Design, Implementation and Control of a Small Industrial Manufacturing System

by

Rajendran Narayanan

as partial requirement for the degree of

Master of Philosophy

at

The Department of
Production Technology and Production Management,
The University of Aston in Birmingham

on

July 1982

Declaration

I declare that no part of the work described in this thesis was done in collaboration, and that this work has not been submitted for any other academic award in this or any other institution.

The University of Aston in Birmingham

Design, Implementation and Control of a Small Industrial Manufacturing System

Rajendran Narayanan Master of Philosophy, 1982

Summary

One part of manufacturing systems that contributes to the well being of British Industry is that of small intermittent production systems, commonly known as 'Job Shops'. Job shops tend to be run by management on a break-even basis and management do not always fully realise the potential capabilities of their production system.

The project undertaken was directly linked to a job shop where a general analysis of the factory was being carried out to improve productivity as well as to set out a means by which information could be made more readily available to management.

The project involved an overall increase in efficiency of the plant (machinery being semi-automatic) by basic analysis and application of computer orientated mathematical techniques. A series of computer programs, together with graphical output, enabled a greater appreciation of the mathematical techniques employed. Some of these programs were later compiled into two general purpose, master packages that could be modified easily to suit any job shop within reason. One of the packages, the cam design package, enables the complete setting out of all resources for the manufacture of a batch of components. Information pertaining to the production of the batch is also available and very useful as the package could be run interactively and results obtained immediately. The other package, the scheduling package, enables orders to be added into a master schedule file, the criteria of schedule being pre-assigned priorities generated from the specifications of the order, and subject to alterations at the discretion of management.

Both packages are executed using macros, written in the Geroge 3 Operating System language, and can be run on any ICL 1900 series installation. The results and conclusions drawn from the project were discussed with the Managing Director and a feasibility study on the implementation of a digital computer was proposed.

Keywords : Job Shop, Computer Aided Design, Scheduling, Management Information.

Acknowledgements

I would like to thank my supervisor, Dr. D.A. Milner for his extreme patience and valuable guidance throughout the entire project.

LIST OF FIGURES Page

End of List

LIST OF TABLES Page

End of List

Chapter 1 : Introduction

A small job shop, situated in the Industrial part of Birmingham, was heavily involved with improving their 'productivity' and 'system efficiency', sometime in March 1977. For reasons of confidentiality, this job shop will henceforth be referred to as X Co. The management of X Co. discovered that due to their policies on resource allocation and planning in the production control section, time wastages and low efficiency figures prevailed.

The managing director of X Co. had direct links with the University's Department of Production Engineering via the Reader of the department, Dr. D A Milner. It was hence suggested that research be carried out in direct link with this job shop, not only as a possible aid to X Co., but also for the opportunity to link academic research work to real world problems that exist in all facets of Industry. Due to this link with X Co., some of the work to be done would be unique to X Co., but a general view was maintained so that any knowledge obtained, as a result, could be applied to Industry in general.

The main basis of the research was set out on the following 3 objectives -

1) Increase the overall efficiency of the job shop.
2) Eliminate or reduce all time wastages if possible.
3) Obtain general information for management.

The methods to be employed to achieve these objectives would cover many aspects of systems analysis and techniques in operations research. Mathematical models would have to be built and extensive use of a digital computer would be necessary to handle the vast quantity of data and extensive manipulation of equations.

On further discussion with the managing director of X Co., and after an initial study, the following were suggested as specific areas of research.

1) A comprehensive study of multi-stage machining process optimization which includes 'Dynamic Programming' (Selection of optimum elements at each decision stage). This will help solve resource allocation problems for a particular batch that has to be produced repeatedly.

2) A simulation of the labour-force to predict the necessary optimum work-force required to minimize production loss (Inclusive of Absenteeism). Aids elimination of time wastages due to non-attendence.

3) A study of economic machining conditions to select cutting feeds and speeds for the machines. The conventional criteria to be investigated being

   a) Maximum production rate
   b) Mininum cost
   c) Maximum profit

This would give an insight to production planning so that the job shop would leave the 'break-even' policy and start maximizing profit.

4) A technique for designing suitable cams for the control of semi-automatic machines. Any improvements over the present manual methods would greatly increase productivity and efficiency.

5) A scheduling technique where jobs encountered could be included progressively into the production system and be scheduled according to priority ratings (Ratings at the discretion of management). This will give management much of the information they require in order to make quick, accurate decisions.

