutting patterns will, in general, be enormous. Let j index the various cutting patterns and define:

a_{ij} = number of pieces of length l_i cut by pattern j

x_j = number of times pattern j is used

n_i = demand for length l_i

The problem may be expressed as a large scale linear integer program.

minimise $\quad z = \sum c_j x_j$ ....(10)

subject to

$\sum a_{ij} x_j >= n_i$ ....(11)

$x_j >= 0$ ....(12)

x_j integer

where $a_{ij}x_j$ is the number of pieces of length l_i cut using pattern j, $c_j = 1$, and the objective function minimizes total rolls of paper used.

With high demands, the number of times different patterns are used will generally be large, and rounding up to integers, after solving the linear program (10) - (12) leads to very good solutions. The large number of variables (cutting patterns), however, makes direct solutions via the simplex method infeasible for all but the most trivial problems. Column generation b generates the coefficient data a_{ij} when needed, and can be thought of as an extension to decomposition. In decomposition, data for any variable corresponds to an extreme point or extreme ray of another linear program. New data are generated by solving this linear program subprogram with an appropriate objective function. In column generation the subproblem need not be a linear program; in the cutting stock problem the subproblem is a knapsack problem.

The relative cost coefficient for a nonbasic variable x_j is $c_j = 1 - \sum_i \Pi_i a_{ij}$ where Π_i are the simplex multipliers from (11). Since (10) - (12) is a minimization problem, we seek the most negative reduced cost, so the k th subproblem becomes:

$$\text{minimise} \qquad \nu^k = [1 - \textstyle\sum_i \Pi_i^k a_{ij}] \qquad \text{........(13)}$$

where the minimisation is over values of j such that $1 <= j <= n$.

A cutting pattern $a = (a_1, a_2, ...a_I)$ must satisfy

(i) $\sum_i l_i a_i <= L$

(ii) a_i = nonnegative integer.

The subproblem reduces to determining the coefficients aij of a new pattern which minimizes (13) or equivalently.

$$\max \sum_i \Pi_i a_i \qquad ....(14)$$

subject to

$$\sum_i l_i a_i <= L \qquad ....(15)$$

$$a_i = \text{nonnegative integer.} \qquad ....(16)$$

problems of this form are called knapsack problems. Suppose there are I objects, the i th object having weight l_i and value Π_i, and it is desired to find the most valuable subset of objects whose total weight does not exceed L. We can view L as being the capacity of a knapsack which we must fill optimally for a camping trip. This combinatorial problem can be viewed as a shortest path problem in a suitable acyclic network, Garfinkel and Nemhauser (1972).

If the optimal objective value in (14) is ν then:

(i) if ν >1, the corresponding pattern vector is formed and enters the basis.

(ii) if ν <= 1, The current solution is optimal.

The column generation approach to the cutting stock problem is most efficient when an optimal solution is obtained before too many columns have been added to the restricted master problem, and the subproblems can be easily solved. If there are several standard parent stock lengths L_i, with cost c_i, several problems like (14) - (16) must be solved.

8.2.3.2 Knapsack Method

Gilmore and Gomory (1961) outlined a method of solution involving dynamic programming. Let $F_{s+1}(x)$ be the value of the most valuable combination of objects that can be fitted into a knapsack of capacity x if only the first $s+1$ items are used. Then the fundamental dynamic programming recursion becomes:

$$F_{s+1}(x) = \max \{ r\Pi_{s+1} + F_s(x - rl_{s+1}) \},$$

$$(0 <= r <= [^x/l_{s+1}])$$

which expresses the principle of optimality. Once we decide how many pieces r of length l_{s+1} to use in the pattern, the remaining total length of $x - rl_{s+1}$ must be optimally divided amongst the s lengths $l_1, l_2, ... l_s$. Dynamic programming does not perform well in this setting since no use is made of the linearity of the objective function and constraints. Gilmore and Gomory (1963) demonstrate that a faster method for solution of the knapsack problem is needed if paper industry problems are to be solved effectively by column generation. A lexographic-based algorithm is one of the contributions of their important (1963) paper.

In addition to developing more efficient knapsack solution techniques, Gilmore and Gomory were successful in decreasing the size of the knapsack problems under consideration. They observed that simplex multipliers for different rows were often identical. This fact can be exploited since if, in (14) - (16), say $\Pi_1 = \Pi_2$ and $l_2 > l_1$, then there is an optimal solution in which a_2 is zero. This is because if a_2 were positive, its value could be assigned to a_1 without changing the objective value or violating constraints (15). Thus variable a_2 may be dropped and a smaller knapsack problem results. In the problems tested identical simplex pieces decreased the average size of the knapsack problems from 30 to 18.2 variables Gilmore and Gomory (1966).

Another ingenious device devised by Gilmore and Gomory is their "median method". By (13) the column with minimum reduced cost c_j is brought into the basis. This assures us maximum decrease of z. Total decrease in z is determined by $c_j x_j$. In general it is computationally prohibitive to search for min $c_j x_j$. In this case, however, one can make progress in this direction. Consider a cutting pattern j containing both high and low demand lengths. The number of times this pattern may be used, x_j, is limited by the lowest demand length. If this were not so, cutting pattern j alone might produce more than demanded of this length. The following proposed remedy has yielded significant improvements on required number of pivots. Simply order lengths by demand and separate the lengths into high and low demand groups about the median demand. Every other pivot, restrict the knapsack problem variables to a high demand group (that is, set variables a_i for the low demand group to zero). This permitted larger x_j values and changes in z.

8.2.3.3 Computational Experiments with Column Generation

Gilmore and Gomory blend theoretical and applied work in their paper (1964 and 1966); the computational experiments are plentiful, meaningful. Golden (1975) reports on five series of test problems for which demands were high, as are the resulting number of times the patterns are used. Thus dropping integer restriction (12) has little effect. A cutting knife limit restricting the total number of cuts per pattern, $\sum_i a_i$, was employed.

This real world limitation is easily incorporated into the knapsack framework, and will be discussed later. When the improved knapsack method, identical simplex prices, and the median method are taken advantage of the test problems were solved by Golden (1975) on an IBM 7094 in about two minutes apiece. The percentage of wasted material ranged from near zero to almost 10%.

Because of the enormous number of columns which implicitly must be considered, column generation calculations tend to slow down in the sense that slightly better combinations of patterns are obtained over many pivots which yield only marginal improvement. This is a general deficiency of the column generation approach. A cut-off criterion was utilized to decide when pivots were more costly than rewarding. If ten pivot steps did not produce 0.1% reduction in waste calculations would stop. Golden reports of remarkable savings in running time. The first series of five problems took a total of 2.28 minutes instead of 13.95 minutes with an average increase in waste of only 0.26% [Gilmore and Gomory (1964) and (1966)].

8.2.3.4 Formulation Extension

Gilmore and Gomory suggest several formulation changes which more closely approximate real world paper industry constraints. They are:

(i) Cutting-knife limitations
(ii) Machine balance problems
(iii) Customer tolerances

Often the number of pieces cut in a pattern is limited by the number of R of cutting knives available. This means the constraint $\sum_{i=1}^{I} a_i <= R$ must be taken into account in the knapsack problem. This modification is minor. When a vector a is determined in the improved knapsack algorithm we must test whether the new constraint is violated. So after having calculated a_r we proceed as follows:

(i) if $\sum_{i=1}^{r} a_i < R$ we calculate a_{r+1} and continue;

(ii) if $\sum_{i=1}^{r} a_i >= R$ we reduce a_r by $\sum_{i=1}^{r} a_i - R$ and continue.

In the machine balance problem there are several cutting machines of different lengths, and one may want to balance production so that no machine is idle for a very long period, or production from each machine may be limited. If there are P machines, with the r th machine capable of producing Q_r rolls of its length L_r, the linear programming problem becomes:

$$\min \sum c_r x_{jr} \qquad \text{.........(17)}$$

subject to

$$\sum a_{ijr} x_{jr} >= n_i \qquad \text{.........(18)}$$

$$\sum x_{jr} <= Q_r \qquad \text{.........(19)}$$

with $x_{jr} >=0$, where c_r = cost or length of a roll from the r th machine

x_{jr} = number of times j th pattern is used on the r th machine

a_{ijr} = number of times pieces of length l_i is produced from j th pattern on machine r.

If we assign simplex prices Π_i to the first I constants in (18) and Π_r to the next P constraints in (19) the subproblem becomes:

$$\min \quad c_r = c_r - \sum \Pi_i a_i + \Pi_r \qquad \text{........(20)}$$

subject to

$$\sum l_i a_i <= L_r \qquad \text{........(21)}$$

The desired column is produced by solving P different knapsack problems

$$\max \sum \Pi_i a_i$$

subject to

$$\sum l_i a_i <= L_r$$

with a_i a nonnegative integer, and choosing the one which solves (20). Once the column is produced, the calculations proceed as in the simple stock cutting case. Notice that if we divide inequalities (19) by Q_r and redefine variables we produce generalised upper rounding constraints, Lasdon (1970). This may be exploited in the restricted master problem.

Another special feature of this type of industry, Gilmore and Gomory (1961) is that customers will often accept a range of order quantities. For instance, instead of requiring n_i pieces of length l_i, any quantity between n_i' and n_i'' is allowed, where $n_i'' > n_i'$. In this case the number of rolls is no longer a suitable objective function, since this would favor demands of n_i'. Minimum percentage waste, which is perhaps a better measure of efficiency, may equire greater production. Percentage waste is introduced as the objective function, resulting in a fractional objective function. The waste incurred in cutting pattern j is $w_j = L - \sum_i l_i a_{ij}$. If patterrn j is used x_j times, then the total percentage waste is $100 \{ \sum_j w_j x_j / L \sum_j x_j \}$. The new problem may be cast in the form:

$$\min \quad \sum_j w_j x_j / \sum_j x_j \qquad ....(22)$$

subject to

$$\sum_j a_{ij} x_j - s_i >= n_i' \qquad ....(23)$$

$$\sum_j a_{ij} x_j <= n_i'' \qquad ....(24)$$

$$x_j >= 0. \qquad ....(25)$$

Including slack variables in (23) and subtracting (23) from (24) we obtain constraints:

$$\sum_j a_{ij} x_j - s_i = n_i'$$

$$0 <= s_i <= (n_i' - n_i'')$$

$$x_j >= 0$$

The upper bounding method can be used to avoid increasing the basis size; the rational objective function can be handled by looking for the variable with the most negative directional derivative. Again the subproblem reduces to a knapsack problem.

8.2.3.5 Zero-One Programming Approach

The logistic problems of vehicle routing, vehicle scheduling, and vehicle loading are very much interrelated. Vehicle routing problems involve finding a set of feasible routes which minimize the total distance travelled in supplying customers. These routes must then be scheduled, taking into account various timing restrictions. If the example of the vehicle loading problem, Eilon and Christofides (1971) and Eilon et al. (1971) is considered where a number of consignments (for examples, bundles of newspapers) must be loaded in vehicles of known size. The problem is to minimise the number of vehicles required. Other examples of loading problems are presented in *Distribution Management* Eilon et al. (1971). Eilon and Christofides (1971) examine two methods of solution to the loading problem and compare their efficiency. The first method is a zero-one programming algorithm, the second a heuristic algorithm. The assignment formulation for the loading problem and comments on their computational experience is disscussed below.

$$\text{set} \quad x_{ij} = \begin{cases} 1 \text{ if item } i \text{ is allocated to box } j \\ 0 \text{ if otherwise} \end{cases}$$

l_i = length or weight of item i

L = capacity for length of box

d_j = penalty assigned to the allocation of items to box j and this penalty increases with j so that $d_{j+1} >> d_j > 0$.

One may take $d_{j+1} > pd_j$ and $d_1 = 1$ where p is the largest number of items that can be loaded into a box. The vehicle loading problem becomes:

$$\min \sum_j \sum_i d_j x_{ij} \qquad \text{........(26)}$$

subject to

$$\sum_i l_i x_{ij} <= L \qquad j=1, ..., N \qquad \text{........(27)}$$

$$\sum_j x_{ij} = 1 \qquad i=1, ..., n \qquad \text{........(28)}$$

$$x_{ij}\ U\ \{0,1\}. \qquad \text{........(29)}$$

Equations (27) state that total capacity taken up by all items i in box j must not exceed capacity. Equations (28) insure that each item is allocated to one box only. Minimisation of the penalty function (26) will yield the smallest number of boxes, since it is always better to fill up occupied boxes before starting new ones.

The heuristic algorithm is similar to the best fit decreasing algorithm, disscussed later in the chapter. Fifty one-dimensional problems with up to 50 items were solved, Golden (1975) using each method on the IBM 7094. Typical problems solved by Gilmore and Gomory involved on the order of 25,000 items, Golden (1975). Computing time for the zero-one code ranged from one minute to over 13 minutes in one case reports Golden. He goes on to say, the heuristic ran in 0.8 minutes in the worst case. When Eilon and Christofides employ a reshuffle routine to make obvious improvements on the heuristic solutions, running times increase to at most 1.2 minutes. The heuristic produced the optimal solution in 90% of the cases, and the heuristic with reshuffle routine produced the optimal solution for all but two of the remaining 10%, Golden (1975).

These results emphasize the merit of the Gilmore and Gomory approach which, although not an optimal approach, can handle problems of such enormous size and obtain such good solutions. This paper (1966), however, presents an interesting alternative formulation for the cutting stock problem. The computational results and reshuffle routine bear directly on bin packing results which are discussed below.

8.2.3.6 Combinatorial Bin Packing Heuristics

A classic example of bin packing problem exists in computer science literature known as file allocation problem, where it is desired to place files of varying sizes on as few tracks of a disk as possible, and files may not be broken between tracks.

Given objects o_i with weights $0 < a_i <= w$, $1 <= i <= n$ and an unlimited supply of bins B_j, each with a maximum capacity of w units, assign all the objects to the minimum number N_o of bins, subject to the constraint that objects in a bin cannot have total weight exceeding w. It should be clear that this is a cutting stock problem where the weights a_i correspond to lengths. Objects with identical weights can be grouped together into group k, and demand n_k is the number of objects in group k.

Algorithms for bin packing have received much attention in recent years. Garey, Graham and Ullman (1972) presented worst case evaluations of various heuristic algorithms. In 1973 these authors, Garey, Graham and Ullman (1973) outlined their previous results, indicating that sharper results were on the way. Meanwhile, Johnson, (1974) resolved several important conjectures. Johnson, Demers, Ullman, Garey and Graham have in 1974 put together a comprehensive paper incorporating both new and old results. The results in this chapter come from this paper.

In that paper, combinatorial heuristics are presented which compute good solutions to the bin packing problem in an acceptable amount of computer time. The motivation is the expected difficulty required by any optimal algorithm. Heuristic algorithms are evaluated on the basis of how closely the constructed solutions approximate optimal solution in the worst case. The algorithm discussed are all considered "sufficiently fast." Running times are not studied. Rather the aim is to seek the best heuristic from this class of algorithms, with respect to the objective value considerations. Worst case rather than average behaviour is studied for the following reasons:

(i) a probability distribution for the problem is difficult to determine;

(ii) worst case results are easier;

(iii) worst case results guarantee that a particular algorithm will never exceed the optimal solution by more than a known percentage;

(iv) the case of worst case behaviour generally has a similar effect on average behaviour.

8.2.4 Algorithms and computational results

The following four placement algorithms are considered in Johnson et al. (1974).

Algorithm 1 (first -fit). For list $L = (a_1, \ldots\ldots, a_n)$ the weights a_k are successively assigned in order of increasing k, each to the bin B_j of lowest index into which it can validly be placed. The number of bins thus required will be denoted N_{FF}.

Algorithm 2 (Best-fit). Assign the next a_k in a list L to the bin whose resulting unused capacity is minimal. N_{BF} will be used to denote the number of bins required in this case.

Algorithm 3 (First-fit decreasing). L is rearranged so that $a_1 => \ldots => a_n$. We then apply Algorithm 1 to the derived list. The number of bins required will be denoted by N_{FFD}.

Algorithm 4 (Best-fit decraesing). L is arranged so that $a_1 => \ldots => a_n$. We then apply Algorithm 2 to the derived list and obtain N_{BFD} bins.

$$\text{Let } R_{FF}(K) = \max \{(N_{FF}/N_o)/N_o = K\} \text{ and define } R_{BF}(K),$$

$R_{FFD}(K)$, and $R_{BFD}(K)$ similarly. We are interested in the ratio of the number of bins required by an algorithm executed on list L to the minimum number of bins N_o. $R_{FF}(K)$, $R_{BF}(K)$, and $R_{BFD}(K)$ are the maximum values achieved by these ratios over all lists, with $N_o = K$. The main results are discussed by Johnson (1974) the first two of which were presented in the conference paper by Garey et al. (1972), and are summarised below:

$$\lim_{K \to \infty} RFF(K) = 17/10$$

$$\lim_{K \to \infty} RBF(K) = 17/10$$

$$\lim_{K \to \infty} RFFD(K) = 11/9$$

$$\lim_{K \to \infty} RBFD(K) = 11/9$$

These ratios reflect performance for essentially all values of K as indicated in theorems 2.1, 2.2, 3.1, 3.2 Johnson et al. (1974). These results are significant in that one can be certain that their solutions will differ from the optimal solutions by no more than about 22% for Algorithms 3 and 4. However, empirical tests of average case behaviour would still be valuable and would complement the worst case results.

Johnson (1974) has performed some preliminary computational experiments on the MIT Artificial Intelligence Laboratory's PDP-10 Time-Sharing System. For Algorithm 1, 2, 3 and 4, 25 different lists of 200 weights uniformly distributed in the intervals (0, 1), (0, 1/2), (0, 1/4) were generated and average percentage of excess bins was calculated, based on best lower bounds for N_o. In Table 1, the outcomes of some of these experiments are exhibited.

Table 8.1 Average percentage of excess bins required.

	Uniform (0,1.0)	Uniform (0,.50)	Uniform (0,.25)
FF	0.70	2.20	0.60
BF	5.60	2.0	0.50
FFD	1.90	0.10	0.20
BFD	1.90	0.10	0.20

The principal conclusion to be drawn here from the above table is that apparently these simple combinatorial bin packing algorithms perform much better than worst-case bounds might indicate. For example, in using Algorithms 3 and 4 one might expect no more than a 2% excess number of bins. It would be interesting to run these algorithms on the data provided by Gilmore and Gomory (1961, 1963, 1964 and 1966) and compare solutions. In the typical test problem mentioned in Gilmore and Gomory's paper of 1961, the largest weight was less than half the capacity. From Table 1, if weights are uniformly distributed in (0, 0.5) one can expect just about optimal results.

Although comparative running times for these heuristic algorithms have not been tested, Algorithm 3 looks the most attractive from the viewpoint of both speed and accuracy. The best-fit decreasing algorithm is rather slow; at each step all bins must be examined before a placement decision can be made. On the other hand, the first-fit decreasing algorithm simply chooses the first bin encountered which can accommodate the placement. Algorithm 3 is very fast and easily programmed (it has been programmed in Fortran and has solved random 50-items problems in less than 0.3 seconds of execution time on an IBM 360/50) Johnson et al. (1974).

8.2.5 Subgradient Optimisation

In this section, a rather recent technique involving Lagrangean relaxation called subgradient optimisation is discussed. This technique has been successfully applied to some large scale optimisation problems. An emphasis on the difficulties in the column generation approach of Gilmore and Gomory will help to demonstrate the subgradient optimisation. Firstly, the linear program being solved is only an approximation to the integer program whose solution is really desired. Secondly, programs with very many columns have long "tails" and an excessive number of iterations may be required.

Subgadient optimisation has been the focus of much attention since large travelling salesmen problems were solved with astounding success by Held and Karp (1970 and 1971) using this approch. The travelling salesmen problem and the cutting stock problem are both members of the polynomial complete problem class. Held, Wolfe, and Crowder (1974) demonstrate that subgradient optimisation has worked on several combinatorial problems already.

Consider the following general mathematical programming problem.

Let $v_k = (v_{k1}, v_{k2}, ..., v_{kn})$, $k = 1, 2, ..., m$, be a finite set of real *n-vectors* and c_k be a scalar quantity associated with v_k. Find a real vector § =($\pi_1, \pi_2 ..... \pi_n$) which solves:

$$\max_{\Pi \cup Q} \; w(\Pi) \qquad ....(30)$$

such that $$w(\Pi) = \min_k [c_k + \S \, . \, v_k] \qquad ....(31)$$

where Q is a convex set of constraints. Π may be unconstrained (Held et al., 1972). All linear programs for which Dantzig-Wolfe decomposition can be applied can be put into form (30) and (31). Consider, for example, the following problem:

$$\min \; CX \qquad ....(32)$$

subject to

$$AX = b \qquad ....(33)$$

$$A'X = b' \qquad ....(34)$$

$$X >= 0 \qquad ....(35)$$

Assuming that:

$$\{X / A'X = b', X >= 0\}$$

is bounded with extreme points X^k we may write (32)-(35) in the form:

$$\min \sum (CX^k)\, \Omega_k$$

subject to

$$\sum (CX^k)\, \Omega_k = b$$

$$\sum \Omega_k = 1$$

$$\Omega_k >= 0$$

The dual to this linear program is given by:

$$\max \ \prod b + \partial \qquad \text{..........(36)}$$

subject to

$$\prod (AX^k) + \partial <= (CX^k) \qquad \text{..........(37)}$$

$$\prod, \partial \text{ unrestricted} \qquad \text{..........(38)}$$

$$\partial <= CX^k - \prod (AX^k) == \prod b + \partial <= \prod b + CX^k - \prod (AX^k).$$

Thus (236)-(38) becomes: $\max\limits_{w, \prod} [w|w <= CX^k - \prod(AX^k - b)]$

which is equivalent to $\max w(\prod)$

$$w(\prod) = \min [\, CX^k - \prod(AX^k - b)]$$

which is in form (30), (31).

If m is relatively small and Q is a polyhedron then (30), (31) can be solved by the simplex method. However, when m is very large the problem can be viewed as one of maximising a concave function over Q and subgradient optimisation may be helpful.

8.2.6 The Subgradient Method

For simplicity let $Q = E^n$, the entire n -space. Problem (30), (31) becomes:

$$\max w(\Pi) \qquad \text{......... (39)}$$

$$w(\Pi) = \min \{ c_k + \Pi . v_k \mid k = 1, ..., m \}. \qquad \text{..........(40)}$$

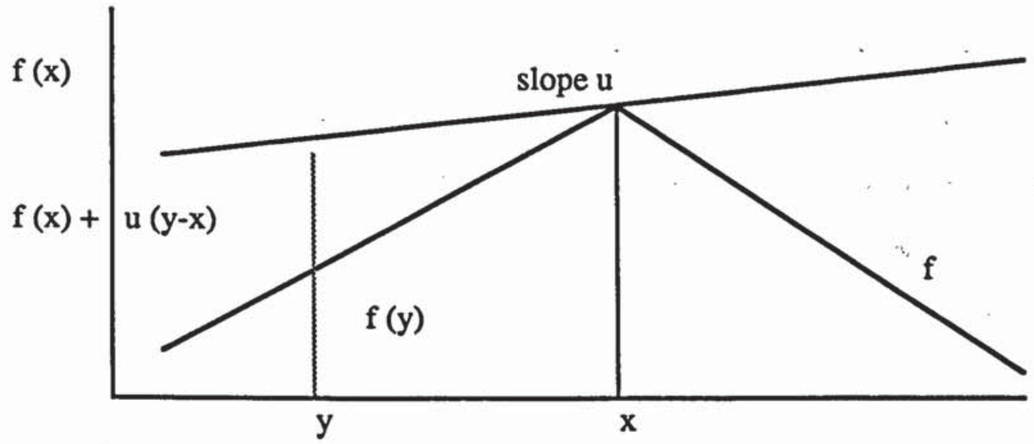
Let us suppose that w is bounded from above. Since w is piecewise linear there exists a point Π^* such that $w(\Pi^*) = \max\ w = w^*$. For any Π, the minimum in (40) is attained for at least one value of k.

Define $V(\Pi) = \{vk \mid c_k + \Pi . v_k = w(\Pi), k\ U \{1, 2, ..., m\}\}$. $V(\Pi)$ will usually contain one element; the function w is differentiable at Π in this case, and vk U$V(\Pi)$ is the gradient $\Delta w(\Pi)$. if $V(\Pi)$ is not a singleton then w is not differentiable at Π. A subgradient extends the notation of gradient here. The n -vector u is a subgradient at x of the concave function f if

$$f(y) - f(x) <= u(y-x) \text{ for all } y. \qquad \text{..........(41)}$$

Figure 8.1 illustrates this concept.

Figure 8.1: Definition of Subgradient.



u is simply the slope of a supporting hyperplane to the graph of f at x. If f is differentiable at x, the only such plane is the tangent plane given by $u = \Delta f(x)$.

A possible strategy might be to determine all v_k U$V(\Pi)$ and define the locally "best" direction vector v_k^* as a direction of steepest ascent. This however, is not done; rather than searching for the entire set $V(\Pi)$, subgradient optimisation solves (40) for a v_k U$V(\Pi)$ and proceeds. In addition, no attempt is made to maximise w in the choosen direction. Thus, this approach differs markedly from the classical feasible directions approach. In fact, subgradient optimisation approaches in euclidean distance a maximum point of the function w.

8.2.7 Applications to Cutting Stock Problem

Subgradient optimisation is an approach for approximating the maximum of certain piecewise linear concave functions, which has been effective in handling some difficult large scale combinatorial problems. It has already been applied to assignment problems, noninteger symmetric travelling salesman problems, and multicommodity maximum flow problems Held et al. (1974), with successful results. This approach seems especially well suited for large scale integer programs where sharp approximations (lower bounds) to the optimal objective values may be incorporated in a branch and bound scheme to produce miniscule search trees, as in the Held and Karp solution of the travelling salesman problem.

Below two applications of subgradient optimisation to the cutting stock problem, corresponding to the two formulations discussed previously (10)-(12), and (26)-(29) are presented.

Express the cutting stock problem (10) - (12) in the following form, where one may add a redundant equation limiting the total number of patterns:

$$\min \textstyle\sum_j c_j x_j \qquad \text{.........(42)}$$

subject to

$$\textstyle\sum_j a_{ij} x_j >= n_i \qquad \text{.........(43)}$$

$$\textstyle\sum_j x_j >= \sum_i n_i \qquad \text{.........(44)}$$

$$x_j >= 0. \qquad \text{.........(45)}$$

Let $\{X \mid \sum_j x_j <= \sum_i n_i , X >= 0 \}$ have extreme points X^k and the above program becomes:

$$\min \sum (CX^k)\, \Omega_k$$

subject to

$$\sum (AX^k)\, \Omega_k >= n$$

$$\sum \Omega_k = 1$$

$$\Omega_k = 0$$

The dual problem

$$\max \ \prod n + \partial$$

subject to

$$\prod(AX^k) + \partial <= CX^k$$

$$\prod >= 0$$

is equivalent to

$$\max \ w(\prod) \qquad ....(46)$$

$$w(\prod) = \min \{(C - \prod A)X^k + \prod n \}. \qquad ....(47)$$

solving (46), (47) reduces to solving the Lagrangean relaxation of (42) - (45) relative to equations (43) Geoffrion (1974). The $w(\prod)$ problem

$$\min \ \{ \sum_j (c_j - \sum_i \pi_i a_{ij}) x_j + \sum_i \pi_i n_i$$

subject to

$$\sum_j x_j <= \sum_i n_i$$

is solved by finding the most negative reduced cost $c_j' = c_j - \sum_i \pi_i a_{ij}$, and setting $x_j = \sum_i n_i$.

In order to determine the most negative $c_j'(c_j = 1)$ we solve the knapsack problem(14) - (16). $(v_k)_i$ is determined by $(v_k)_i = n_i - \sum_j a_{ij}x_j$ and can be thought of as the number of pieces demanded of length i that still must be cut. Notice that the restricted master problem has been eliminated by this approach.

Alternatively, begin with zero-one programming formulation (26) - (29) but replace (29) with $x_{ij} >= 0$. The dual to this linear program is written as:

$$\max \ -L \sum_i \pi_i a_{ij} + \sum_i \partial_i$$

subject to

$$- l_i \pi_j + \partial_i <= d_j$$

$$\pi_j >= 0.$$

$\partial_i <= d_j + l_i \pi_j == \partial_i = \min \{ d_j + l_i \pi_j \}$. Equivalently, the dual is:

$$\max \ w(\Pi)$$

$$w(\Pi) = -L \sum_j \pi_j + \sum_i \min_j [d_j + l_i \pi_j]$$

which is a piecewise linear concave function. To solve $w(\Pi)$ one must find the minimum of N numbers $d_j + l_i \pi_j (j = 1, \ldots\ldots, N)$, for all values of i. $(v_k)_i$ is the coefficient of π_i in $w(\Pi)$.

Subgradient optimisation may be imbeded in a branch and bound framework for both suggested approches in order to overcome nonintegral solutions. Geoffrion (1974) and Fisher et al. (1974) discuss this topic in detail.

Haessler (1974), considered and tried a number of mathematical programming methods but rejected them for two main reasons: (1) Solutions based on linear programming methods for minimising trim loss tend to comprise many patterns which are used for relatively short run lengths. Where tolerances are included in the formulation, the tendancy is for minimum demand quantities to be scheduled in order to reduce trim loss. (2) The large number of feasible patterns leads to prohibitively long search times if branch and bound methods are used.

Having found the more general heuristic methods proposed by Pierce (1964) to be unsatisfactory, Haessler developed a heuristic procedure using the same basic pattern enumeration technique and obtained solutions to a set of twenty scheduling problems for which manually obtained solutions were available.

Haessler (1974) argues that it is not an appropriate criterion to accept 'optimal' solutions considering only the minimisation of trim loss. However, in most real-world cutting stock problems, there are other factors that must be considered. For example:

1. The solutions must be integer-valued. A discussion of this requirement is given in Woolsey (1972).

2. Existing manual solution procedures usually generate solutions in which the number of active patterns is far less than the number of sizes. Barring degeneracy, linear programming theory specifies that the number of active variables will equal the number of sizes required. In many cases there is a setup

cost or fixed charge associated with changing cutting patterns so that in reality the one-dimensional cutting stock problem is not a linear programming problem, but an integer programming problem with fixed charges associated with pattern changes.

As a result of this, Haessler (1974) revised the (10) - (12) formulation of the cutting stock problem to:

$$\min\ C_1\Sigma_j T_j X_j + C_2\Sigma_j \partial(X_j),$$

subject to

$$R_1 <= \Sigma_j A_j X_j <= R_u,$$

$$X_j >= 0, \text{ integer valued,}$$

where

C_1 is the value of trim loss per unit length,

C_2 is the cost of changing patterns,

R_1 and R_u are lower and upper bounds on the customer order requirements, reflecting general industry practice to allow limited overruns or underruns,

and

$(X_j) = 1$ for $X_j > 0$ and 0 otherwise.

Haessler aknowledges that the mathematical programming problem formulated above is clearly far beyond the capability of existing integer programming codes. Pierce (1964) presents a method presents a method for generating all undominated cutting patterns and demonstrates that the number can be very large. He gives an example with seven sizes that has over 800 cutting patterns and one with 50 sizes that has over 900,000 cutting patterns.

The only realistic way to approach a problem such as this is to develop a heuristic procedure that will generate good, but not necessarily optimal, solutions in a reasonable amount of time. These problems are currently being solved manually, and in any given situation it is possible to measure the quality of the manual solutions and the associated cost of developing them. To be economically justifiable, Haessler argues that a computerised heuristic procedure must be able to generate good solutions at a reasonable cost, such that the sum of the controllable production costs plus the cost of obtaining the solution are less for the heuristic program than for the manual procedure.

The development of a heuristic solution procedure depends upon identifying the important solution charactaristics and then constructing search procedures and decision rules that will generate solutions with these desired characteristics. Clearly, the key is to find cutting patterns with low trim loss and high usage levels. Patterns with low trim loss will control the trim loss as desired, while patterns with high usage levels will tend to reduce the total number of different patterns in the solution. In addition, there is need to control the number of rolls in the pattern to prevent an imbalance of either wide or narrow rolls from occurring at some future stage in the solution process.

8.3 Summary

The cutting stock problem is a large scale combinatorial problem for which several solution techniques exist in the literature; each with its drawbacks and/or approximations. In this chapter some of the more popular techniques were surveyed. An in-depth study was carried out based on subgradient optimisation. Subgradient optimisation may also be a useful tool for evaluating combinatorial heuristic algorithms. Johnson et al (1974). Computational efficiency, objective value accuracy, and code availability with respect to all the algorithms considered must be examined before any definitive conclusions are reached.

The best algorithmic methods for cutting stock problems to date are the linear programming procedures of Gilmore and Gomory adapted to use subsequently developed better solution methods for the knapsack problem. However these are computationally expensive. Stainton observes that from a practical standpoint the cost of proceeding in the way they described would be prohibitive (the cost of the computing would be greater than any savings that might be achieved in steel utilisation).

The state of the art is such that anyone faced with a cutting stock problem is likely to have to develop his own method, unless the Gilmore and Gomory approach is feasible. Such methods are almost bound to be heuristic. However, note should be taken that the amount of literature dealing with the cutting stock problem is very limited due to the fact that many models, procedures and programming have been developed commercially and hence kept confidential.

8.3 Summary

The cutting stock problem is a large scale combinatorial problem for which several solution techniques exist in the literature; each with its drawbacks and/or approximations. In this chapter some of the more popular techniques were surveyed. An in-depth study was carried out based on subgradient optimisation. Subgradient optimisation may also be a useful tool for evaluating combinatorial heuristic algorithems. Johnson et al (1974). Computational effeciency, objective value accuracy, and code availability with respect to all the algorithems considered must be examined before any definitive conclusions are reached.

The best algorithmic methods for cutting stock problems to date are the linear programming procedures of Gilmore and Gomory adapted to use subsequently developed better solution methods for the knapsack problem. However these are computationally expensive. Stainton observes that from a practival standpoint the cost of proceeding in the way they described would be prohibitive (the cost of the computing would be greater than any savings that might be achieved in steel utilisation).

The state of the art is such that anyone faced with a cutting stock problem is likely to have to develop his own method, unless the Gilmore and Gomory approach is feasible. Such methods are almost bound to be heuristic. However, note should be taken that the amount of literature dealing with the cutting stock problem is very limited due to the fact that many models, procedures and programming have been developed commercially and hence kept confidential.

CHAPTER 9

DEVELOPMENT OF TECHNIQUES OF TWO/THREE-DIMENSIONAL CUTTING STOCK PROBLEM

9.1 Introduction

The aim of this chapter is to present the survey of relevant papers concerned with the two-dimensional cutting stock problem and take into account sheets with defective areas. Gilmore and Gomory (1965) formulate the 2-dimensional trim loss problem in a similar manner to the one-dimensional, with the complication that the cutting patterns are now for rectangular sheets. The methods adapted by other authors and their merits over the Gilmore and Gomory's methods are also presented. Finally a brief review of a three-dimensional cutting stock problem is discussed.

9.2 The Two-Dimensional Cutting Stock Problem

The problem of cutting a bill of rectangular pieces from stock sheets is commonly encountered in the shipbuilding, garment, glass, metal cutting and electronics industries. The primary objective of such a cutting operation is the generation of good parts layouts with minimal trim losses. This process is often referred to as *parts nesting* or*layout* Hinxman (1979).

An equivalent definition for the layout problem is as follows:

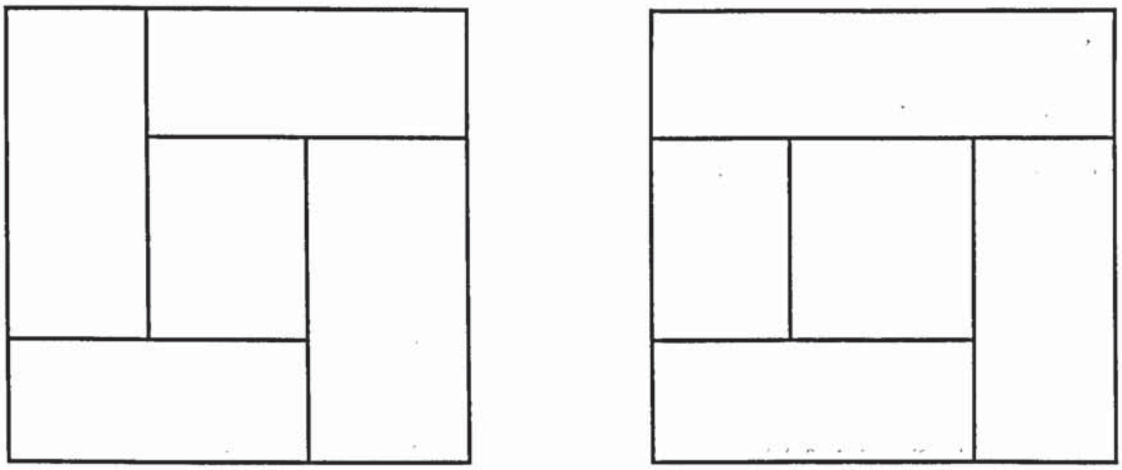
minimise the number of stock sheets used while cutting a bill of materials,

$$\{(l_1, w_1), (l_2, w_2), \ldots\ldots\ldots, (l_n, w_n)\}$$

where (l_i, w_i) are the length and width dimensions of the i^{th} rectangular piece. The demand for each type of piece is an integer d_i, $i = 1, \ldots\ldots, n$.

Numerous papers Wang, (1981), Chambers and Dyson (1976) Eisemann (1957) Eilon and Christofides (1971) and Christofides and Whitlock (1977) on 2-D CSP and 2-D nesting have been published to give a full account of the techniques and procedures involved in deriving solutions to these problems.

The CSP literature contains several algorithms that require substantial computing time and power to solve problems of modest size Hinxman (1979) Furthermore, these algorithms are bound by the "guillotine" constraint i.e. all cuts must go from one edge of the rectangle to the opposite edge. Finally, they do not guarantee that the bills of material will be exactly met, without overruns or underruns.



Example of a non-guillotine cutting pattern   Example of a guillotine cutting pattern

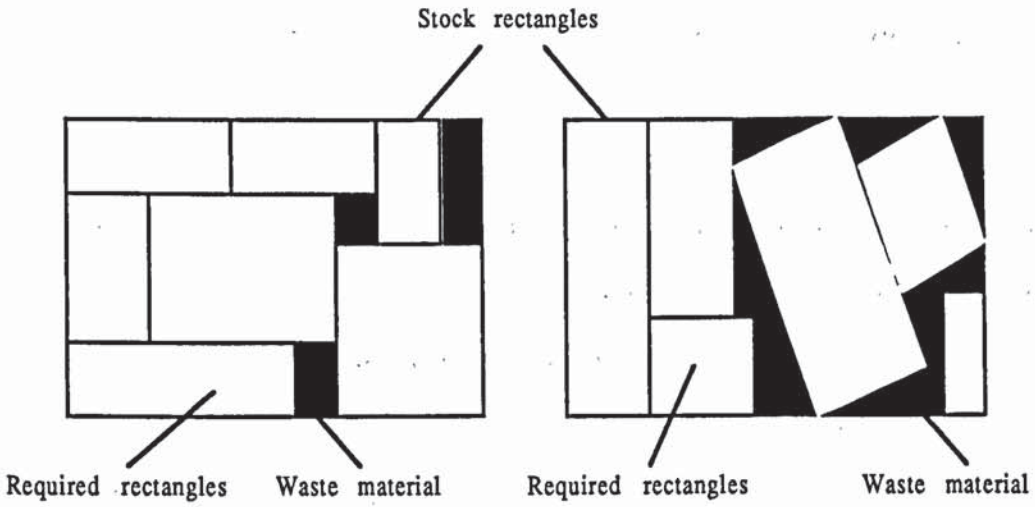
Figure 9.1 Two Examples of Cutting Operations.

However in more recent papers Hinxman (1976 and 1977) other algorithms and techniques have been developed to generate patterns that cut a given set of rectangles without the edge to edge constraint. As the technology of the cutting processes advances, in particular where flame, laser, or waterjet cutting torches are used the need to generate complex and feasible cutting patterns to minimise the waste becomes apparent.

9.3 The Cutting Patterns and Layout

A set of instructions for the cutting of a stock unit is a *cutting pattern* . The cutting patterns are of the type, *orthogonal* and *non-orthogonal* shape. The type and dimensions of each pattern is bound by the process capablity, and constraints specified by the machine/machines in the process route.

De Cani (1978) obseves that on occasion less stock need be used to satisfy an order if the restriction that all the cuts must be parallel to the sides of the stock sheet is removed. However, according to Hinxman (1979) there does not seem to be any computational studies of the use of non-orthogonal cutting.



Example of an orthogonal cutting pattern Example of a non-orthogonal cutting pattern

(a) (b)

Figure 9.2 Two Examples of Cutting Patterns.

The rectangular shears consist of blades that are mounted at right angles to each other. Setting the blades to the proposed generated pattern will produce the desired shapes and hence produces lower trim losses Israni et al. (1984), see Figure 9.2. Other cutting machines and torches capable of producing more complex cuts will also result in lower scrap, and more accurate cuts.

However, practical aspects of 2-D CSP require special treatment and in some cases extensive modification of theoretical techniques to suit a particular environment. Hinxman states that, where an algorithm method is iterative, it may be converted to a heuristic method by causing it to be terminated before all the iterations have been performed (i.e. before the optimum value is reached). The criterion for termination may be the computational cost of the iterations so far performed, or it may be that the values of solutions produced by successive iterations differ by less than a prescribed amount. In these cases a cut-off heuristic will be said to have been adopted.

When a set of feasible solutions is being generated, it may be decided to use the first one that satisfy some criterion. Haessler (1971) uses the term *aspiration level* for such a criterion.

Gilmore and Gomory (1961, 1963, 1964 and 1966) have developed the most powerful techniques in the field of cutting stock problems and generation of cutting patterns. However, their techniques are restricted to 2D-CSP. Most papers Chang and Wysk (1983) Hinxman (1976) Metzger (1958) and Tulkoff (1985) in this field have in one way or another adapted the Gilmore and Gomory techniques.

Christofides and Whitlock (1977) propose a tree-search algorithm for the solution of two-dimensional, single plate, and rectangular cutting stock problems in which guillotine cuts are considered (see Figure 9.1). During the solution process, a dynamic programming procedure is used to produce upper bounds to limit the size of the search.

The constrained two-dimensional cutting problem is therefore defined as follows:

Let a large rectangle $A_0 = (L_0, W_0)$ (i.e., of length L_0 and width W_0 units) be given, together with a set R of m smaller rectangular pieces;

$$R = \{(l_1, w_1), (l_2, w_2), \dots, (l_m, w_m)\},$$

each piece in R having associated with it a value v_i and a maximum nimber b_i of pieces of type i that can be cut from A_0.

The problem is to find the maximum value of:

$$z = \sum k_i v_i$$

so that

$$k_i <= b_i, \quad i = 1, ..., m$$

k_i non-negative integer variables

and there exists a series of cuts on A_0 so that k_i pieces of type i in R can be cut from A_0.

In order to distinguish between the given pieces in set R and the rectangles produced by the cuts on A_0 at any stage during the cutting process, is referred to the former as "pieces" and the latter as "rectangles."

It is also assuumed that L_0, w_0, and l_i, w_i, $i = 1, ..., m$ are integers and that the cuts on the rectangles are to be made in integer steps along the x or y axes. This limitation is not serious since in practice the actual dimensions can be scaled up. It should also be noted that the orientation of the pieces is considered to be fixed, i.e. a piece of length l and width w is not the same as a piece of length w and width l.