To carry out the above tasks, computer programmes would have to be written in a general form, so that only slight modications need be made when considering different job shops. Also utility programs will have to be written so that the information required by management would be in a more readily acceptable form, i.e in the form of tables or graphs.

Chapter 2 : Analysis of the Factory

The project undertaken was directly linked to a small job shop located in the industrial part of the city of Birmingham. This particular job shop produces small or light components of varying shapes, sizes, design and purposes for sale, usually to other Engineering Industries.

This chapter deals with the breakdown of the job shop into its component departments. Brief descriptions and explanations of its workings under the respective departments enable the reader to appreciate the type of job shop involved in the project.

2.1 : The Model - A Systems Approach

The systems approach in analysing any problem would be by logical restructuring of the main problem at hand. By restructuring we split the main problem down into smaller sub-problems. Once this has been done the smaller sub-problems can be solved without great difficulty. If there is still more difficulty in solving the sub-problems we can further split the sub-problems into even smaller problems. The smaller the sub-problem the easier the analysis and solution will be.

In the case of the job shop, the complete production system can be broken down into separate and independent systems. The main systems that will be considered in this project are -

1) The Communication system
2) The Personnel system
3) The Sales system
4) The Finance system
5) The Material system

These sub-systems do fully describe the complete job shop and are discussed in the rest of this chapter.

2.2 : The Communication System

Every manufacturing system must have a comprehensive communication system for high efficiency and maximum utilization of the whole system. If the communication system is inadequate all other systems will fail to function as they should i.e with maximum efficiency.

The communication system is basically concerned with the flow of information between all systems, be it major, minor, basic or control. This involves all possible information, verbal or written, from the time a prospective customer makes an enquiry, through the stages of placement of orders to the final stage of delivery and satisfaction of the customer.

To briefly describe the communication system, it is necessary to outline the various departments and main sections involved in the factory. Initially it was thought that Figure 2.1 below represented the skeleton of the manufacturing system.

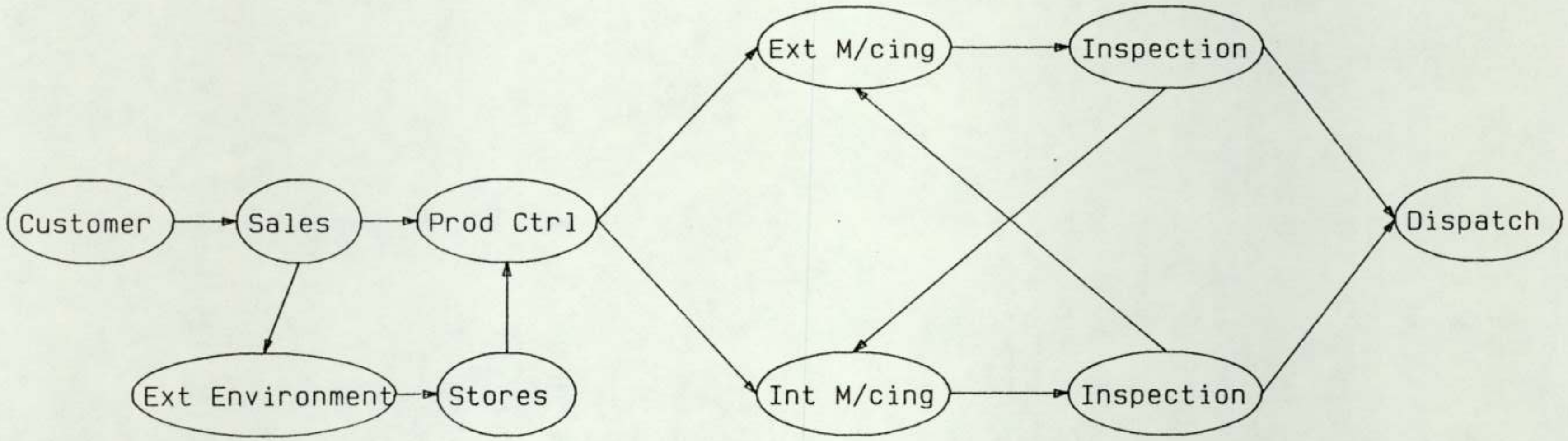


Figure 2.1 : Initial Factory Information Flow

This did not clarify the interactions of the various sections and departments. A more comprehensive analysis led to the development of the initial representation (Figure 2.1) to a final and fully detailed representation as in Figure 2.2. The important departments and sections are listed below :

1) Customer
2) Sales Department
3) Production Control Department
4) Inspection Department
5) Machine Shop
6) Hold Bay
7) Stores
8) Dispatch
9) Commercial Department
10) External Suppliers
11) Transport Department

The actual interactions between the above mentioned departments and sections are best tabulated. Tables have been set out to provide information on the interactions between the above mentioned departments.