9.4 General Form of 2-Dimensional Knapsack Function

Gilmore and Gomory define the two-dimensional knapsack function G as: rectangles of positive dimensions $(l_i, w_i), i = 1, ..., m$ that have nonnegative values $\Pi_1, ..., \Pi_m$ associated with them; then $G(x, y)$ is the maximum of $\Pi_1 Z_1 + ... + \Pi_m Z_m$, where Z_1, ..., Z_m are nonnegative integers such that there exists a way of dividing a rectangle (x, y) into Z_i rectangles (l_i, w_i), for $i = 1, ..., m$. For example, in Figure 9.3 a permitted division of the rectangle (x, y) into two rectangles marked (1), two rectangles marked (2), and one rectangle marked (3) is illustrated. In that case, $G(x, y)$ is known to be at least $2\Pi_1 + 2\Pi_2 + \Pi_3$ if Π_i is the value of a rectangle marked (i). The calculation of $G(x, y)$ for given x and y is not an easy task. Fortunately, for many practical cutting stock problems, the calculation of another knapsack function sufices. F is defined like G except that in dividing a rectangle (x, y) into Z_i rectangles (l_i, w_i) for $i = 1, ..., m$, the following restriction is imposed: The division must take place by a series of straight lines that extend from one edge of a rectangle to an opposite edge, parallel to the other two edges; this is known as "guillotine cuts." Here by a 'rectangle' is meant either the original rectangle (x, y) or a rectangle obtained from it by one or more guillotine cuts. In Figure 9.2 a permitted division of the rectangle (x, y) in the calculation of $F(x, y)$ is illustrated. In that case, $F(x, y)$ is known to be at least $\Pi_1 + \Pi_2 + 2\Pi_3 + 2\Pi_4$.

Because of the way F has been defined in terms of guillotine cuts it satisfies two fundamental divide-in-two inequalities:

$$F(x_1 + x_2, y) >= F(x_1, y) + F(x_2, y)$$

and

$$F(x, y_1 + y_2) >= F(x, y_1) + F(x, y_2)$$

Hence from the above disscusion it is clear that the rectangles (l_i, w_i) of cost Π_i, $i = 1, \ldots, m$, are given, the knapsack function F(x, y) defined from them the following three sets of inequalities:

$$F(x, y) >= 0 \qquad \ldots\ldots.(1)$$

$$F(x_1 + x_2, y) >= F(x_1, y) + F(x_2, y) \qquad \ldots\ldots.(2)$$

$$F(x, y_1 + y_2) >= F(x, y_1) + F(x, y_2) \qquad \ldots\ldots.(3)$$

$$F(l_i, w_i) >= \Pi_i \qquad \ldots\ldots.(4)$$

$$(i = 1, \ldots\ldots, m)$$

In equalities (2) and (3) are self evident, but inequality (1) and (4) are a consequence of the permitted method of cutting a large rectangle (x, y) into the smaller rectangles (l_i, w_i). F is not the only function to satisfy these inequalities, although it is the minimal function in the sense of the following theorem.

F is a knapsack function defined from the rectangles (l_i, w_i) with values Π_i, $i=1, \ldots, m$, if and only if F satisfies (1), (2), (3) and (4). For any G satisfying (1) to (3),

$$F(x, y) <= G(x, y) \qquad \text{for all } x \text{ and } y.$$

The value of the knapsack function F for arguments x and y is the maximum value that one can obtain from a rectangle (x, y) by dividing it into the smaller rectangles (l_i, w_i) of values Π_i by a series of guillotine cuts. F satisfies (1) to (3). There remains therfore to prove that F satisfies (4) also. The following proof carried out by induction on the total number k of cuts needed to achieve the value $F(x, y)$ from the rectangle (x, y).

If $F(x, y)$ is achieved by no cutting, that is $k = 0$, then necessarily either $x = l_i$, $y = wi$, and $F x, y) = \Pi_i$ for some i, or $x = y = 0$. In the former case since G satisfies (3), $G(x, y) >= F(x, y)$. In the latter case, $F(0, 0) = 0$ because of (1) and (2) and therefore $G(0, 0) >= F(0, 0)$ by (1).

Assume therefore that (4) is true for those x and y for which $F(x, y)$ is achieved with k or fewer cuts. Also for those achieved with $K+1$ cuts.

$$\text{Let} \quad F(x, y) = Z_1 \Pi_1 + ... + Z_m \Pi_m,$$

where Z_i is the number of rectangles (l_i,w_i) used to achieve $F(x, y)$. At least one of the $k+1$ cuts divides (x, y) into two rectangles and without loss in generality one can assume that the cut produces two rectangles (x_1, y) and (x_2, y), where $x_1 + x_2 = x$. Within the rectangle (x, y) it is assumed that for each i there are contained Z_i^1 of the *ith* rectangle and within (x_2, y), Z_i^2 of the *ith* rectangle, where $Z_i = Z_i^1 + Z_i^2$. By (2),

$$F(x_1 + x_2, y) >= F(x_1, y) + F(x_2, y)$$

and by the definition of F ,

$$F(x_1, y) >= Z_1{}^1\Pi_1 + ... + Z_m{}^1\Pi_m$$

and

$$F(x_2, y) >= Z_1{}^2\Pi_1 + ... + Z_m{}^2\Pi_m$$

but

$$F(x_1 + x_2, y) = Z_1\Pi_1 + ... + Z_m\Pi_m$$

so that

$$F(x_1, y) = Z_1{}^1\Pi_1 + ... + Z_m{}^1\Pi_m$$

and

$$F(x_2, y) = Z_1{}^2\Pi_1 + ... + Z_m{}^2\Pi_m$$

By the induction assumption:

$$G(x_1, y) >= F(x_1, y)$$

and

$$G(x_2, y) >= F(x_2, y)$$

for any G satisfying (1) to (3). But since such a G satisfies (2) it follows that:

$$G(x_1 + x_2, y) >= G(x_1, y) + G(x_2, y) >= F(x_1, y) + F(x_2, y) = F(x_1, x_2, y)$$

and therefore;

$$G(x, y) >= F(x, y) \qquad \text{since } x_1 + x_2 = x.$$

Since a minimal function F satisfying (1) to (3) is necessarily unique, Gilmore and Gomory's theorem is established. The characterisation of a knapsack function given in the above theorem is a characterisation of the function for given rectangles (l_i, w_i) of values Π_i.

Gilmore and Gomory (1966) state that the above theorem does have as an interesting consequence that the problem of computing the knapsack function F, for given (l_i, w_i) and Π_i and for a finite range of its argument is a linear programming problem. For let $X(x, y)$ denote a variable for each (x, y) in the range for which F is to be computed. Since by the above theorem

$$F(x, y) <= X(x, y)$$

when

X(x,y) satisfies (1) to (30, it follows that:

$$\Sigma_x\Sigma_y F(x, y) <= \Sigma_x\Sigma_y X(x, y)$$

F is therefore the solution to the linear programming problem of minimising $\Sigma_x\Sigma_y X(x, y)$ subject to the inequalities obtained by substituting X for F in (1) to (3).

An efficient method for computing F is by a modified dynamic programming technique that is based upon the functional equation:

$$F(x, y) = \max \{F_0(x, y), F(x_1, y) + F(x_2, y), F(x, y_1) + F(x, y_2)\} \qquad \text{........(5)}$$

$x >= x_1 + x_2,$

$0 < x_1 <= x_2,$

$y >= y_1 + y_2,$

and

$0 < y_1 <= y_2$

where

$$F_0(x, y) = \max \{ 0, \Pi_j, l_j <= x \text{ and } w_j <= y \} \qquad \text{........(6)}$$

Consider the two auxiliary functions F_1 and F_2 defined as follows:

$F_1(x, 0) = 0,$

$$F_1(x, y) = \max \{F(x, y-1), F(x_1, y) + F(x_2, y)\} \qquad \text{........(7)}$$

$x >= x_1 + x_2,$

$0 < x_1 <= x_2,$

$F_2(0, y) = 0,$

$$F_2(x, y) = \max \{F(x-1, y), F(x, y_1) + F(x, y_2)\} \qquad \text{........(8)}$$

$y >= y_1 + y_2,$

$0 < y_1 <= y_2.$

It follows immediately from (5) that F also satisfies:

$$F(x, y) = \max \{F_0(x, y), F_1(x, y), F_2(x, y)\} \qquad \text{.......(9)}$$

Gilmore and Gomory (1966) introduced the concept of *step-off point* and *step-off length*. These concepts have been generalised for the purpose of the two-dimensions. In (7) any point (x_2, y) is a step-off point and any x_1, $0 < x_1 <= x_2$, a step-off length. Similarly in (8) any point (x, y_2) is a step-off point and any y_1, $0 < y_1 <= y_2$, a step-off length.

In order to make these ideas completely precise, Gilmore and Gomory have described an algorithm for the calculation of $F(x, y)$, $0 <= x <= L$, $0 <= y <= w$. The algorithm requires three memory grids $F^*(x, y)$, $l^*(x, y)$ and $w^*(x,y)$ although frequently in practice $l^*(x, y)$ and $w^*(x,y)$ can be combined into one. The grids $l^*(x, y)$ and$w^*(x,y)$ are used, like $l^*(x)$ for $F^*(x)$, to record how the value $F^*(x, y)$ is achieved. When the computation is completed $F^*(x, y) = F(x, y)$ while the memory grids l^* and w^* will have computed two functions l and w which is now defined.

Thus the set of possible step-off lengths is a function of the step-off point and the direction in which the step-off is to take place, that is whether the step-off is to take place along a line of constant y or constant x. Nevertheless, different orderings of the step-off lengths can be considered; for example, in decreasing size or in increasing price density; that is in increasing size of $F(x_1, y)/x_1$ for a step-off length x_l or of increasing size of $F(x, y_1)/y_1$ for a step-off length y_l. To simplify the presentation of the algorithm, the decreasing order of size has been chosen, although it should be emphasized that other orders can be considered.

If the value F(x, y) for a point (x, y) has been achieved by stepping-off with a length x_1 from a step-off point (x_2, y), where $0 < x_1 <= x_2$, $x_1 + x_2 = x$, and $F(x, y) = F(x_1, y) + F(x_2, y)$, choose for the value of l for the point (x, y) the smallest step-off length used in achieving F(x, y). Otherwise it is x. Therefore define for x , $y >= 1$,

$$l(x, y) = \min \{ x_1, x; 0 <= x_1 <= x - x_1 \},$$

$$F(x, y) = F(x_1, y) + F(x-x_1, y) \qquad ....(10)$$

and similarly

$$w(x, y) = \min \{ y_1, y; 0 <= y_1 <= y-y_1 \},$$

$$F(x, y) = F(x, y_1) + F(x, y-y_1) \qquad ....(11)$$

Prior to (x_2, y_2) being used as a step-off point the situation is as follows: $F^*(x, y) = F(x, y)$ for all x and y, $0 <= x <= x_2$ and $0 <= y <= y_2$, as well as $x_2 < x <= L$ and $0 <= y < y_2-1$. Further, $F^*(x, y_2+1) = F_2(x, y_2+1)$ for all x, $0 <= x < x_2$ where F_2 is defined in (8). The value $F^*(x, y_2)$ for $x_2 < x <= L$ is the maximum of $F_2(x, y_2)$ and some incompletely computed value $F_1(x, y_2)$, as F_1 is defined in (7). This situation is illustrated in Figure 9.3. During the step-offs from (x_2, y_2) along $y = y_2$, $F^*(x_2+1, y_2)$ becomes $F(x_2+1, y_2)$ while during the step-offs along $x = x_2$, $F^*(x_2, y_2+1)$ becomes $F(x_2, y_2+1)$. Along the line $y = y_2$, one steps-off with all lengths x_1, $0 < x_1 <= x_2$ and along the line $x = x_2$ with all lengths y_1, $0 < y_1 <= y_2$.

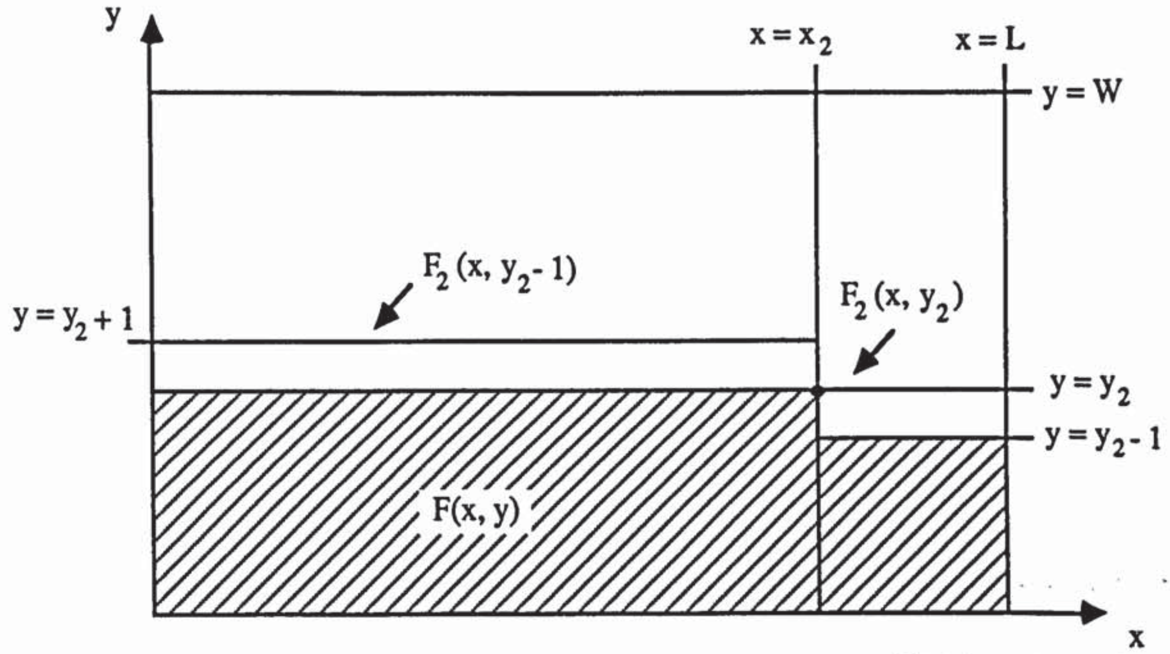


Figure 9.3 The Analysis of Gilmore & Gomory's Algorithm.

Wang (1981) has proposed two combinatorial methods that generate constrained cutting patterns by successive horizontal and vertical build of ordered rectangles. Each of the algorithms uses a parameter to bound the maximum acceptable percentage of waste they create. Like Gilmore and Gomory, Wang also assumes that acceptable cutting patterns are limited to those of guillotine type. However, Wang takes an opposite approach to determining all feasible guillotine patterns. Instead of enumerating all possible cuts that can be made on the stock sheet, his combinatoric algirithms find the guillotine cutting patterns by successively adding the rectangles to each other. Wang defines the problem as follows:

Let $H * w$ be a rectangular stock sheet having hight H and width W, and let R be a set of rectangles $R_1, R_2, ..., R_n$ with dimensions $h_1 * w_1, h_2 * w_2, ..., h_n * w_n$. Determine the guillotine pattern with minimum trim waste that cuts the rectangles using no more than b_i replicates of rectangle R_i for $i = 1, 2, ..., n$ in the pattern. The problem can also be in the form:

$$\max_G \sum_i x_i h_i w_i$$

subject to

$$0 <= x_i <= b_i$$

$$x_i \quad \text{integer} \qquad (i = 1, 2, \ldots, n)$$

In this formulation x_i is an integer indicating the number of times the rectangle R_i appears in a guillotine cutting pattern G. The mathematical program requires an additional constraint to ensure that H and W are not exceeded.

As discussed by Christofides and Whitlock, every guillotine pattern has an equivalent normalised guillotine form. In this form, all rectangles in a pattern are left-justified at the lowest possible position in the stock sheet and placed adjacently as in Figure 9.5.

For a given guillotine pattern G, the corresponding guillotine rectangle S is defined to be the rectangle that contains the rectangles R_i of G and has the smallest possible hight and width dimensions. Figure 9.4 shows the guillotine rectangle corresponding to Figure 9.3. Furthermore, it is considered as equivalent, two guillotine rectangles containing the same set of rectangles and having the same height and width dimensions.

9.4.1 The Method

A horizontal build of two rectangles $A_1 = p_1 * q_1$ and $A_2 = p_2 * q_2$ is a rectangle Su having dimensions max $(p_1, p_2) * (q_1 * q_2)$ and containing A_1 and A_2. A vertical build of A_1 and A_2 is a rectangle S_v of dimensions $(p_1 + p_2) * \max(q_1, q_2)$ that contains A_1 and A_2. An example is given in Figure 9.5. Also the hight and width dimensions of S_u and S_v do not exceed the corresponding dimensions of the stock rectangle.

Two parameters $ß_1$ and $ß_2$ are used to denote the maximum acceptable percentage of waste of any guillotine rectangle T generated in the algorithm. Firstly, $ß_1$ is measured with respect to the area of the stock sheet $H * W$ and secondly, $ß_2$ is measured with respect to the area (T) of T.

Wang's approch is to build horizontally and vertically the rectangles $R_1, R_2, ..., R_n$ with each other. The resulting rectangles are then added to the original rectangle R_i to form a larger set of guillotine rectangles. The process is repeated so that each successive horizontal and vertical build forms a larger guillotine rectangle from two smaller guillotine rectangles. In addition the two algorithms proposed by Wang reject any waste exceeding $ß_1$ or $ß_2$ respectively. Guillotine rectangles that contain more than b_i replicates of a rectangle R_i are also eliminated from further consideration. The two algorithms proposed by Wang are as follows:

9.4.2 Wang's Algorithm One

define $L^{(0)} = F^{(0)} = \{ R_1, R_2, ..., R_n \}$

so that

$0 <= ß_1 <= 1$ and $k = 1$.

compute (A)

$F^{(k)}$ which is the set of all rectangles T satisfying conditions:

(i) T is formed by a horizontal or vertical build of two rectangles from $L^{(k-1)}$,

(ii) the amount of trim waste in T does not exceed $ß_1HW$, and

(iii) those rectangles R_i appearing in T do not violate the bound constraints $b_1, b_2, ..., b_n$.

set $L^{(k)} = L^{(k-1)} \cup F^{(k)}$

and remove any equivalent rectangle patterns from $L^{(k)}$. If $F^{(k)}$ is nonempty,

set k ----- $k+1$ and go to (A), otherwise.

set $M = k - 1$.

choose the rectangle of $L^{(M)}$ that has the smallest total trim waste when placed in the stock sheet $H * W$.

Figure 9.4 A Normalised Guillotine Pattern *G.*

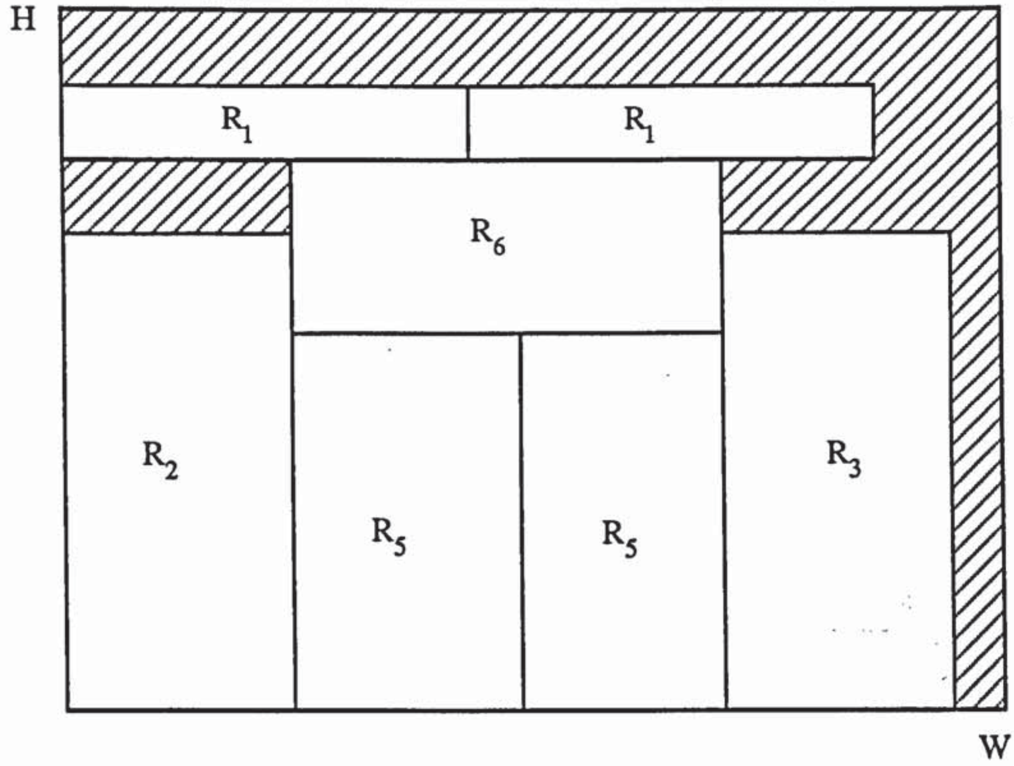
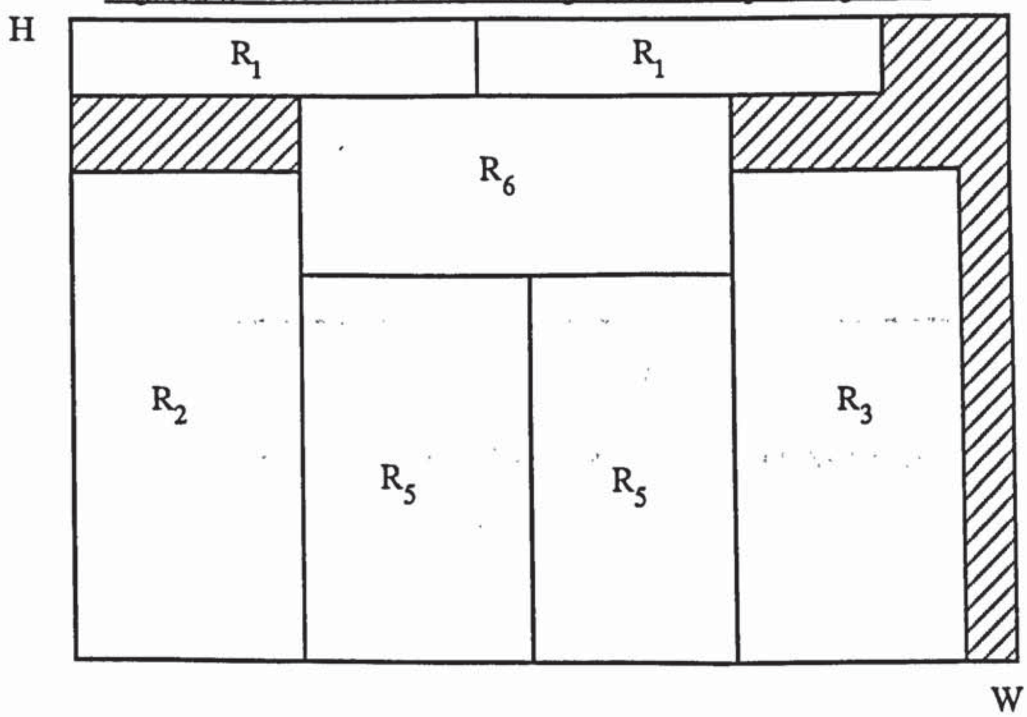


Figure 9.5 The Guillotine Rectangle, S, Corresponding to *G.*



9.4.3 Algorithm Two

This is a modification of algorithm one. β_2 is replaced by β_1 in (A) and the following condition replaces condition (ii):

the amount of trim waste in T does not exceed $\beta_2 \cdot (T)$.