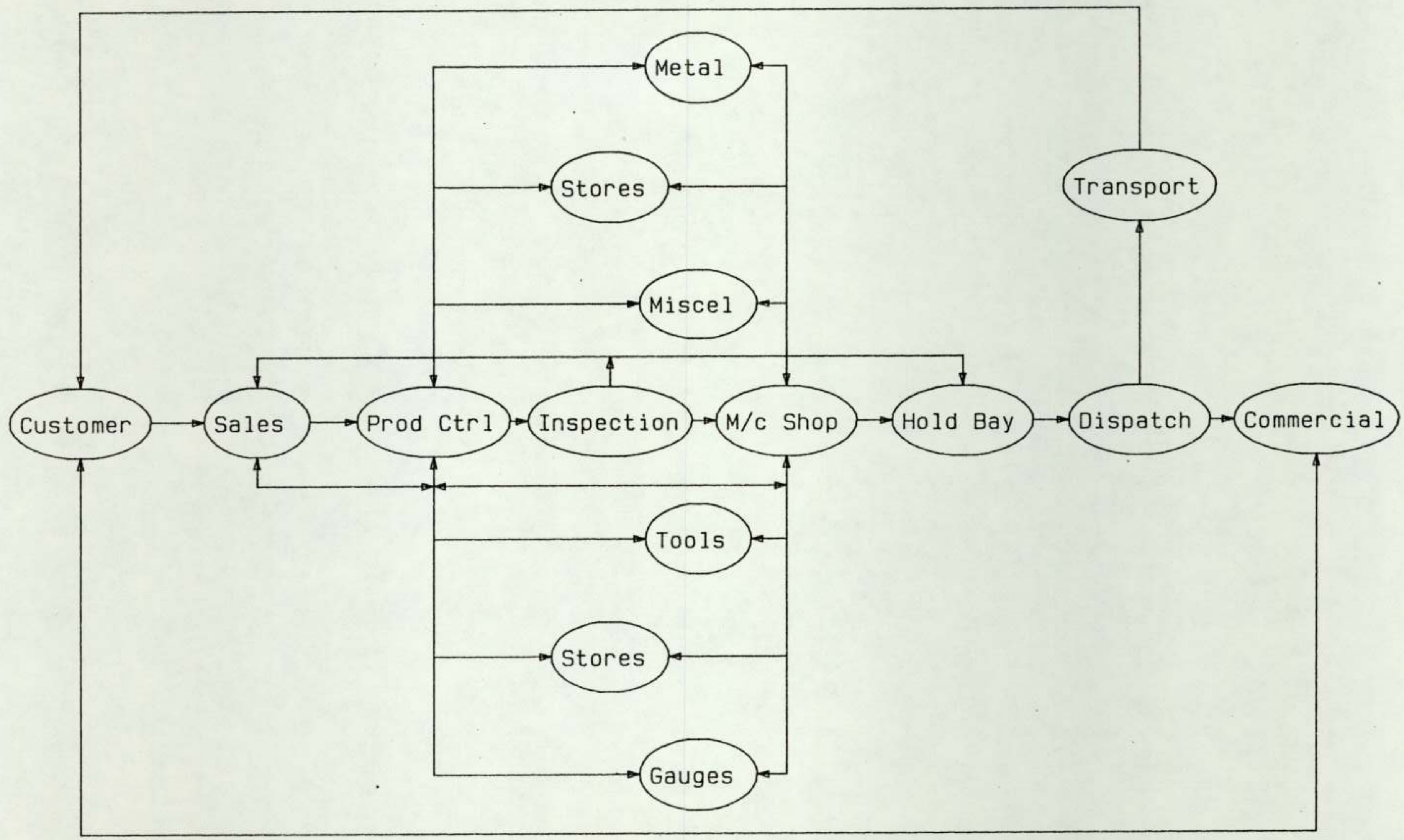


Figure 2.2 : Finalized Factory Information flow

Table 2.1 : Information flow of the Inspection Department

A) Machine Shop	1) Issue of sales order 2) First off inspection 3) Approval of first off sample 4) Patrol inspection for in-process quality 5) Defective analysis 6) Issue of gauge 7) Approval of in-process quality
B) Hold Bay	1) Final inspection 2) Final inspection report
C) Metal Supplier	1) Material advice notes 2) Quality control
D) Tool Supplier	1) Tool advice notes 2) Quality control
E) Gauge Supplier	1) Gauges advice notes 2) Quality control