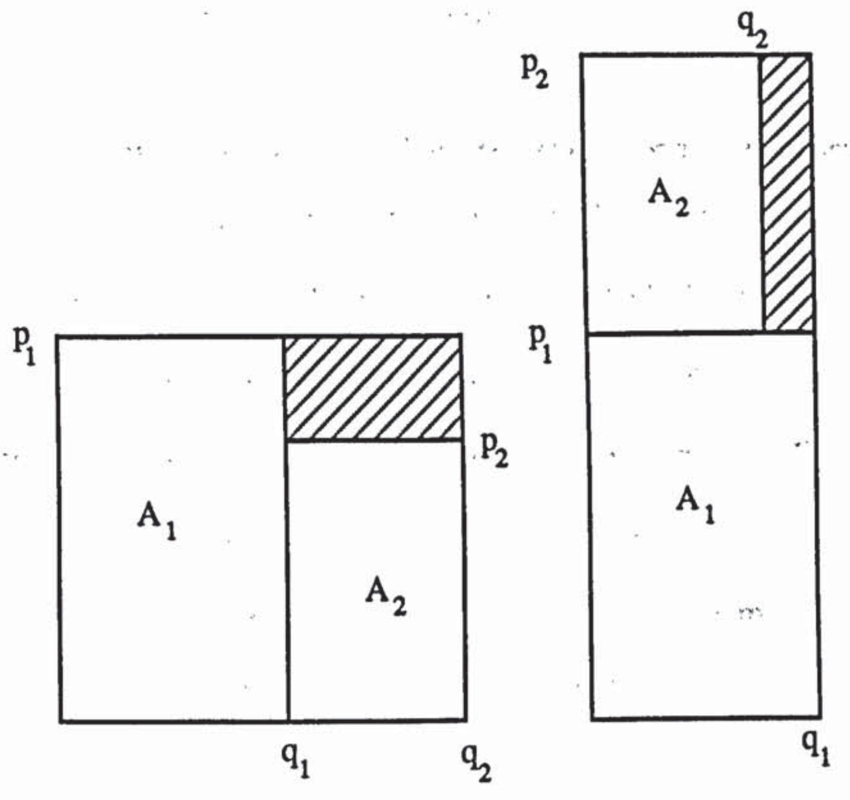


Figure 9.6 A Horizontal and Vertical Build of A_1 and A_2.

9.5 Error Bounds and Optimality Conditions

The rectangles in $L^{(M)}$ of the above algorithms are guillotine rectangles formed by a sequence of horizontal and vertical builds of the rectangles $R_1, R_2, ..., R_n$. The addition of two rectangles in this manner to form a larger rectangle is the reverse of guillotine cutting the larger rectangle into two smaller ones. With β_1 or β_2 set equal to one, either algorithm generate $L^{(M)}$ as the set of all possible guillotine rectangles that can be constructed from the R_i subject to the bound constraints. With β_1 or β_2 set equal to zero, $L^{(M)}$ will consist of rectangle patterns having no trim waste.

When the value of the parameters are increased from zero to one for a given constrained problem, the number of rectangles generated by reapplying the algorithms with these parameter values increases dramatically. With larger values of the β_i, better solutions are obtained claims Wang, since the size of $L^{(M)}$ increases. However, the algorithm then requires more computing time and storage space to generate each $L^{(k)}$. Thus these algorithms, may usually be employed with small values for β_1 and β_2. It is therefore, advantageous to obtain upper bounds measuring the closeness of the optimal solution of the given problem to the best solutions obtained by the algorithms.

Let μ denote the trim waste in a rectangle $S = h * w$ that was formed by a sequence of horizontal and vertical builds of $R_1, R_2, ..., R_n$ subject to the constraints $b_1, b_2, ...,$ bn and the value of β_i. μ is reffered to as the inner waste. Let ç be the total trim waste associated with S when it is placed in the stock sheet $H * W$ as in figure 9.6. Then ç = HW - hw + μ.

If ç* denotes the minimum amount of total trim waste that can be attained by any guillotine pattern which cuts R_1, R_2, ..., R_n from the stock sheet without rotating the rectangles or violating the constraints, then ø can be defined to be the set of all guillotine rectangles that have waste ç* Wang (1981) discusses this in detail.

9.6 The 3-Dimensional Cutting Stock Problem

Not surprisingly, there is few papers written spescifically on the 3-D CSP. Gilmore and Gomory (1964) have made an attempt to explain the three-stage three-dimensional problem by using their two-stage two-dimensional problem techniques.

An example of this type of problem occures in the cutting of graphite blocks for anodes. By proceeding the same manner similar to the 2-D CSP the problem was solved by Gilmore and Gomory keeping one of the dimensions constant.

In the rolling industry, the dimensions of the castings or blanks being rolled are variable in the three axies. In theory, when a piece of metal is rolled between two rolls, all the dimensions can be varied at the same time but in practice, depending on the direction of rolling operation, the variation in the third dimension is small enough to be totally ignored and the dimension is assumed to be constant. This may be a useful point to consider when searching for the most suitable technique to determine cutting patterns.

9.7 Summary

In this chapter three methods of cutting patterns were identified;

a) Guillotine cutting as shown in Figure 9.1.

b) Non-Guillotine cutting again shown in Figure 9.1.

c) Non-Orthogonal Cutting shown in Figure 9.2.

From the above list much effort was spent in analysing the guillotine cutting pattern by various authors in this field. Moreover, it was revealed in Section 9.2.4 that most papers in the field of two-dimensional cutting stock problem have adapted the Gilmore-Gomory techniques.

Section 9.4 it was shown that Gilmore and Gomory's quick and efficient knapsack method based on the modified dynamic programming algorithms. This is shown by the modified equation formulated in (5). This section also demonstrates the two concepts of step-off point and step-off length theories generalised for the purpose of the two-dimensional cutting pattern as formulated in (7-9).

Although a fairly detailed study of the two-dimensional cutting stock problem was carried out, it was not required to use these techniques within the rolling mill environment.

However, the applications of these techniques are more relevant in the garment or glass industry where smaller rectangular pieces are cut from a larger rectangular piece. This is also known as the 'nesting' technique.

CHAPTER 10

IMPLEMENTATION OF CSP SOLUTIONS AND THE DEVELOPMENT OF ROUNDING TECHNIQUE

10.1 Introduction

The generation of practical cutting patterns to maximise yield and minimise production costs is the core of the computer aided material optimisation system developed for the rolling mill industry. In this chapter, the combinatorial problem and functional requirements of both the Sheet mill and the Strip mill at Company B are considered. These are then explained to show in both cases, the one-dimensional CSP nature of the cutting patterns required at each rolling stage.

The manufacturing constraint file described in Chapter 5, is used to read the constraints imposed by the planners on the CSP problem at each stage of rolling where a cutting pattern is being considered. The constraints are dependent on the capacity of the machine (i.e. max length, max width, gauge, etc.). Constraints such as the mill speed, maximum gauge reduction per pass, and material handling limitations can also be considered when the system is fully implemented.

Finally, the one-dimensional CSP is explored in more detail, as a preliminary to the development of a simple rounding technique, which is needed when the basic CSP algorithm gives a solution involving fractional quantities.

Later, it will be shown that a technique based on Gilmore & Gomory's (1961 and 1963) methods for the one-dimensional cutting stock problem, already discussed in Chapter 8, can be used to tackle this problem. Gilmore & Gomory acknowledged that their methods will not, in general, give integer solutions and their recommendation of simple rounding to meet integer constraints proved to be highly unsatisfactory in the rolling-mill context. It is therefore necessary to develop algorithms to find practical integer solutions to this type of problem.

As discussed in Chapter 3, one of the main aims of the process planners is to maximise yield. On many occasions, this means that the planners have to produce a cutting pattern which generates minimum scrap. The sources of scrap and their types generated in this environment are explained below.

10.2 Sources Of Scrap

From amongst the many problems discussed in Chapter 3, the most important to both Companies A and B is 'material yield'. We have seen from Chapter 2 that sheets and metal strips are produced by a combination of rolling and cutting operations; both types of operation may affect yield through factors which may be classified into two groups:

1. Manufacturing allowances.
2. Process planners' decisions on dimensions and blank cuts (i.e. manual generation of some kind of cutting pattern).

We shall refer to the former as *unavoidable production losses*, and the latter as the *cutting stock problem*.

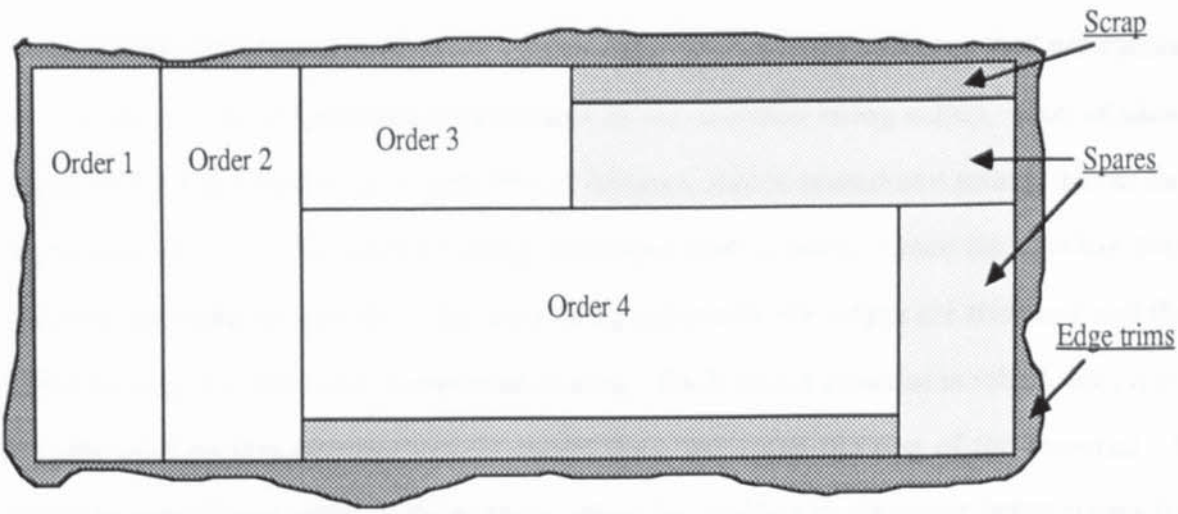


Figure 10.1a An Example of a Two-Dimensional Cutting Pattern With Three Types of Material Losses.

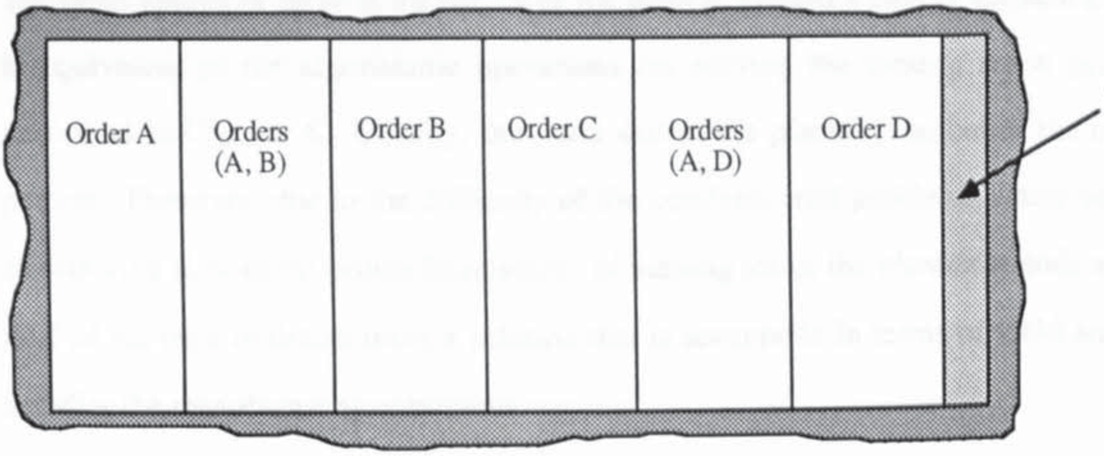


Figure 10.1b An Example of One-Dimensional Cutting Pattern With Material Scrap.

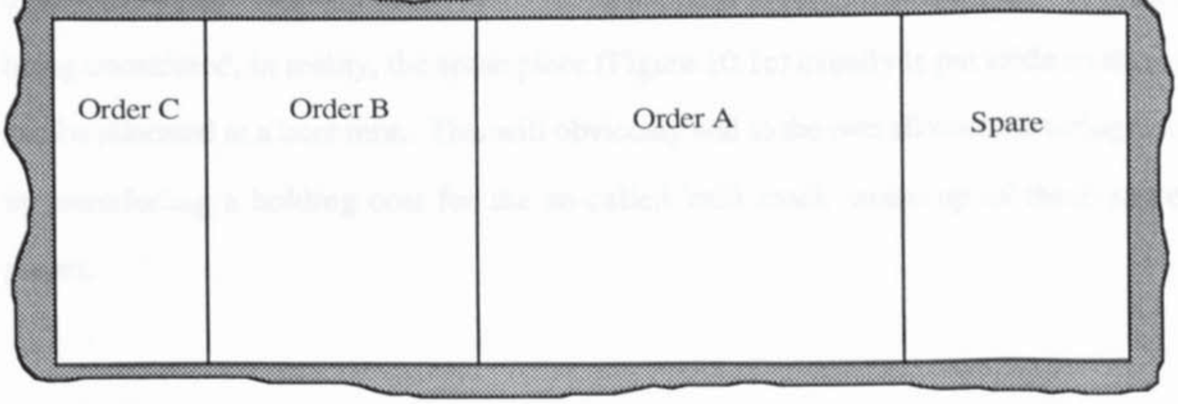


Figure 10.1c An Example of One-Dimensional Cutting Pattern With Material Spare.

The unavoidable production losses, as the name implies refer to losses that are caused by rolling and the characteristic behaviour of the material being rolled. One of these losses is the *edge trimming* (Figure 10.1). When a slab is heated and rolled, due to the high reduction ratios during hot rolling, the edges start to crack. Once the slab has been rolled to the required gauge by the hot rolling operation, the edges are trimmed and the rolled piece is cut down for further cold rolling. Each time a material is rolled, the edges are trimmed so that propagation of cracks does not scrap the rest of the material. It might be argued that some of these losses might be avoided by choosing better routes for hot rolling, but this possibility will be put on one side for the moment.

The other source of scrap is the choice of the process planner's cutting sequence. This is equivalent to the algorithmic operations for solving the cutting stock problem described in Chapter 8. Usually, the more skilful the planner, the better the cutting pattern. However, due to the difficulty of the combinatorial problems which arise in considering how to cut orders from stocks of varying sizes, the planner spends a great deal of his time in determining a solution that is acceptable in terms of yield and also satisfies the manufacturing constraints.

In the calculation of material yield, the planners include the pieces for which there are no orders scheduled. Although this results in higher yield for the particular batch of orders being considered, in reality, the spare piece (Figure 10.1c) usually is put aside so that it can be allocated at a later time. This will obviously add to the overall manufacturing cost by introducing a holding cost for the so-called 'mill stock' made up of these spare pieces.

As mentioned previously, the shape of each blank is dependent on the maximum capacity of the rolling mill being considered at that stage and the mix of orders for which the cutting pattern is being considered, this therefore leads us to generate a manufacturing constraint file so that the CSP problem formulated by this system can use this file to generate a practical cutting pattern for that stage.

10.3 Practical Implications of Selected Objective Functions

It was shown in Chapter 8 that, in practice, the Gilmore-Gomory algorithm satisfies the orders with stocks that have the lowest costs per unit length, whenever the lengths of such stocks are compatible with the order lengths. This has two main advantages over methods which only seek to optimize on yield.

1. An attempt is made to satisfy orders by using material which has been produced on the cheapest production routes.

2. Any mill stock and scrap produced is from low-cost stock.

The 'cost' approach of the Gilmore-Gomory method makes the objective function ideal when considering the stock selection, since the undesired stocks can be assigned a lower cost during the selection process. If dimensionally suitable, this stock will be selected and the remaining orders will carry on. However, this technique imposes an additional updating of stock master record in terms of evaluating the cost of producing the mill stocks. Although a method for the costing of work in progress at any stage in rolling is considered by Husseini (1989), it is merely for the cost evaluation of each order. Therefore the true cost of evaluating the mill stock can be determined by his method. However, if these costs were to be considered when the CSP algorithm is considering feasible cutting patterns, it finds that using slabs with no operation cost added are the cheapest stocks available and hence they become the basis for the proposed cutting pattern and the mill stocks will hardly be considered. This obviously, is not practical and somehow the mill stocks must appear cheaper so that the system considers those prior to selecting slabs. Considering the mill stock first has several advantages:

(a) Lower requirements for expensive slabs with long lead times.

(b) Utilisation of mill stock and hence smaller storage space.

(c) Lower overall production costs, since most mill stocks have gone through the expensive process of the hot mill rolling.

(d) Higher yield factors are achieved since the mill stocks are usually near the order dimensions and hence require less rolling and cutting operations. This overlaps with point (c), which results in lower production costs for any batch of orders.

(e) Less opportunity for the refinery to use the mill stocks as scrap for producing castings of lower grades, which results in higher manufacturing costs indirectly caused by the planners not utilising the mill stock.

Therefore the following method of evaluating the cost of each mill stock is considered.

Let c_i be the cost of the ith item per unit length in the mill stock,

L_i be the weight of the ith item in the mill stock,

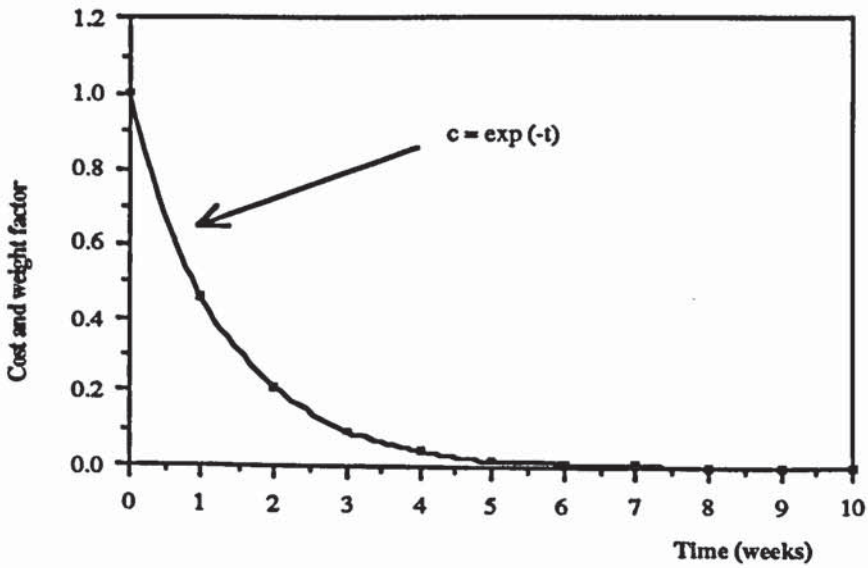
and t_i be the time elapsed from when the item was produced.

We need to establish a relationship between the cost of an item and the time that item is being kept on the shelf. Now the longer it remains in store, the more value is being added in the form of managing that item and holding costs. The item in store also has a higher probability of being used in the refinery and thus loosing its original manufacturing cost mentioned in Point (e) above. Therefore, the following relationship is used.

$$c_i = \exp(-t_i)$$

This results in the following figure:

Figure 10.2 The Cost Calculation Criterion



From this graph, it can be seen that a slab being considered for rolling it has time ti = 0 and cost and weight factor $c_i = 1$; any other stocks will have this factor taking some value between 1 and 0. This therefore ensures that mill stocks must be used prior to considering the slabs. An additional constraint is required to cater for the 'cost/weight factor'. This means that when a spare piece is produced and recorded in the stock control system, a date is recorded. At a later date, when the CSP algorithm is considering the cost of an item, it sorts all mill stocks by their weight or volume, then assigns the oldest production date to the smallest piece and calculates the cost for all stock items that fall within the gauge range being considered.

This will therefore ensure that all mill stocks of the same gauge range may be assigned a relative cost per unit length at time t_i.

10.4 Design Of CSP Software Logic

The concept of Computer Aided Material Optimisation (CAMO) was discussed in Chapter 7. Here in this section the logic of how CSP uses the manufacturing constaint file and the inventory system is explained and the CSP software logic developed is investigated.

The flowchart shown in Figure 10.3 shows the working engine of the CAMO system and its functions. The CSP function is heart of the CAMO system which uses the combinatorial methods of one-dimensional CSP developed by Gilmore and Gomory (1961). Figure 10.4 shows the CSP software logic which has been developed and the relevant data files necessary to drive the algorithm to produce practical cutting patterns. When the optimal solution is generated, there may be fractional requirements of several stock lengths. In practice, these fractions would be very rarely acceptable, especially at the first rolling stage where most of the stock items being considered are slabs (the cost of which can vary between several hundred and several thousand pounds).

Because of this, a method of rounding these fractions to give an optimum cutting pattern with integer solutions has been developed and is discussed in Section 10.5.

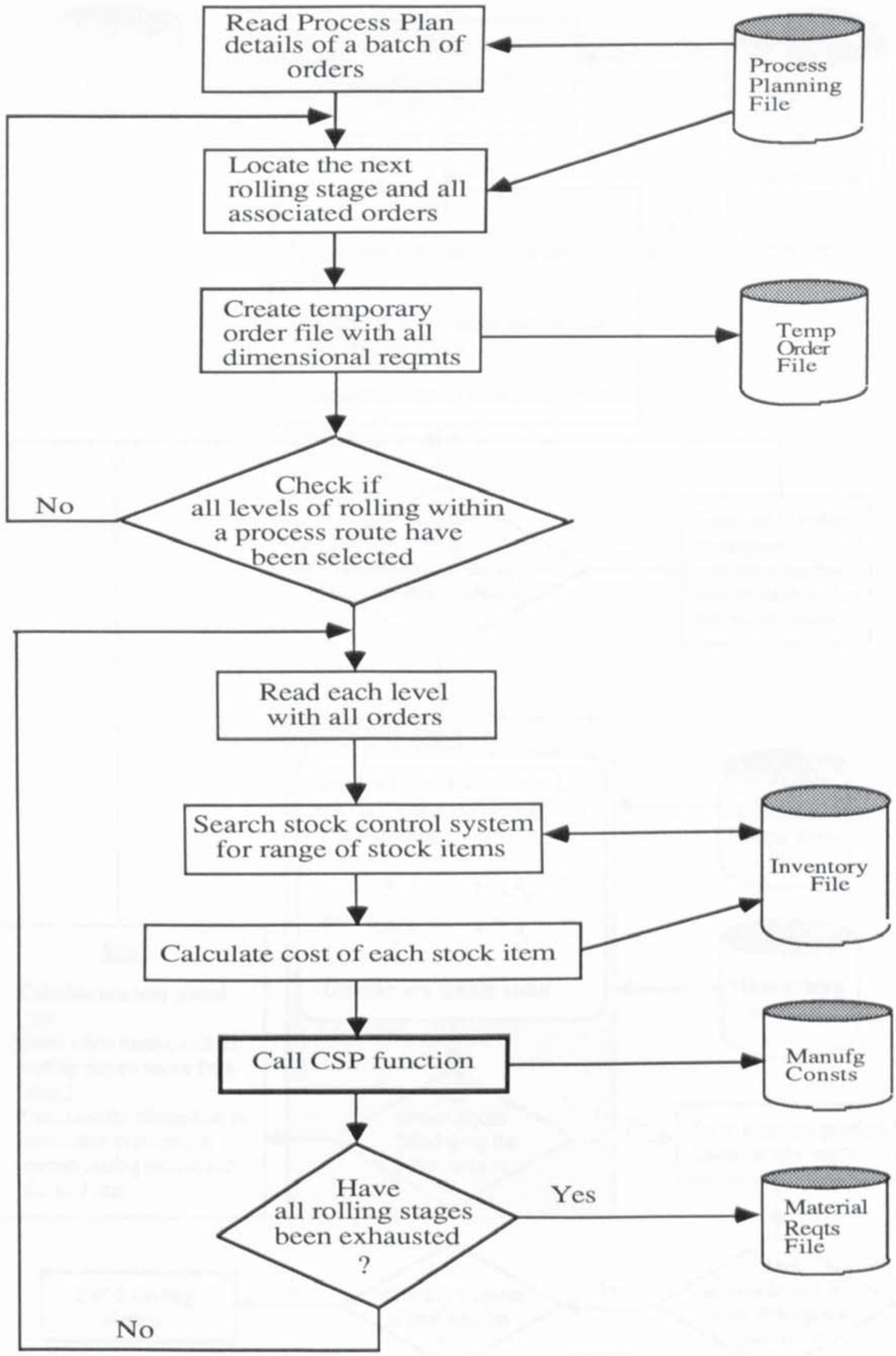


Figure 10.3 The Functionality of the CAMO System and the Integration of the CSP Module.

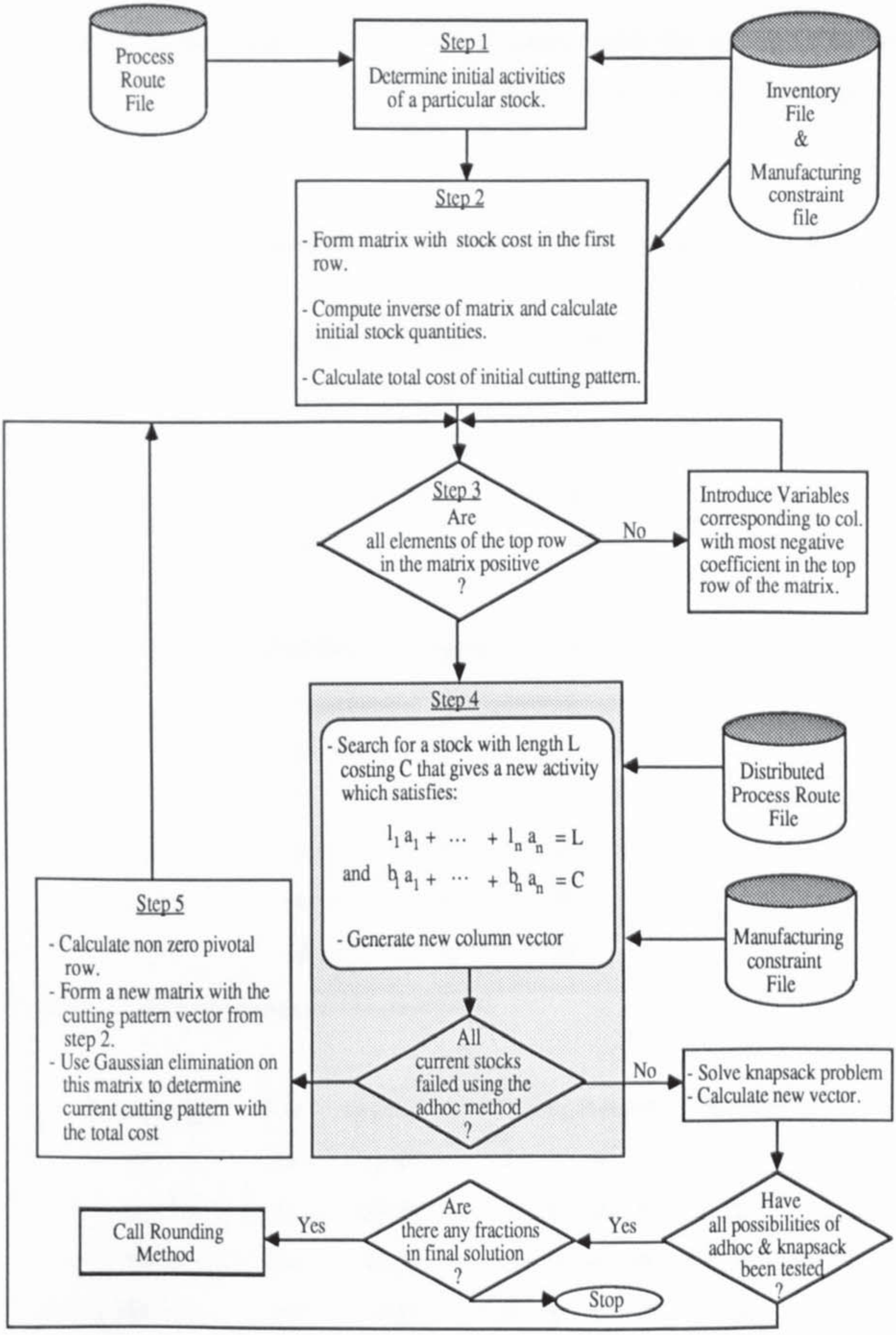


Figure 10.4 The One-Dimensional CSP Software Logic.

However, to demonstrate the impractical fractional solutions generated by the Gilmore-Gomory method, consider the following example of which a brief account was given by Wilson et al. (1988).

The following orders are to be satisfied at minimum cost from the available stocks.

Order No.	Length	Quantity
1	5	50
2	7	40
3	8	40
4	10	50

Stock No.	Length	Cost
1	11	12
2	12	11
3	15	16
4	17	18
5	19	20
6	35	34

The Gilmore-Gomory solution to this problem is:

Stock Length	Cost	Quantity	Cutting Pattern*	Activity
35	34	16.67	1 0 0 3	1
12	11	33.33	1 1 0 0	2
35	34	10.00	0 0 4 0	3
35	34	1.33	0 5 0 0	4

Table 10.1 The Optimum One-Dimensional CSP Solution

*Column i of "cutting pattern" shows the number of pieces of order i produced by this activity, e.g. activity 1 produce 1 piece of length 5 (order 1) and 3 pieces of length 10 (order 4).

The solution yields a total cost of 1318.67, which is the theoretical optimum. Note that the quantites for activities 1, 2, and 4 include fractional parts.

Now suppose that there happens to be these exact fractions (2/3, 1/3 and 1/3 respectively) in stock as non-standard lengths left over from previous orders. This would give the following lengths for activities requiring fractional parts:

Activity	Length	
1	23.33	
2	4.00	
4	11.67	Total length = 39

and the remaining orders yet to be satisfied as a result of these fractions are:

Order No.	Length	Quantity	
1	5	2	
2	7	1	
4	10	2	Total length = 37

Comparison of these last two lists shows that, although the total stock length is apparently sufficient to satisfy all the orders, the orders cannot actually be satisfied because of the way in which the stock lengths are partitioned.

Figures 10.5 (a-g) show the schematic representation of the solution given in Table 10.1. It can be seen that although the solution is claimed to be optimum, it is not possible in practice to achieve such a pattern. Studying the fractional parts of the cutting pattern in Figure 10.5 (b), the recommended usage of stock of length 35 is 23.33 units hence producing 11.67 units of scrap. Figure 10.5 (d) suggests a usage of only 4 units of the stock of 12 unit length and Figure 10.5 (g) is only utilising 11.67 units of the stock of 35 unit length. Therefore adding all fractional unit lengths together, the total fractions become 39 units.

The shortages can be seen from Table 10.1 to be 2 x 10, 1 x 5 and 2 x 7. Therefore, the total shortages are also equal to the total fractions (i.e. 39 units). Hence it can be deduced from the above analysis and the generated cutting patterns shown in Figure 10.5, that the solution is trying to 'weld' the fractional pieces together to satisfy the desired shapes.

In practice, this is totally unacceptable and therefore a method of rounding the solutions is developed so that it satisfies a set of slightly modified constraints, at the same time very nearly minimising the cost and generating integer solutions.

In Section 10.5 the problems involved in this environment is analysed in depth and a method of rounding the solutions up or down is recommended. The solution is then integrated within the CSP function to provide the planners with practical integer solutions based on the manufacturing constraints and available stock.

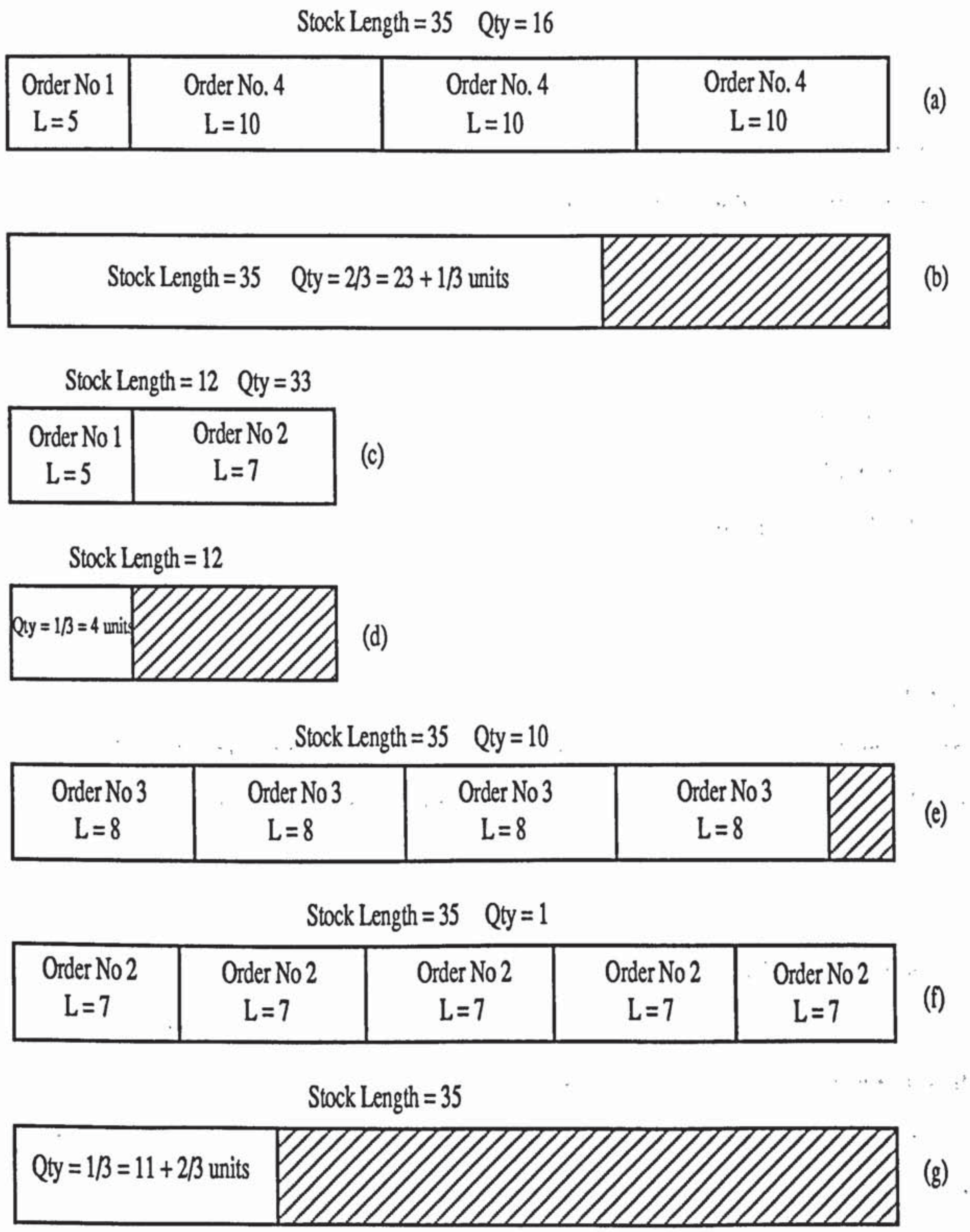


Figure 10.5 The Optimum Cutting Pattern Generated for the Example in Section 1.5.

10.5 The Rounding Algorithm

As the starting point a non-integer solution generated by the Gilmore-Gomory algorithm is used and then applied to a special rounding algorithm which uses dynamic programming. The method is illustrated in continuing with the example introduced in Section 10.4.

If inexpensive materials were being dealt with, Gilmore & Gomory's suggestion of simple rounding up might have be acceptable. In this case, however, rounding 16.67 to 17, 33.33 to 34 and 1.33 to 2 would give a total cost of 1360 (an increase of 3.1 per cent) and 43 units of scrap.

Confining the attention to the stocks in the Gilmore-Gomory solution (which has already been noted to be the cheapest stocks), therefore, a search is made for a rounding pattern which gives a lower cost and less scrap, but it must contain at least 37 length units to satisfy all outstanding orders.

Using a binary code (0 for "round down", 1 for "round up") it is possible to generate all rounding patterns as shown in the Table 10.2, where the bottom row shows deviation from the required 37 units. The advantages of this technique is that the system would generate all the possibilities from rounding every activity down to rounding all activities up. The optimum solution however, falls some where in between. It can also be seen at a glance that the smallest positive elements at the bottom row of Table 10.2 are the stock which need to be selected. This easy and powerful method of selecting items which guarantee to yield the optimum solution can be used to generate the final cutting pattern.

stock length \ rounding pattern		0	1	2	3	4	5	6	7
35	+11.67 -23.33	0	1	0	1	0	1	0	1
12	+8 -4	0	0	1	1	0	0	1	1
35	+23.33 -11.67	0	0	0	0	1	1	1	1
Surplus (+) Shortage (-)		-39	-4	-27	+8	-4	+31	+8	+43

Table 10.2 The Rounding Pattern Sequence

It is clear that rounding patterns 0, 1, 2, and 4 cannot satisfy all the orders, Patterns 3 or 6 might satisfy them - if so, they would produce less scrap than Patterns 5 or 7. To check this, it is required that a method of comparing stock lengths and order lengths be developed.

At this stage, the objective of pure cost minimisation is abandoned and a new optimisation problem is introduced, namely maximising the numbers of pieces of each unsatisfied length subject to their total length not exceeding that available on the chosen stock and the total numbers exactly equalling the shortfall left by rounding down the non-integer solution.

The longest unsatisfied order length is 10, call this l_1, and the corresponding outstanding quantity is 2, call this s_1. Similarly for the other two orders: $l_2 = 7$, $s_2 = 2$ and $l_3 = 5$, $s_3 = 1$.

Let x_{ij} denote the number of pieces of length l_i to be made from stock length L_j. Then use dynamic programming to solve a set of knapsack problems of the type;

$$\text{Maximise } x_{1j} + x_{2j} + x_{3j} \qquad \text{.............} \quad (1)$$

$$\text{subject to } 10\,x_{1j} + 7x_{2j} + 5x_{3j} \leq L_j$$

$$\text{and } \Sigma\, x_{1j} = s_1,$$

$$\Sigma\, x_{2j} = s_2,$$

$$\Sigma\, x_{3j} = s_3.$$

The first constraint is applied in turn to each of the stock lengths, so L= L_j for each j considered. Following standard dynamic programming practice and temporarily dropping the j suffix for simplicity, the following is defined.

$$F_1^*(L) = \max_{x_1 \leq [L/l_1]} \{ x_1 \}$$

$$F_2^*(L) = \max_{x_2 \leq [L/l_2]} \{ x_2 + F_1^*(L\text{-}l_1x_1) \}$$

$$F_3^*(L) = \max_{x_3 \leq [L/l_3]} \{ x_3 + F_2^*(L\text{-}l_2x_2) \}$$

and $x_i^*(L)$ is the optimal value of x_i when cutting from a stock length L, which is considered as varying from zero to the maximum L_j to be examined (35 in our example).

It is important to use a form of dynamic programming solution which shows all possibilities when there are ties for $x_i^*(L)$, so that the otherwise attractive quick method of Gilmore and Gomory (1963) which is used to find the initial (non-integer) solution is not suitable here.

The resulting table for this example is shown as Table 10.2, where tying values for $x_i^*(L)$ are separated by commas. This single table can then be used to examine the suitability of any rounding pattern and compare possible cutting patterns from the stocks under consideration. For example, let us start by considering rounding pattern 3. Let $L_1 = 35$ and $L_2 = 12$. Solve the knapsack problem for L_1 by entering the dynamic programming table at the bottom right, where L = 35 and it is found that $x_3^*(35) = 0$ or 1, so $x_3^* = 0$ or 1 indicating 0 or 1 pieces of length $l_1 = 5$ can be cut. Then $x_2^* = x_2^*(35\text{-}0)$ or $x_2^*(35\text{-}5)$, etc. until the possibilities of cutting from $L_1 = 35$ are exhausted. All the possibilities are shown in the top left portion of the cutting sequence diagram (see Figure 10.1), where figures in the nodes denote remaining stock. From that diagram it can be seen that there are four possible sequences - A, B, C, and D - each of which leaves at least one order unsatisfied. Therefore, introduce $L_2 = 12$ and read off values from the dynamic programming table, starting with the highest numbered order which is still unsatisfied (i.e. for which $\Sigma\, x_{ij} \neq s_i$). Thus for sequence A, s_1 and s_2 have been satisfied but not s_3, so start with $x_3^*(L)$ for L = 12. This is seen to give $x_3^* = 1$, which satisfies $\Sigma\, x_{1j} = 1$, therefore sequence A has been completed, satisfying all orders and leaving 7 units of scrap from L_2 and 1 unit from L_1.