Table 2.2 : Information flow of the Commercial Department

A) Dispatch	1) Advice on goods for dispatch
B) Customer	1) Advices, invoices and statements 2) Payments

Table 2.3 : Information flow of the Machine Shop.

A) Hold Bay	1) Completed Machine components 2) Defective components returned (rework)
B) Metal Supplier	1) Supply of metal
C) Tool Supplier	1) Supply of tools
D) Gauge Supplier	1) Supply of gauges
E) Misc. Supplier	1) Supply of miscellaneous items
F) Stores	1) Supply of stocked items

Table 2.4 : Information flow of the Sales Department

A) Customer	1) Enquiry as to orders 2) Quotations on enquiry 3) Placement of orders 4) Compromise of acceptable price level 5) Schedules and due dates 6) Approval of first off inspection sample 7) Special concessions - negotiable
B) Prod. Control	1) Issue of sales order 2) Advice on efficiency and machine availability 3) Issue of cam design 4) Request for batch delivery
C) Inspection	1) Request first off sample approval 2) Approval from customer via sales 3) Report on defective batch (D.B.A.R.)
D) Metal Supplier	1) Enquiry 2) Price and delivery
E) Tool Supplier	1) Enquiry 2) Price and delivery
F) Gauge Supplier	1) Enquiry 2) Price and delivery
G) Misc. Supplier	1) Enquiry 2) Price and delivery

Table 2.5 : Information flow of the Stores

A) Metal Supplier	1) Supply to store
B) Tool Supplier	1) Supply to store
C) Gauge Supplier	1) Supply to store
D) Misc. Supply	1) Supply to store

Table 2.6 : Information flow of the Production Control Department

A) Stores	1) Check all stock request 2) Obtain and inform on stock figures
B) Metal Supplier	1) Placement of order 2) Schedule to maintain minimum inventory
C) Tool Supplier	1) Placement of order 2) Schedule to maintain minimum inventory
D) Gauge Supplier	1) Placement of order 2) Schedule to maintain minimum inventory
E) Misc. Supplier	1) Placement of order 2) Schedule to maintain minimum inventory
F) Inspection	1) Issue of sales order 2) Advice notes on sales 3) Approval for dispatch of batch
G) Machine Shop	1) Issue of cam design 2) Scheduling or shop loading

Table 2.7 : Information flow of the Dispatch Department

A) Transport	1) Transport availabilities 2) Schedules on transport
B) Hold Bay	1) Machined components for dispatch
C) Stores	1) Packing material request
D) Customer	1) Delivery of batch

The passage of information is via written forms and sheets. All necessary information and data at different stages of the production system are written or tabulated in sheets and passed from one section to another, usually with more relevent information and data added to it. Examples of such paperwork are :

1) Drawing Sheets
2) Cam design charts
3) Material Order Sheets
4) Tool Order Sheets
5) Sales Order Sheets
6) First Off Inspection Reports
7) Defective Batch Approval Reports
8) Finished Inspection Reports

2.3 : The Personnel System

The personnel system can be considered as that part of the manufacturing system where human factors are involved. As this manufacturing system is not one of total and complete automation human factors are necessary to make possible the existance of the manufacturing system. The personnel system or elements of the personnel system is present and actively participating in all the other systems and sub-systems that make up the factory. The personnel system hence constitutes all human interactions within the factory. From Figure 2.2 one could approximately determine the necessary labour and staff that would constitute the personnel system. At the start of the project, the labour force was segregated into 5 different categories. They are :

1) Production Control Staff
2) Tool Setters
3) Bar Feeders
4) Machine Operators
5) General Workers

The managing director is directly responsible for the employment and deployment of personnel. The system and organization is such that most of the workers are of type 'temperory' or 'part-time'.

For details of the specifiction of the various categories see following tables below.

Table 2.8 : Job capabilities of employee categories.