L	[L/10]	$x_1^*(L)$	$F_1^*(L)$	[L/7]	$x_2=0$	$x_2=1$	$x_2=2$	$x_2^*(L)$	$F_2^*(L)$	[L/5]	$x_3=0$	$x_3=1$	$x_3^*(L)$	$F_2^*(L)$
0	0	0	0	0	0	-	-	0	0	0	0	-	0	0
1	0	0	0	0	0	-	-	0	0	0	0	-	0	0
2	0	0	0	0	0	-	-	0	0	0	0	-	0	0
3	0	0	0	0	0	-	-	0	0	0	0	-	0	0
4	0	0	0	0	0	-	-	0	0	0	0	-	0	0
5	0	0	0	0	0	-	-	0	0	1	0	1	1	1
6	0	0	0	0	0	-	-	0	0	1	0	1	1	1
7	0	0	0	1	0	1	-	1	1	1	1	1	1,0	1
8	0	0	0	1	0	1	-	1	1	1	1	1	1,0	1
9	0	0	0	1	0	1	-	1	1	1	1	1	1,0	1
10	1	1	1	1	1	1	-	1,0	1	2	1	1	1,0	1
11	1	1	1	1	1	1	-	1,0	1	2	1	1	1,0	1
12	1	1	1	1	1	1	-	1,0	1	2	1	2	1	2
13	1	1	1	1	1	1	-	1,0	1	2	1	2	1	2
14	1	1	1	2	1	1	2	2	2	2	2	2	1,0	2
15	1	1	1	2	1	1	2	2	2	3	2	2	1,0	2
16	1	1	1	2	1	1	2	2	2	3	2	2	1,0	2
17	1	1	1	2	1	2	2	2,1	2	3	2	2	1,0	2
18	1	1	1	2	1	2	2	2,1	2	3	2	2	1,0	2
19	1	1	1	2	1	2	2	2,1	2	3	2	3	1	3
20	2	2	2	2	2	2	2	2,1,0	2	4	2	3	1	3
21	2	2	2	3	2	2	2	2,1,0	2	4	2	3	1	3
22	2	2	2	3	2	2	2	2,1,0	2	4	2	3	1	3
23	2	2	2	3	2	2	2	2,1,0	2	4	2	3	1	3
24	2	2	2	3	2	2	3	2	3	4	3	3	1,0	3
25	2	2	2	3	2	2	3	2	3	5	3	3	1,0	3
26	2	2	2	3	2	2	3	2	3	5	3	3	1,0	3
27	2	2	2	3	2	3	3	2,1	3	5	3	3	1,0	3
28	2	2	2	4	2	3	3	2,1	3	5	3	3	1,0	3
29	2	2	2	4	2	3	3	2,1	3	5	3	4	1	4
30	3	3	3	4	3	3	3	2,1,0	3	6	3	4	1	4
31	3	3	3	4	3	3	3	2,1,0	3	6	3	4	1	4
32	3	3	3	4	3	3	3	2,1,0	3	6	3	4	1	4
33	3	3	3	4	3	3	3	2,1,0	3	6	3	4	1	4
34	3	3	3	4	3	3	4	2	4	6	4	4	1,0	4
35	3	3	3	5	3	3	4	2	4	7	4	4	1,0	4

Table 10.3 The Resulting Knapsack Table Produced by the Objective Function (1)

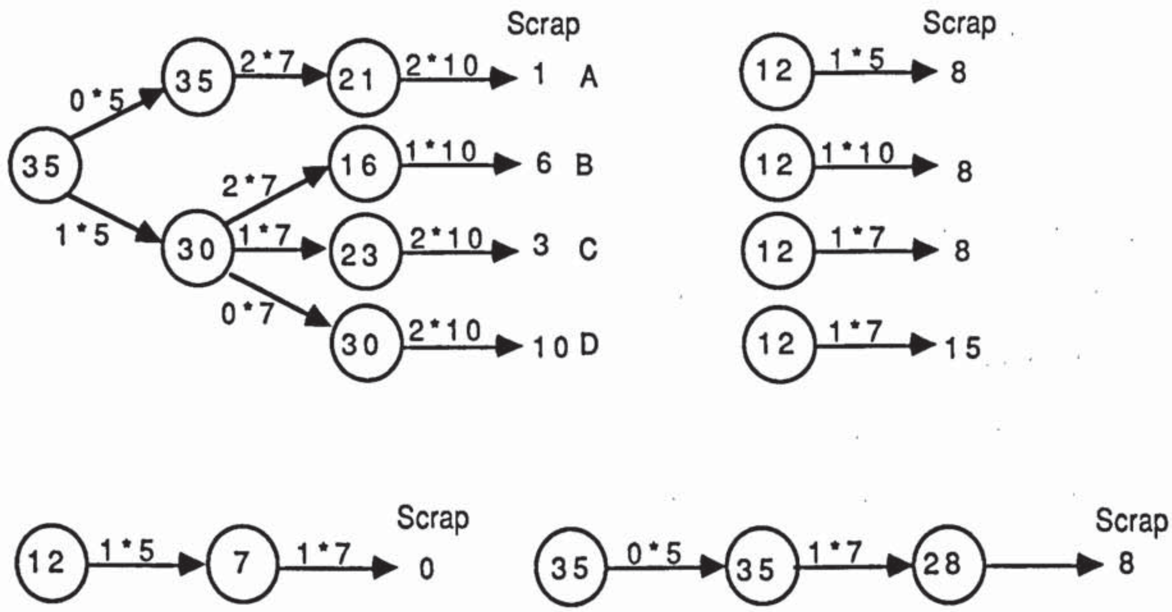


Figure 10.6 The Cutting Sequence Diagram

Similar computations are completed for sequences B, C, and D, leading to the results shown in the cutting sequence diagram (see figure 10.6). Note that in sequence D, although $x_1*(30) = 3$, we only select $x_{11} = 2$ because $s_1 = 2$. The order in which the stock lengths are considered are then reversed, so $L_1 = 12$ and $L_2 = 35$, giving sequence E which is shown in the lower part of the diagram.

Examining the five sequences, it can be seen that all except D satisfy all the outstanding orders (D is short by one piece of length $l_2 = 7$), then A, B, C and E have to be examined on the basis of;

(1) total scrap,

(2) total number of cuts.

It can be seen that sequence E minimises both of these and is therefore chosen as the best integer solution so far.

The total cost of stock for the all-integer solution using sequence E in the final stage is 1326. This is not the minimum value for an all-integer solution, since we can achieve a cost of 1325 by selecting four pieces of length 12 stock, using the same dynamic programming table, and entering the table four times. The disadvantage of multiple use of the same stock as well as rounding of each stock in the Gilmore-Gomory solution is that the number of searches will expand without any guaranteed pay-off.

10.6 Further Applications of The Rounding Algorithm

The following example provides a broader view of the rounding pattern where the Gilmore-Gomory solution leaves more than two types of stock to be considered, with more fractional parts in the solution.

We now have six orders to be made from ten possible types of stock, as detailed below.

Order No.	Length	Quantity
1	5	50
2	6	40
3	10	20
4	12	20
5	13	20
6	30	50

Stock No.	Length	Cost
1	11	12
2	13	15
3	15	18
4	18	21
5	25	29
6	30	37
7	36	42
8	40	45
9	46	53
10	50	65

The Gilmore- Gomory solution to this problem is:

Stock Length	Cost	Quantity	Cutting Pattern	Activity
40	45	20	0 0 1 0 0 1	1
40	45	13.33	2 0 0 0 0 1	2
11	12	23.33	1 1 0 0 0 0	3
36	42	6.67	0 0 0 3 0 0	4
13	15	20	0 0 0 0 1 0	5
36	42	16.67	0 1 0 0 0 1	6

Table 10.4 The Optimum Suggested CSP Solution of the Above Orders.

The cost generated for this solution is £3060.00 which is the theoretical optimum. Here again some activities (2, 3, 4 and 6) include fractional stocks.

The resulting orders that are contained within these fractions are:

Order No.	Length	Quantity	
1	5	1	
2	6	1	
4	12	2	
6	30	1	Total length = 65 units

Following the approach developed in the previous example, we generate a binary coded table as shown in Table 10.4. This shows all possible rounding patterns and the deviation from the required 65 units.

Table 10.6 The Rounding Pattern Sequence

Stock Length \ Rounding Pattern	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
40	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
11	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
36	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
36	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Surplus (+) Shortage (-)	-65	-25	-54	-14	-29	+11	-18	+22	-29	+11	-18	+22	+7	+47	+18	+58

Using the dynamic programming approach of the example above, Table 10.6 is generated where the tying values of $x_i^*(L)$ are again separated by commas. This table is used to examine the suitability of any rounding pattern and comparison of cutting patterns from the stocks under consideration. From Table 10.3 it can be seen that unlike

Len	L/30	x1*(L)	F1*(L)	L/12	x2=0	x2=1	x2=2	x2*(L)	F2*(L)	L/6	x3=0	x3=1	x3*(L)	F3*(L)	L/5	x4=0	x4=1	x4*(L)	F4*(L)
0	0	0	0	0	0	-	-	0	0	0	0	-	0	0	0	0	-	0	0
1	0	0	0	0	0	-	-	0	0	0	0	-	0	0	0	0	-	0	0
2	0	0	0	0	0	-	-	0	0	0	0	-	0	0	0	0	-	0	0
3	0	0	0	0	0	-	-	0	0	0	0	-	0	0	0	0	-	0	0
4	0	0	0	0	0	-	-	0	0	0	0	-	0	0	0	0	-	0	0
5	0	0	0	0	0	-	-	0	0	0	0	-	0	0	1	0	1	1	1
6	0	0	0	0	0	-	-	0	0	1	0	1	1	1	1	1	1	1,0	1
7	0	0	0	0	0	-	-	0	0	1	0	1	1	1	1	1	1	1,0	1
8	0	0	0	0	0	-	-	0	0	1	0	1	1	1	1	1	1	1,0	1
9	0	0	0	0	0	-	-	0	0	1	0	1	1	1	1	1	1	1,0	1
10	0	0	0	0	0	-	-	0	0	1	0	1	1	1	2	1	1	1,0	1
11	0	0	0	0	0	-	-	0	0	1	0	1	1	1	2	1	2	1	2
12	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	2	1	2	1	2
13	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	2	1	2	1	2
14	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	2	1	2	1	2
15	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	3	1	2	1	2
16	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	3	1	2	1	2
17	0	0	0	1	0	1	-	1	1	2	1	1	1,0	1	3	1	3	1	3
18	0	0	0	1	0	1	-	1	1	3	1	2	1	2	3	2	3	1	3
19	0	0	0	1	0	1	-	1	1	3	1	2	1	2	3	2	3	1	3
20	0	0	0	1	0	1	-	1	1	3	1	2	1	2	4	2	3	1	3
21	0	0	0	1	0	1	-	1	1	3	1	2	1	2	4	2	3	1	3
22	0	0	0	1	0	1	-	1	1	3	1	2	1	2	4	2	3	1	3
23	0	0	0	1	0	1	-	1	1	3	1	2	1	2	4	2	4	1	4
24	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	4	2	4	1	4
25	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	5	2	4	1	4
26	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	5	2	4	1	4
27	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	5	2	4	1	4
28	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	5	2	4	1	4
29	0	0	0	2	0	1	2	2	2	4	2	2	1,0	2	5	2	5	1	5
30	1	1	1	2	1	1	2	2	2	5	2	3	1	3	6	3	5	1	5
31	1	1	1	2	1	1	2	2	2	5	2	3	1	3	6	3	5	1	5
32	1	1	1	2	1	1	2	2	2	5	2	3	1	3	6	3	5	1	5
33	1	1	1	2	1	1	2	2	2	5	2	3	1	3	6	3	5	1	5
34	1	1	1	2	1	1	2	2	2	5	2	3	1	3	6	3	5	1	5
35	1	1	1	2	1	1	2	2	2	5	2	3	1	3	7	3	6	1	6
36	1	1	1	3	1	1	2	2	2	6	2	3	1	3	7	3	6	1	6
37	1	1	1	3	1	1	2	2	2	6	2	3	1	3	7	3	6	1	6
38	1	1	1	3	1	1	2	2	2	6	2	3	1	3	7	3	6	1	6
39	1	1	1	3	1	1	2	2	2	6	2	3	1	3	7	3	6	1	6
40	1	1	1	3	1	1	2	2	2	6	2	3	1	3	8	3	6	1	6

Table 10.5 The resulting knapsack table for example 2.

the previous example, where two patterns (3 and 6 in Table 10.2, each involving one unit of length 35 and one of length 12) were possible, here there is only one feasible stock length which generates 7 units of scrap (pattern 12). Therefore, taking $L_1 = 36$ and $L_2 = 36$, and using Table 10.4 it can be seen that $x_4^*(36) = 1$, indicating 1 piece of length $l_1 = 5$ can be cut. Then $x_3^* = x_3^*(36\text{-}5)$; this gives 1 piece of length $l_2 = 6$; $x_2^* = x_2^*(31\text{-}6)$ indicates 2 pieces of length $l_3 = 2{*}12 = 24$, leaving 1 unit of scrap from the stock length 36. This is shown by the cutting sequence diagram Figure 10.7. Taking $L_2 = 36$, and entering the knapsack table with $x_1^*(36)$ indicates 1 piece of length $l_6 = 30$ leaving 6 units of scrap. Therefore the total scrap generated from the suggested cutting pattern would amount to 1+ 6 = 7 units. The cost of this sequence is 42*2 = 84. Therefore, the total cost of stock for the all-integer solution is £3069.00.

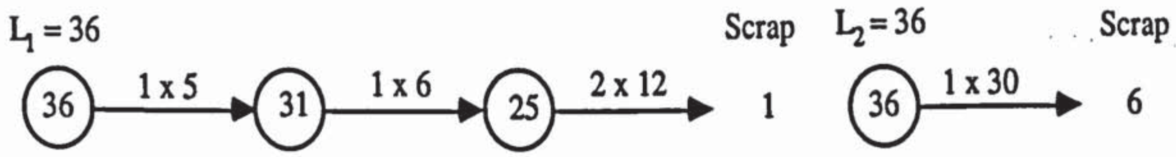


Figure 10.7 The Cutting Sequence Diagram.

The Gilmore-Gomory suggestion of rounding all fractions to the nearest integer number would yield a cost of £3126.00 as opposed to our optimum method costing £3069.00. The rounding method clearly shows an improvement of about 2 per cent in cost and 7 units of scrap as opposed to 58 units.

It should be noted that, although the two examples considered above showed that the pattern with the lowest amount of surplus material gave a feasible solution, this may not always be the case. It might happen that the distribution of the surplus between the chosen stock lengths meant that we could not cut all the unsatisfied orders. In this case, the pattern giving the next highest surplusshould be examined, and so on until a feasible solution is found. It will rarely be necessary to use this full form of the rounding algorithm since the pattern with the lowest surplus usually yields the required solution.

10.7 Summary

In Section 10.2, the sources of scrap and some of the terminology were discussed. Figure 10.1 also demonstrates the two-dimensional and one-dimensional aspects of CSP and the orientation of sequencing various orders to form a one-dimensional CSP algorithm.

In Section 10.3 a formula was derived so that it would be possible for the planners to modify costs in such a way as to make the stocks being considered by the CSP function to look attractive from the point of view of their relative cost. When a range of stocks are selected this formula can be used to calculate costs attached to each item. Figure 10.2 shows this relationship with the slabs (i.e. virgin metal) having the highest costs at time zero after rolling.

Sections 10.4 and 10.6 discuss the overall design of the software logic and the detailed working of the CSP function. The manufacturing constraints and stock control files are shown to be an integral part of the CSP system.

Two examples of rounding are considered in detail. From these it is seen that the complexity of the rounding problem is not directly related to the number of orders or stocks to be considered. The first problem, which might be expected to be much simpler, resulted in a larger choice of feasible solutions, while the second (larger) problem only resulted in one feasible solution. In both cases, although, the new rounding algorithm gave an all-integer solution which was very close to the non-integer optimum.

In small-batch production, where the cost of stock is high, automatic rounding-up of Gilmore-Gomory solutions to the CSP to give all-integer solutions may be too costly. It was demonstrated that non-integer Gilmore-Gomory solutions may not have any practical reality. Because of this, a practical method of achieving all-integer solutions close to the Gilmore-Gomory theoretical optimum is required in this kind of production environment.

By concentrating on low-cost stock that has already been selected by the Gilmore-Gomory algorithm, the proposed method, which examines feasible rounding patterns, provides a heuristic solution that results in costs that are well within the boundaries set by crude rounding up or down of the Gilmore-Gomory solution and generally very close to the Gilmore-Gomory theoretical optimum. Also, by restricting the choice to stock lengths which are already being cut as part of the initial solution we avoid charges incurred by handling smaller quantities of other stocks which are only required in the final stages of the solution. Special handling of these small quantities would generally prove to be uneconomic.

The version of the method presented here is reasonably compact and easily implemented for a moderate number of order types. If further improvement is required, more extensive searches can be carried out on the dynamic programming table which forms the core of the proposed method.

The rounding method would be particularly cost effective in companies with high turn over of expensive material. In Company B, where most of this work was carried out an average saving of three percent per year for a total throughput of about 12,500 tonne would amount to something which the management could not easily ignore.

As a result of this work, we have been able to make the benefits of Gilmore & Gomory's cost-based approach available as part of a practical CAPP system, which releases planning staff from the time-consuming task of order mixing and achieves their goal of high yield.

CHAPTER 11

CONCLUSION AND SUGGESTIONS FOR FURTHER WORK

11.1 Case Studies and Problem Definition

After a review of the hot and cold rolling of copper-alloys and the applications in which the resulting products are used, it became clear that there was considerable scope for a new approach to computerised production planning in this industry. The general points observed from the case studies suggested that for any company in this industry to survive, it is essential to implement computerised systems and train their staff to respond rapidly to market changes and to provide the necessary technical and managerial information more efficiently and within narrow time frames.

The almost impossible task of manually achieving manufacturing objectives in a practical environment makes the use of computers highly desirable in the development of systems to overcome the difficulties of planning in the rolling mill industry.

The case studies in Companies A and B gave an insight into the functional requirements for an integrated computerised planning system. From the functional requirements, three major areas were identified and the research work was split into two major sections.

The first section formed the basis of the research work carried out by Husseini (1989), who concentrated on the development of a computerised generative process planning system (CAPP). The second formed the basis of the present thesis: the development of a full functional specification and a proposal for the development of a computer integrated production management system (CAPM) for the rolling mill industry, together with the development of a computerised material optimisation system which can be integrated with the proposed CAPM.

The proposed CAPM system has been devised so that it can be implemented in three phases (Manufacturing Management and Support Functions, Design Management and System Wide Functions and Commercial and Administrative Management Functions), with Phase 1 being the core modules and Phases 2 and 3 the outer shells. The design concept has been a system appropriate in the copper-alloy industry. The only customising which would be required is the modification of various details of the data base and some of the general terms used. The system can be configured to input and output data collected manually or captured automatically (e.g. by the use of bar code readers and of counters at the entry and exit on each work centre).

It is desirable for any CAPM systems to be fully integrated with the activities of the shop floor and the controlling of rolling mill processes. For this purpose, further work is required to investigate the possibilities of automating existing mills with sophisticated hydraulic drive mechanisms and so controlling the mechanical operation of such machines via computers. The next stage would be to establish rolling schedules and feed this information direct to the rolling mill at the time of rolling, as was explained in Chapter 2 Section 2.3.

11.2 System Design and Proposed Model

The functional specifications described in Chapters 4-7 have followed the well-tried design principles laid down by Dixon (1966) and Groover (1980) and other authors who were referred to in those chapters.

The principal modules of the resulting system are:

- Manufacturing Management Functions
- Manufacturing Support
- Computer Aided Material Optimisation (CAMO)

Each of these consists of many sub-modules. Therefore, for the practical development of a commercially viable system a team of well trained programmers are required to carry out the detailed coding and one or more systems engineers would be needed to coordinate the building up of the system. This must be done in such a way that an independent system analyst could modify or expand the system at a later date with a minimum of recoding.

The use of a well established data base management system such as Oracle or Informix would enable the programmers to import data from or export data to third party packages. For example, the prototype system developed in conjuction with Husseini (1989) used the Prolog language to generate the process routes and these could then be linked to the CSP module developed in the present thesis, which used the 'C' language.

If the system is developed using sophisticated data base management systems with Fourth Generation Language tools such as SQL, this would enable the user to customise various reports and facilitate the presentation of data in any format deemed necessary

The hierarchical structure of the modules makes it possible for the systems programmers to use time-phsed life cycles and beta-testing of these modules on customer sites, i.e. checking whether the system meets the customer's requirements.

One of the main achievements of this work was the recognition of the need for an MRP type of system, so that the planners can run the software and determine the total material requirements and the schedule of intermediate rolling stages. This was achieved by breaking down the complete process routes for those orders that had similar routings. Using the idea of distributed MRP and BOM, as developed by Love and Lung (1988), it was possible to generate BOM-type process routes that only contain one stage rolling - i.e. some orders that need to be rolled to some pre-determined gauge, width and length. At this stage, the CSP module would generate the optimum practical cutting pattern and proceed to the next stage. Once the entire process route has been exploded, the total material required for the batch of orders considered can be accumulated to give the net requirements.

Since the CSP module refers to the stock control system each time it selects a new stock, it was possible to provide the facility for deducting the stocks required from the stock-on-hand value and calculating the net requirements. Also, having the lead time stored in the stock master file suggested that purchase-order and works-order report can be produced in order to give the planners a complete picture of what is needed and when to take action.

11.3 The Cutting Stock Problem and The Rounding Technique

The one-and two dimensional CSP was fully explored and a thorough literature survey was presented in Chapters 8 and 9. It was shown in these chapters that a large number of authors have tried to tackle the problems involved in this complex subject. However, most companies that develop practical cutting patterns do not appear to publish their findings nor are they willing to make their systems available commercially. This was the author's experience, for example, when he contacted the Alcoa rolling mills in the USA.

Because of the philosophy behind the route-selection and order-batching module of the CAPP system developed by Husseini (1989), it is only necessary to consider a one-dimensional cutting stock problem since the gauge and width are constant within each batch - only the lengths need cutting.

Having studied a number of techniques for solving the one-dimensional CSP, the Gilmore-Gomory methods (1961) seemed to be the most suitable in this environment. However, this technique needed further modification in terms of rounding the final solution to produce practical integer solutions.

The core of the CSP module is a coding of the Gilmore-Gomory method for the one-dimensional CSP including provision for degeneracy as explained in Chapter 8. It was then shown in Chapter 10 that solutions produced by this core algorithm may imply unrealistic methods of satisfying orders which amount to welding together residual pieces from several stock lengths. Simple rounding up to ensure sufficient material to satisfy all orders can increase material costs considerably. This will be especially true when dealing with material whose manufacturing cost has a high value-added component or when the raw material cost (i.e. cost of slabs is high).

Rather than reopen all the options on available stock, the rounding algorithm developed here has limited the selection of stocks required to satisfy the fractional part of the orders to those low-cost stocks already identified by the Gilmore -Gomory algorithm.

Dynamic programming is used to pack as many unsatisfied orders as possible into a minimum number of additional stock lengths. Although the resulting cutting patterns are not strictly minimising material cost, it has been found that the method always produces patterns which are close to the true theoretical minimum cost, which could only be achieved with non-integer solutions to the CSP.

11.4 Viability of the System

Tests at Company B have demonstrated the essential soundness of the system and its ability to produce production plans which are consistently as good as (and often better than) those now being generated by the semi-manual methods.

The tests concentrated on the order-mixing and cutting-stock features of the system which are peculiar to the operation of rolling mills. From these tests it seems reasonable to conclude that the system could be used with advantage in any rolling mill with a reasonably high throughput of high value material.

Functionality of some of the other modules, such as the Sales Order Processing, Works Order System, SFDC and Supporting Functions have been tested outside the rolling mill environment in other systems developed by the author. These have met with the functional requirements of the Department of Trade and Industry.

The critical factors from the point of view of a company installing such a system are whether their individual company must pay for the whole cost of the software development and whether this cost is to be met entirely from savings in production costs over a short planning horizon of, say, two years.

Decisions on this will be affected by the tonnage of material processed and by accountancy conventions used for recovering costs. The choice of a micro-computer based system has ensured that development and running costs would be low enough for a moderate size company to develop a commercial version of the system for itself, but failing this there seems to be a good opportunity to develop a core system which can be customised for a number of users within the rolling mill industry.

REFERENCES

Adlington, A. G. (1976). "Modern Production Processes for Copper and Copper Alloy Sheet and Strip." Sheet Metal Industries, Feb. 1976.

Anderson, J. C. (1981). Material Requirements Planning. American Production and Inventory Control Society.

Archer, L. B. (1965). Systematic Methods for Designers. London: Council of Industrial Design.

Barron, D. W. (1971). Computer OperatingSystems. London: Cox and Wyman Ltd.

Benz, H. (1974). "Production Control and Scheduling in a Foundry by Means of Electronic Data Processing" Foundry Trade Journal, **11**, April 1974, 395-400.

British Telecom (1989). "Computer Integrated Manufacture", Marketing Publication, London: British Telecom.

Burbridge, J. L. (1978). Principles of Production Control. Plymouth: Macdonald Evans Ltd.

Burcher, P. G. (1985). "Master Production Scheduling (Work Shop Notes)." U.K. BPICS Conf. 180-191.

Chambers, M. L. and Dyson, R.G. (1976). "The Cutting Stock Problem in the Flat Glass Industry - Selection of Stock Sizes". Operational Research Quarterly, **27**, 949-957.

Chang, T. and Wysk, R. (1983). "Integrating CAD/CAM through Autonmated Process Planning". International Journal of Production Research.

Christofides, N. and Whitlock, C. (1977). "An Algorithm for Two-Dimensional Cutting Problems". Operations Research, **25**, 30-44.

Conway, R. W., Maxwell W. L. and Miller L. W. (1967) Theory of Scheduling. Reading: Addison Wesley.

De Cani, P. (1978). "A Note on the Two-Dimensional Rectangular Cutting-Stock Problem". Operational Research, **29**, 703-706.

Dendle D. W. (1978) "Hydraulic Position - Controlled Mill and Automatic Guage Control." Proceedings of an International Conference by The Metal Society, held at University College, Cardiff.

Dixon, R. (1966). Design Engineering. London: McGraw Hill.

Drucker, P. F. (1970). The Effective Executive. London: Pan Books

Dyckhoff, H. (1981). ""A New Linear Programming Approach to the Cutting Stock Problem." Operations Research, **29**, 1092-1104.

Edwards, J. K. (1978). "Automation of Cold Tandem Mills". Proceedings of an International Conference by the Metal Society, held at University College, Cardiff.

Eilon, S. and Christofides, N. (1971). "The Loading Problem". Management Science, **17**, 259-268.

Eilon, S., Watson-Gandy, C., and Christofides, N. (1971). Distribution Management., New York: Hafner Publishing Co.

Eisemann, K. (1967). "The Trim Problem." Management Science, **3**, 279-284.

Engleman, C. (1964) "The past, Present, and future of General Simulation Language" Management Science **11**, 199-208.

Eversheim, W. and Schultz, J. (1985) "A Survey of Computer Aided Process Planning Systems", First CIRP Working Seminar on CAPP, Paris, Jan. 1985.

Farr, M. (1961). Design Management. London: Hutchinson.

Fischer, C. H. (1985). "Expert Systems and Artificial Intelligence." Systems Development, Inland Steel Co., USA Iron and Steel Engineer, Nov. 1989.

Fisher, M., Northup, W., and Shapiro, J. (1974). "Using Duality to Solve Discrete Optimization Problems". Working Paper OR Operations Research Center, M.I.T.

French, M. J. (1971). Engineering Design: The Conceptual Stage. London: Pitman Press.

Fox, R. E. (1983) "OPT Vs. MRP Thoughtware Vs. Software Part I", Inventories and Production Magazine. Nov.-Dec. 1983.

Garey, M., Graham, R., and Ullman, J. (1972). "Worst Case Analysis of Some Memory Allocation Algorithms". Proc. Fourth Annual ACM Symposium on Theory of Computing, May 1972, 143-150.

Garfinkel, R. and Nemhauser, G. (1972). Integer Programming. New York: John Wiley andSons.

Geoffrion, A. (1974). "Lagrangean Relaxation for Integer Programming". Math. Prog Study, **2**, 82-114.

Gilmore, P. C. and Gomory, R.E. (1961). A Linear Programming Approach to the Cutting Stock Problem. Operations Research, **9**, 849-859.

Gilmore, P. C. and Gomory, R.E. (1963). A Linear Programming Approach to the Cutting Stock Problem Part II. Operations Research, **11**, 863-888.

Gilmore, P. C. and Gomory, R.E. (1964).

Multistage Cutting Stock Problems of Two and More Dimensions. Operations Research, **13**, 94-120

Gilmore, P. C. and Gomory, R.E. (1966).

"The Theory and Computation of Knapsack Functions. Operations Research, **14**, 1045-1074

Golden, B. L. (1976). "Approaches to the Cutting Stock Problem". AIIE Transactions, **8**, 265-273.

Gregory, S. A. (1966). The Design Method. London: Butterworth.

Groover, M. P. (1980). Automation, Production Systems, and Computer Integrated Manufacturing. Englewood Cliffs, N.J.: Prentice Hall Inc.

Groover, M. P. and Zimmers, E. W. (1984).

CAD/CAM: Computer-Aided Design and Manufacturing. Englewood Cliffs, N. J. Prentice Hall International Editions.

Groppetti, R. and Semeraro, Q. (1985).

"Generative Approach to Computer Aided Process Planning." Proc. Conf.Computer and Automated Systems, Assoc. of S.M.E.

Haessler, R. W. (1971). A Heuristic Programming Solution to a Nonlinear Cutting Stock Problem. Management Science, **17**, B793-B802.

Haessler, R. W. (1974). "Controlling Cutting Pattern Changes in One-Dimensional Trim Problems." Operations Research, 23, 483-493.

Haessler, W. and Vonderembse, M. A. (1978). "A Procedure for Solving the Master Slab Cutting Stock Problem in the Steel Industry". AIIE Transactions, 11, 160-165

Harvey, R. E. (1983). "CAPP Critical to CAD/CAM Success." Iron Age, 10, 156-160.

Held, M. and Karp, R. (1970). "The Travelling Salesman Problem and Minimum Spanning Trees." Ops. Res., 18, 1138-1162.

Held, M. and Karp, R. (1971). "The Travelling Salesman Problem and Minimum Spanning Trees." Math. Prog. 1, 6-25.

Held, M., Wolfe, P. and Crowder, H. (1974). "Validation of Subgradient Optimization." Math. Prog. 6, 62-64.

Higgins, R.A. (1979). Engineering Metallurgy, Part 1. London: Hodder and Stoughton

Higgins, R. A, (1979). Materials for the Engineering Technician . London: Hodder and Stoughton.

Hinxman, A. I. (1976). "Problem Reduction and the Two-Dimensional Trim-Loss Problem with Sequencing Constraints". AISB Conf., Edinburgh, 1976, 158-165

Hinxman, A. I. (1977). "The Use of Geometric Information in Heuristic Optimisation". Unpublished PhD Thesis, Stirling Univeristy.

Hinxman, A. I. (1979). "The Trim-Loss and Assortment Problems: A Survey." European Journal of Operational Research, **5**, 8-18.

Hitomi, K. (1979). Manufacturing Systems Engineering. London: Taylor and Francis Ltd.

Husseini, M. (1989). "Computer Aided Process Planning (CAPP) for Flat Rolling of Copper Alloys." Unpublished PhD. Thesis, Aston University.

Israni, P., Sharat, S. and Sanders, J. L. (1984). "Performance Testing of Rectangular Parts-Nesting Heuristics". Conf. on Industrial Eng. Wisconsin, April 1984.

Johnson, D. (1974). "Fast Algorithms for Bin Packing." J. Comput. Systems Sci., **8**, 272-314.

Johnson, D., Demers, A., Ullman, J., Garey, M. and Graham, R. (1974) "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms." SIAM J. Comput, **3**, 297-325.

Katzan, H. (1980). Operating Systems: A Pragmatic Approach. New York: Van Nostrand Reinhold.

King, J. R. (1972). "Production Planning and Control by Computer - A Survey" Production Engineer, October 1972, 333-336.

Kendall, K. E. and Kendall, J. E. (1988). Systems Analysis and Design. Englewood Cliffs, N.J: Prentice Hall.

Kochar, A. K. (1979). Development of Computer Based Production Systems. London: Edward Arnold

Lasdon, L. (1970). Optimization Theory for Large Systems. New York: Macmillan.

Lister, A. M. (1984). Fundamentals of Operating Systems. London: Camelot Press.

Lloyd, E. (1986). An Introduction to Some Metal Forming Theory, Principles and Practice. London: Portcullis Press.

Logan, F. A. (1985). "Process Planning. The Vital Link between Design and Production". Autofact conf. proc. Detroit, November 1985

Looney, M. (1986). "Integration is the Key" Computerised Manufacturing. June 1986, 31-34

Love, D. M. and Lung, W. M and Sawyer, J. H. F. (1986). "Distributed Planning and Control in a cellular CIM System". 1st Internat. Conf. on Engineering Mangement: Theory and Application, Swansea 1986.

Love, D. M. and Barekat, M. (1988)

"An Introduction to the Decentralized, Distributive Mrp Philosophy; a Solution to Control Problems in Cellular Manufacturing Systems." Proc.3rd International Conference Computer-Aided Production Engineering, University of Michigan.

Lung, W. M. (1988). "Computer Integrated Manufacturing Control" Unpublished PhD Thesis, Aston University.

Metzger, R. W. (1958). Elementary Mathematical Programming, New York: Wiley.

Morrison, A. (1970). Storage and Stock Control for Industry and Public Undertakings. London: Camelot Press Ltd.

Morrison, D. (1968). Engineering Design. New York: McGraw Hill.

Nicholson, T. A. J. (1978). "New Developments in Production Control" Proceedings of BNF 5th Computer Conference, 27th - 28th Nov. 1978, BNF Technology Centre, Oxfordshire.

Norris, K. W. (1963). "The Morphological Approach to Design Engineering", Proceedings of Conference on Design Methods, Sept. 1962, 115-140.

Orlicky, J. (1975). Materials Requirements Planning. New York: McGraw Hill.

Orr, J.B. (1960). Roll Pass Design. London: Lund, Humphries Company Ltd.

Pahl, G. and Beitz, W. (1984). Engineering Design. Berlin: Springer-Verlag.

Paull, A. E. and Walter, J. R. (1954). "The Trim Problem: An Application of Linear Programming to the Manufacture of Newsprint." Paper presented at Annual Meeting of the Econometrica Society, Montreal, September 1954.

Pierce, J.F. (1964). "Some Large Scale Production Scheduling Problems in the Paper Industry". Englewood Cliffs, New Jersey, Prentice Hall.

Prince, G. M. (1968). "Synectics - A Method of Creative Thought" Engineering Education, March 1968, 810-811.

Reswick, J. B. (1966). Prospectus for the Engineering Design Centre Case Institute of Technology.

Sabaratnam, S. (1982). "Development of Microcomputer Based Foundry Production Control System" Unpublished Ph.D. Thesis, Aston University.

Sammet, J. E. (1969) History and Fundamentals. New Jersey: Prentice-Hall.

Shaw, R. (1984). "Process Control Systems - A Case Study of Continuous Creation", Process Measurement and Control Applications. Institute of Measurement, 5-13.

Smolik, D. P. (1983). Materials Requirements of Manufacturing. New York: Van Nostrand Reinhold.

Studel, H. J. (1982). "Computer Aided Process Planning: Past, Present and Future." Dept. of Industrial Eng., Univeristy of Wisconsin, Madison, USA.

Thomas, A. B. (1980). Stock Control In Manufacturing Industries. London: Gower Press.

Tulkoff, J. (1985). "Computer Aided Process Planning CAPP." Computer and Automated Systems,

Udeze, P. A. (1984). "Aspects of Computer Aided Process Planning in a Tube Mill" Unpublished Ph.D. Thesis, Aston University.

Wallace, T. T. (1985). MRP II. London: Oliver Wight Publications Ltd.

Wang, P. Y. (1981) "Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems".University of Maryland Mathematics Research, 81-83.

Watts, G. A. (1978). "What do we mean by Production Control?" Proceedings of B.N.F. Fifth Computer Conference, B.N.F. Technology Centre, Oxfordshire, November 1978.

Webster, N. (1961). Webster's Third New Dictionary of the English Language, London: Bell Springfield.

Wilson, D. A. L, Beaumont, G. and Yazdianpour, H. (1989).

"Solutions of a Cutting Stock Problem and their Integration with a CAPP System in a Strip Rolling Mill." Proc. Fourth Internat. Conf. on Computer-Aided Prod. Eng., Edinburgh, Nov 1988, 307-311

Woolsey, R. E. D. (1972). "A Candle to Saint Jude, or Four Real World Applications of Integer Programming." TIMS Interface, 2, 20-27.

Weill, R. Spur, G. and Eversheim, W. (1982).

"Survey of CAPP System." Annals of the CIRP, February 1982

APPENDIX A

COMPUTER OPERATING SYSTEMS

The operating system is the root of a computer's software. It is the operating system which allows a user to interact with a computer. Katzan (1980) defines an operating system as "... organised collection of programs and data that is specifically designed to manage and utilise the resources of a computer system and to facilitate the creation of computer programs and control the access to the system". The operating system therefore looks after the inputs and out puts, messeges, data storage, screen management, resource allocation, processing and various other tasks.

The operating system can be classified into 3 broad categories depending on the specific required applications as follows.

(1) Batch operating systems
(2) Time sharing operating systems
(3) Real-time operating systems

A1.1 Batch Operating Systems

The jobs are not processed in the interactive mode in batch operating systems. The users first have to prepare the jobs and submit them for processing. The operating system will then form streams of these jobs into batches of work which are processed in the best possible manner.

Normally the batch jobs are submitted as a deck of cards and have to be collected after a period of time. This has to be done even for compilation of a source program and therefore is time consuming process and is not suitable for a real time application. However, it is very useful for processing very long jobs where the users do not have to wait at a terminal. The computer time in a batch operating system is utilised in the most effective manner as oposed to other types of operating systems.

A1.2 Time Sharing Operating Systems

The basic function of a time sharing operating system is to distribute computer time to a set of terminals in such a way that it seems to each user who is logged in via those terminals, that he has a dedicated computer which is providing him a service although in actual fact the computer attends to all the users' requests in turn.

The terminals are multiplexed to the computer and each port or input/output line has to be serviced (polled) regularly without discrimination. Therefore the system may not utilise its hardware effectively. This type of a system is most suitable for short jobs.

A1.3 Real Time Operating Systems

A Real Time Operating System is essential when the processing has to be done within a rigid time limit. This processing activity is typically triggered by randomly accepted events and these events will receive data requests for service from an on line device which is not operated by a human. The operating system can in this case request data directly from the device.

This type of an operating system is most suitable for real time on-line tasks when a minimum amount of human intervention is required. The computer can be a dedicated device which will carry out these tasks continuously and specialised interfaces must be available between the computer and the process as discussed above and the computer hardware too can be large or small depending on the task.

Microprocessor based systems are mainly used in such a system in many industrial applications and these can function over long periods of time without change. The computer sometimes becomes transparent to the user where the application program can reside in an EPROM (Erasable Programmable Read Only Memory). Usage of microprocessors this way is much cheaper than employing a conventional hard wired controller.

The most essential feature in a real time operating system is the quick response time and therefore all the data has to be on-line, while all the on-line programs must be in an executable form.

The three types of operating systems described above have their advantages and disadvantages, while each of them is suitable for a different application. Since the types of operating systems and the requirements are known, morphological analysis can be used to list out these and select the most suitable combination.

A 2.4 Unix

Unix operating system was developed at Bell Laboratories and is a derivative of the old Multics Operating System. Unix at the early stages was only a single user system, but now multi user systems often use Unix. The user work is carried out by a single sequence of events and this consists of some computer memory and the files being accessed. Another process is created by a copy of the process being made and the two processes are only distinguished by the parent being able to wait for the child process to finish. A process may replace itself by another program to be executed. A unique feature of this operating system is that the priorities are assigned dynamically. A command language called 'SHELL' allows the user to interact with Unix.

A 2.5 DOS

DOS is the abbreviation for Disc Operating System and is sometimes known as MS/DOSS where MS stands for Micro Soft, who are the manufacturers. This operating system is common in small IBM compatible computers like Apricot or Amstrad and the operating system is kept on a floppy disc in the absence of hard discs. This operating system is suitable only for single user environments as the system start up procedure has to be automatic in a multi user operating system. One of the major disadvantages of DOSS is that it can only address 640K of memory.

A 2.6 VMS

VMS, or the Virtual Memory System, is quite popular on multi user VAX machines, which are manufactured by the Digital Equipment Corporation (DEC). The user has to interact with this system via a command language called DCL, which is the abbreviation for Digital Command Language.

The principle of a virtual memory system is image invocation. The disadvantage of such a system is the large number of different programs which run in quick succession and causes a severe degradation of response time. Each new image which is called for execution must be loaded to the physical memory to be executed and this will take a long time, especially when the files are scattered all over the disc. This limits the usage of menu driven programs which call other menus to run different programs. It has a range of priviledges from 1-255 and these can be used to give preference to some jobs over the others. The processing becomes even worse if priviledges are increased for some jobs, because the others will virtually come to a stand still. Therefore this system is not really suitable for real-time applications.

A 2.7 VOS

VOS is the abbreviation for Vulcan Operating System and it is found in Harris Multi User Computers. It is a multi-user, multi programming and multi-lingual time sharing operating system. It support up to 128 software priorities and they are used by the operating system to determine which program or terminal will have control of the central processing unit at one time. The high priority jobs are executed before the others and all the equal priority jobs are executed by the technique of 'time slicing' i.e. a small unit of time is allocated to each program and control is passed onto another program whenever this is used up. This goes on in a cyclic fashion, but is transparent to the user. This too uses the principle of virtual memory.

A 2.8 ELN

ELN, or External Local area Networks operating system is also found on DEC's VAX range of machines. This is one of the few real time operating systems available today. The manufacturers recommend it to be used with real-time on-line systems which run on VAX and MicroVAX processors. Unfortunately this is one of the new systems and the information available is not sufficient to have a full understanding of the system.

A 2.9 MOS

The letters MOS stand for Machine Operating System which is a very flexible operating system and works on Acorn's range of computers. The access times are very fast because the operating system is kept on a PROM and one of the major advantages is that it can be accessed via service entry points through any language or program. The disadvantage is that it is available only in a single user range of microcomputers..

A 2.10 CP/M

CP/M is the abbreviation for Control Program for Microcomputers. It is very popular today and is found on micro and mini computers. CP/M is useful for data processing a management activities. It is normally found on floppy discs on small computers.

A 2.11 P/OS

This operating system is found on DEC's 'Professional' range of computers and is appropriately named Professional Operating System. This too is found on floppy discs in the professional 325 computers and the system disc has to be inserted to start the system. It is found on a hard disc in higher versions (350) but the system initialization is very slow. This too is suitable for a single user system.

APPENDIX B

COMPUTER LANGUAGES

The computers handle data in a different way to humans. The data in computers is handled as a series of ones and zeros. These are called binary digits or bits. The instructions within a computer in this form are called machine code instructions. It is very difficult for a programmer to provide instructions to a computer using the machine code as the programming errors or bugs will be very difficult to locate.

The assembly language programming is an easier way of programming as the instructions of the assembly language, which are called mnemonics are like parts of English words. An assembler is a pre-programmed code which resides in the computers and interprets assembly language. When a programmer writes a piece of code in assembly language it is interpreted or assembled by the resident assembler program into a form of ones and zeros which can then be executed by the computer. This type of programming is called low level programming.

The usage of a language to program a computer is called high level programming and this technique is much easier than any other due to the number of instructions which are available to the programmer. Most of these instructions are simple English words and therefore error detection is easy. These programs will first have to be converted to a set of machine code instructions before being executed and this process if very similar to the usage of the assembler.

There are two ways of interpreting the instructions, the first being interpreting each line of instructions just before it is executed, which takes relatively longer execution times and the second is to interpret the source program completely to form a set of executable machine code instructions. Obviously the execution time of the latter is quicker because the instructions do not have to be interpreted as they are executed. Such a case is referred to as a compilation and the interpreter is called a language compiler. Any errors in the source program will halt the compiler in this case, while such errors will be detected only when the corresponding line of instructions have been reached for execution in the former case. A language, therefore, is an accepted set of instructions which have to be translated to machine code by using a previously written program.

Morphological analysis can once again be employed to look at the available languages and select the most suitable by comparing each language with the combination of requirements.

B.1. Ada

Ada is a relatively new language which was developed in the 1970s by the American Department of Defence in order to replace several different languages which were being used. It was intended for the use in a large and diverse range of applications, but the language itself turned out to be diverse.

It has all the features of variable handling such as arrays, integers and strings while being capable of making calls to sub programs. The biggest disadvantage of the language is its unpopularity and it can be said as 'unknown' with respect to the engineering industry, although it is available on VAX machines.

B.2 APL

APL which is the abbreviation for A Programming Language was implemented at IBMs Watson Research Centre on an IBM 360/50. APL is normally used as a time sharing system and it interacts with the user in a conversational style, which allows the programmer to use it dynamically.

A useful feature in APL is that there is no concept of writing a main program. The subprograms can be called either from another subprogram or by the user via this terminal. The control, therefore can be passed back and forth between the user and the program. Like Ada, APL too is not known to be used in the engineering industry to a great extent.

B.3 Algol

Algol was developed in the late 1950s and it has three versions to date namely; Algol 60, Algol 68, and S-Algol. An essential feature of Algol is that it is not a card or a line orientated language i.e. one statement may occupy more than one card and more than one line may appear on one card. Algol is not a very popular language today.

B.4 Basic

Basic is by far the most popular language in microcomputers but not to that extent in larger machines. It too has the essential features like different types of loops, conditional statements, subroutines, transfer of controlled statements and a host of other utilities. When used in a micro environment, it cannot only read the contents of a memory location, but also write to any location directly, although this is impossible in a multi user system because of the system protection procedures.

Basic allows the user to mix all variables other than character strings, which are indicated by a '$' at the end of the variable name. The lengths of these however do not have to be declared and therefore the complete string is saved regardless of its length. All variables which are used in the main program and the subprograms are regarded as common unless declared as local in any order. This eliminates the problem of common statements.

Basic can be found either as a compiler or an interpreter. The biggest disadvantage of Basic is the difficulty in transportation between different machines because of the number of versions available. Basic is regarded as a poor language in some circles and the main argument against it is that it is lacking suitability in developing programs in a systematic manner and that it tends to produce programs with no or little structure.

B.5 Cobol

Cobol or Common Business Oriented Language has been used since 1960s for business applications. It is widely used for business data processing, payroll and accounts while it gives a high priority for input/output files which are most important in buisness. Another important characteristic is that it contains syntax which are closely related to English words. This feature makes Cobol self documented. Cobol has not been known to be used for scientific or engineering purposes.

B.6 Forth

This is a strong contender to replace Basic for the simple reason that the programs which are written in Forth occupy less memory and run faster than the equivalent programs in Basic. The disadvantage of Forth is its different style of programming when compared with other conventional languages. This and the lack of documentation makes Forth a less used language.

B.7 Fortran

Fortran which is known as a scientific language was developed in the late 1950s by IBM workers and it stands for FORmula TRANslation. Development of Fortran was the biggest step in the development of computers as the majority of programming was done using assembly language or machine code before Fortran. In describing the history of this language, Sammet (1969) states:

The earliest significant document that seems to exist is one marked "PRELIMINARY REPORT, Specifications for the IBM Mathematical FORmula TRANslating System, FORTRAN," dated November 10, 1954, and issued by the Programming Research Group, Applied Science Division, of IBM. The first sentence of this report states, "The IBM Mathematical Formula Translating System, or briefly, FORTRAN, will comprise a large set of programs to enable the IBM 704 to accept a concise formulation of a problem in terms of a mathematical notation and to produce automatically a high-speed 704 program for the solution of the problem." It is interesting to note that the authors (who are not identified in the document) felt a need to justify such a development. They devoted several pages to a discussion of the advantages of such a system. They cited primarily the virtual elimination of coding and debugging, reduction in elapsed time, doubling of machine output, and the feasibility of investigating mathematical models.

FORTRAN is considered to be rather easy to learn. It is rare that a course in this language runs for more than 1 or 2 weeks, and many technically oriented presons have learned enough in 1 day to continue on their own.

B.8 Languages for Formal Algebraic Manipulation

Languages designed for formal algebraic manipulation are used for the computer processing of formal mathematical expressions without concern for their actual numerical values. The types of operation involved include differentiation, integration, and substitution of algebraic expressions. Below are some of the more popular packages.

B.9 FORMAC

FORMAC (FORmula MAnipulation Compiler), an extension of FORTRAN, is designed for the manipulation of mathematical expressions. Its basic concepts were developed in July 1962 by Jean E. Sammet and Robert G. Tobey, working at IBM's Boston-based Advanced Programming Department. At the time it was recognized that there was a need for a formal algebraic capability associated with a language already in existence, and FORTRAN was the logical choice. The first draft of the specifications for FORMAC was completed in December 1962, and the design implementation was begun shortly afterward. The basic goal of this work was to develop a formal mathematical manipulation system for the IBM 7090/94. Originally FORMAC was intended to be merely experimental and available only within the IBM Corporation. By April 1964, after thorough testing, the FORMAC was released to a number of IBM customers. Even though it was considered to be experimental and lacked the "official" status of other compilers, FORMAC was used extensively.

In the opinion of the developer, the most significant contribution of FORMAC is that it introduced the concept of adding a formal mathematical manipulation capability for solving numerical scientific problems to a language already in existence.

B.10 MATLAB

Engleman (1964) began developing MATLAB, an on-line system for formal algebraic manipulation problems. MATLAB has since undergone further development. The original version was replaced by MATLAB 68, a system that became operational in the autumn of 1967. MATLAB was the first language to include high-level operations such as integration, and its most significant contribution is considered to be the development of routines that could complete these operations. MATLAB is a valuable tool to the scientist with access to an on-line console that incorporates the system. Consider that within seconds the solution to a complex set of mathermatical expressions can be obtained merely by sitting at the typewriter and feeding the information into the computer. Equations that could take days to solve can thus be reduced to meaningful form with minimal human effort.

B.11 GPSS

The General Purpose Simulation System (GPSS) was one of the earliest simulation languages developed and was first introduced to the public in 1961. It is an IBM computer program for conducting simulated evaluations of any type of discrete system. In contrast to most simulation languages, GPSS programs are based on a block diagram approac. In brief, the user draws a flow chart of the system to be represented. Each block in the flow chart represents a process in the system, and changes in the system are caused by a series of transactions that flow through the blocks at specified time intervals. The connections between the blocks indicate the primary and alternative sequences of processes, with the conditions of the system at the time determining which path is chosen.

GPSS is relatively easy to learn. Its simplicity derives from the fact that the creation of the transactions, events, and activities, as well as the output, are almost completely programmed for the user. In addition, the criteria for the sequence of events and the conditions of the system are imbedded in the block diagram itself, and little extra programming effort is required.

B.12 SIMULA

SIMULA is a process-oriented discrete simulation language that is used more widely in Europe than in the United States. Like SIMSCRIPT it was conceived as an extension of a general purpose language, in this case ALGOL. In SIMULA, the system is viewed as sets of processes, and simulation is accomplished by program blocks that effect these processes. It is felt that this language, particularly in its implementation as SIMULA 67, is an elegant and powerful discrete simulation language. However, because there is little interest in or availability of ALGOL and ALGOL-based languages in the United States, the use of SIMULA almost certainly will continue to be inhibited.

B.13 MPS

IBM's Mathematical Programming System (MPS) is designed to solve systems of linear or separable functions using the accepted techniques of linear or separable programming. Linear programming is a technique used to solve a set of linear inequalities. Separable programming is used in solving certain types of nonlinear function within the framework of the general linear programming procedure. Problems of these types often occur in such areas as machine loading, materials allocation, ingredient blending, distribution and

In the opinion of the developer, the most significant contribution of FORMAC is that it introduced the concept of adding a formal mathematical manipulation capability for solving numerical scientific problems to a language already in existence.

B.10 MATLAB

Engleman (1964) began developing MATLAB, an on-line system for formal algebraic manipulation problems. MATLAB has since undergone further development. The original version was replaced by MATLAB 68, a system that became operational in the autumn of 1967. MATLAB was the first language to include high-level operations such as integration, and its most significant contribution is considered to be the development of routines that could complete these operations. MATLAB is a valuable tool to the scientist with access to an on-line console that incorporates the system. Consider that within seconds the solution to a complex set of mathermatical expressions can be obtained merely by sitting at the typewriter and feeding the information into the computer. Equations that could take days to solve can thus be reduced to meaningful form with minimal human effort.

B.11 GPSS

The General Purpose Simulation System (GPSS) was one of the earliest simulation languages developed and was first introduced to the public in 1961. It is an IBM computer program for conducting simulated evaluations of any type of discrete system. In contrast to most simulation languages, GPSS programs are based on a block diagram approac. In brief, the user draws a flow chart of the system to be represented. Each block in the flow chart represents a process in the system, and changes in the system are caused by a series of transactions that flow through the blocks at specified time intervals. The connections between the blocks indicate the primary and alternative sequences of processes, with the conditions of the system at the time determining which path is chosen.

GPSS is relatively easy to learn. Its simplicity derives from the fact that the creation of the transactions, events, and activities, as well as the output, are almost completely programmed for the user. In addition, the criteria for the sequence of events and the conditions of the system are imbedded in the block diagram itself, and little extra programming effort is required.

B.12 SIMULA

SIMULA is a process-oriented discrete simulation language that is used more widely in Europe than in the United States. Like SIMSCRIPT it was conceived as an extension of a general purpose language, in this case ALGOL. In SIMULA, the system is viewed as sets of processes, and simulation is accomplished by program blocks that effect these processes. It is felt that this language, particularly in its implementation as SIMULA 67, is an elegant and powerful discrete simulation language. However, because there is little interest in or availability of ALGOL and ALGOL-based languages in the United States, the use of SIMULA almost certainly will continue to be inhibited.

B.13 MPS

IBM's Mathematical Programming System (MPS) is designed to solve systems of linear or separable functions using the accepted techniques of linear or separable programming. Linear programming is a technique used to solve a set of linear inequalities. Separable programming is used in solving certain types of nonlinear function within the framework of the general linear programming procedure. Problems of these types often occur in such areas as machine loading, materials allocation, ingredient blending, distribution and shipping scheduling, labour allocation and so on. To use MPS, the analyst first builds a model of the problem in terms of the equalities or inequalities that express the objective of the system and the constraints that have been imposed on it, and then uses the computer to find an optimal solution. The effect on the optimal solution of changing key elements can then be found, or alternative answers computed by systematically varying cost or requirement data. The final solution can be prepared by the computer in the form of a management report using MPS's report generator facility.

The design of MPS has been refined and improved by many years of implementation. The applicaiton of linear programming techniques accounts for a large portion of the computer time used by industrial organizations, and although this and similar languages are rather limited in the scope of the problems they can solve, their importance to the efficient operation of many large business firms cannot be overemphasized. Problems incoluving 100 variables and 500 inequalities (or constraints) are not uncommon. The solution of these problems "by hand" could take so long that the answer would be almost useless by the time it was obtained.

B.14 MATLAN

IBM's MATrix LANguage (MATLAN) is a general purpose computational system for any application that is expressible in matrix notation. It is a problem-oriented language incorporating many functional statements designed to perform such operations as matrix gnereation, matrix manipulation, and matrix algebra. The language can handle data in the form of scalars, matrices or arrays and can operate on either real or complex mathematical values.

MATLAN has been found to be a useful tool in solving mathematical problems in such areas as structural analysis, network analysis, statistics, and econometrics and in such fields as aeronautical, civil and electrical engineering. Specific problems to which it has been successfully addressed include systems of linear equations, ordinary and partial differential equations, integral equations and Boolean matrix algebra.

APPENDIX C

FUNCTIONAL SPECIFICATION

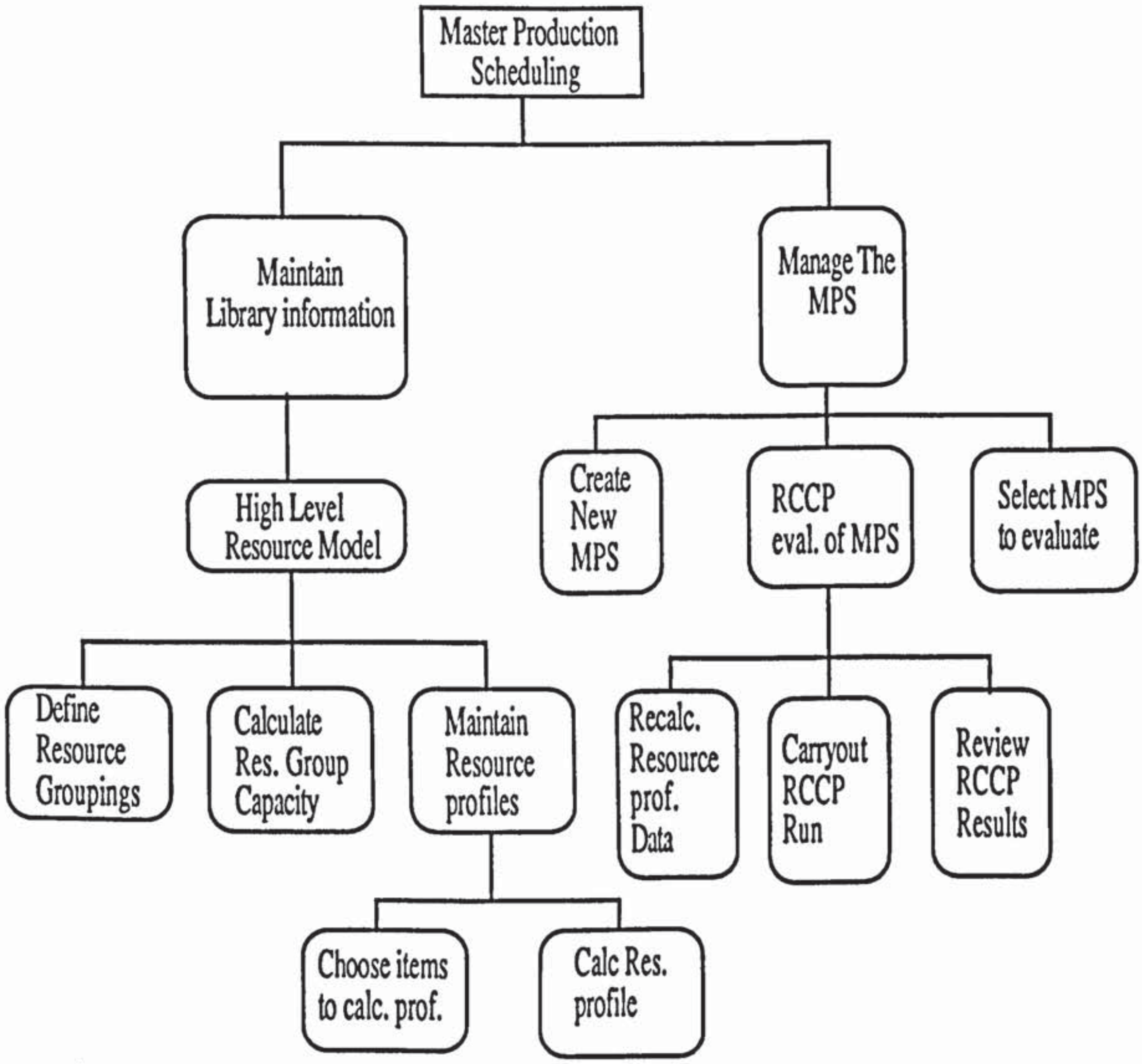


Figure C.1 The Functions of The MPS Module.

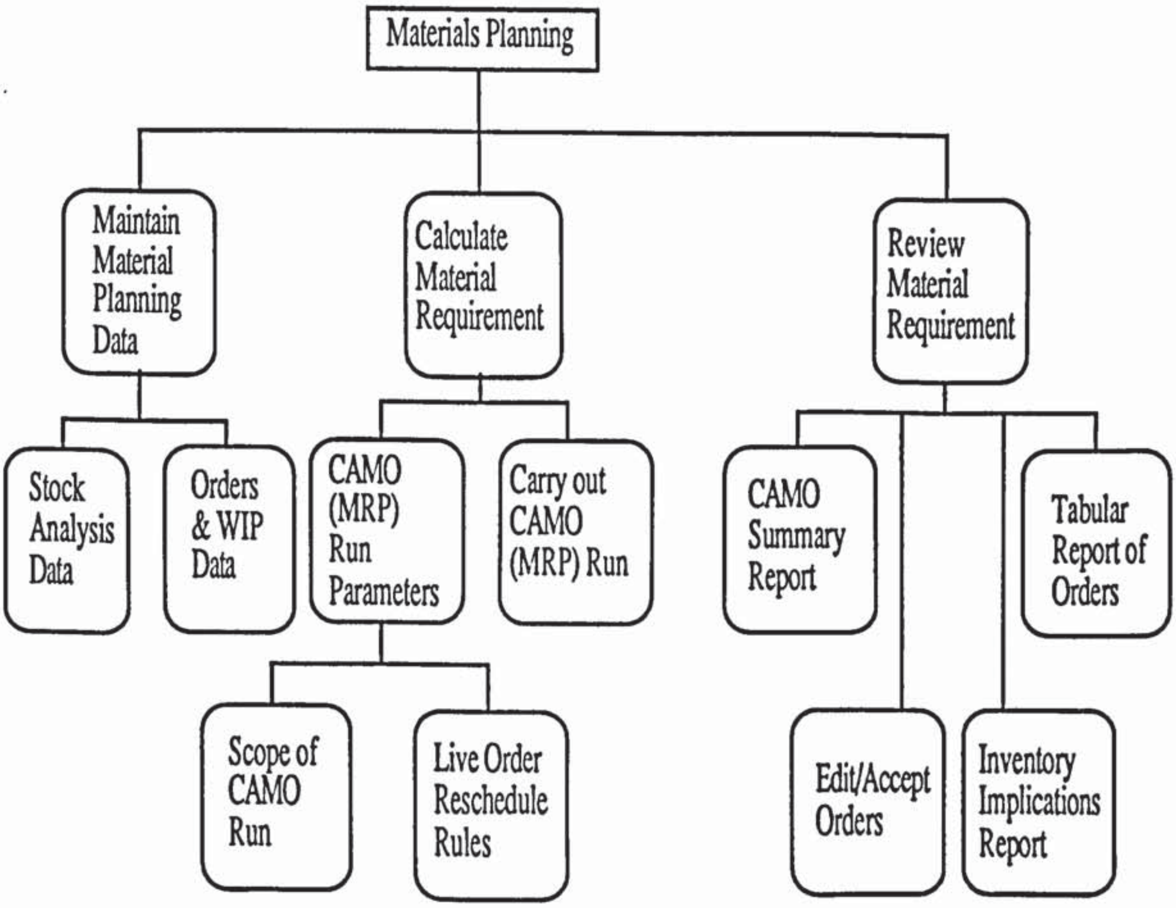


Figure C.2 The Material Requirements Planning Functions.

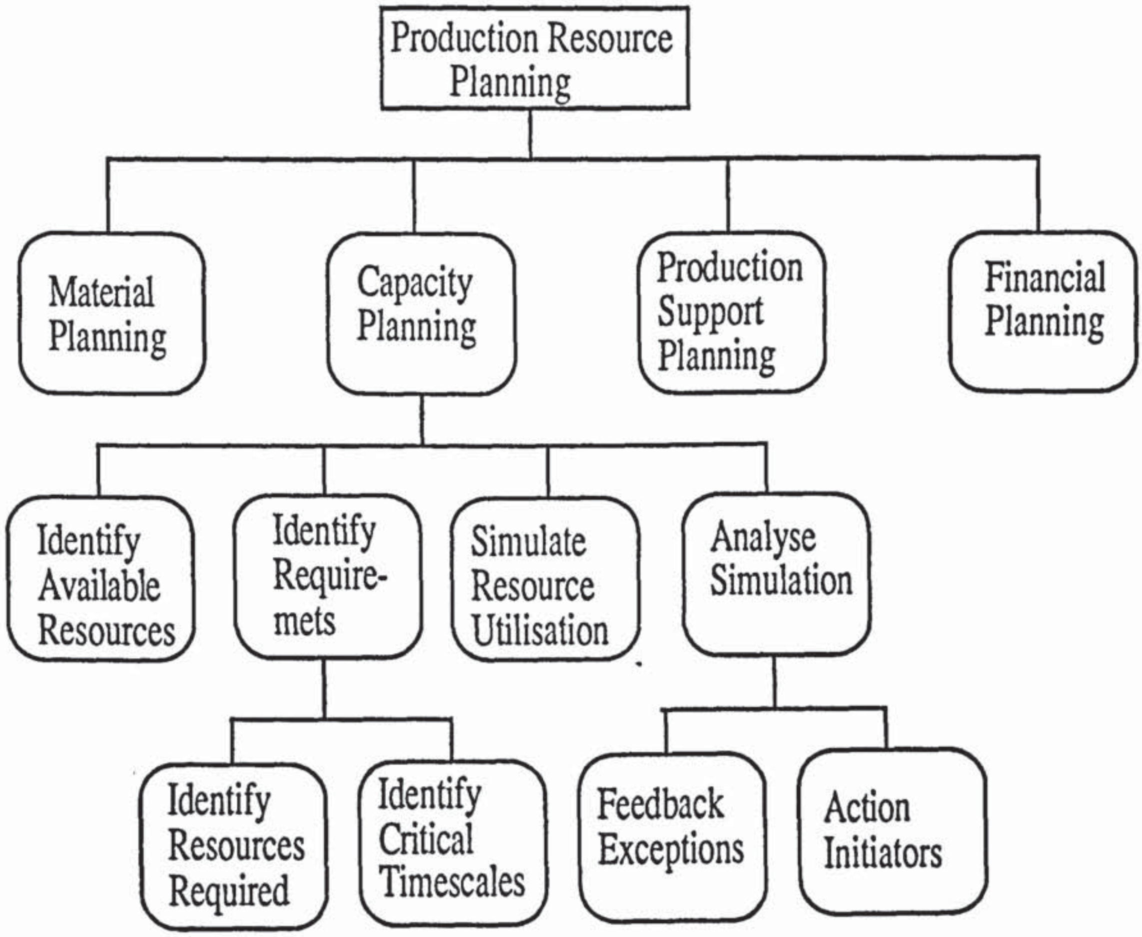


Figure C.3 The Production Resource Planning Functions.