	Manage & Inspect	Set Tools	Feed Bars	1st M/c	2nd M/c
Prod.Controller	*				
Tool Setters		*	*	*	*
Bar Feeder			*	*	*
Machine Operator			*	*	*
General Worker					*

Table 2.9 : Basic information on employee categories.

	Employement Type	Wages (coded)	No Off
Prod.Controller	Full Time	N.A	2
Tool Setters	Full Time	175	3
Bar Feeder	Part Time	110	3
Machine Operator	Part Time	100	3
General Worker	Part Time	80	6

The production rate works on the basis that there are 12 months in a working year, 20 days in a calendar month and 8 hours in a full working day. It must also be mentioned that a bonus scheme exists but will be explained at a later stage.

Note : The names of employees and the hours worked from period Dec 75 to Dec 76 are given in Appendix I

2.4 : The Sales System

The sales system is concerned with the added interactions of external bodies or systems for the maintenance of organization within the factory. Its major involvments are with customers and external suppliers. All necessary requirements of both material and information not obtainable within the factory are initiated by the sales department. A simple schematic diagram depicting its associations is shown below in Figure 2.3.

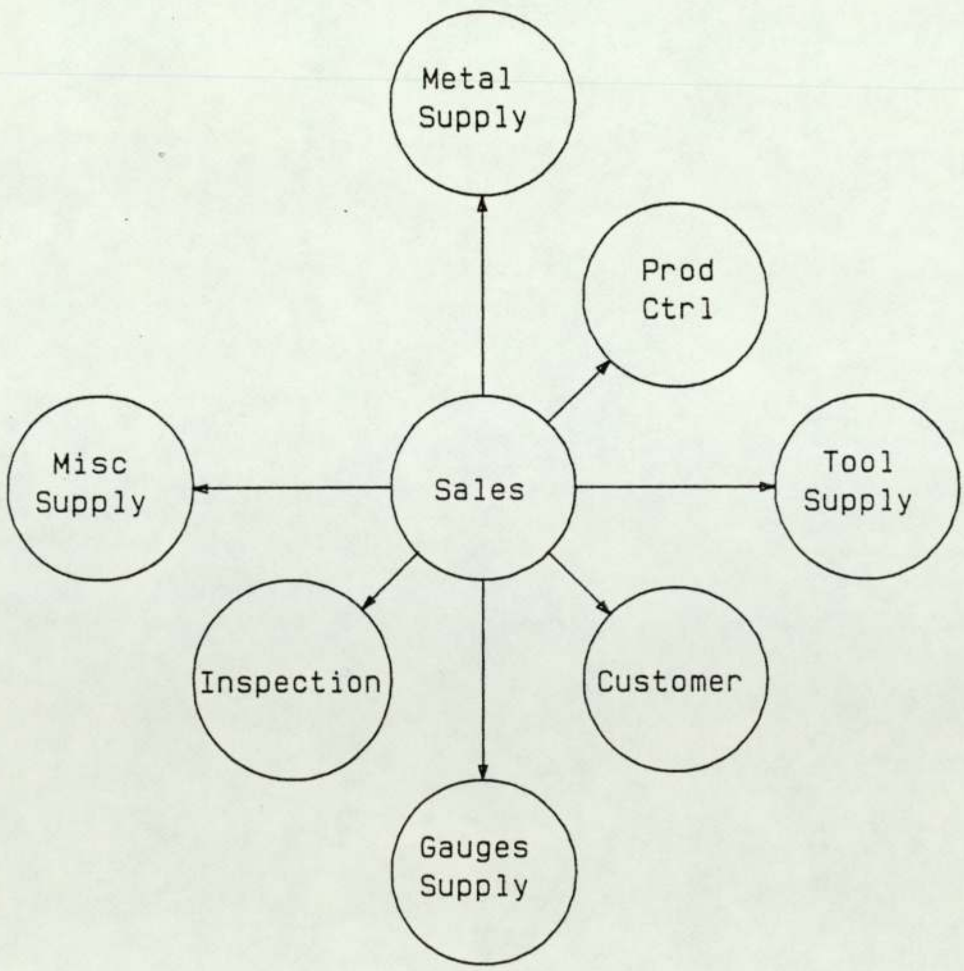


Figure 2.3 : Interactions of Sales Department

2.4.1 : Interactive process of the Sales Department

Customers make enquiries to the sales department pertaining to the manufacture of certain components. This factory survives on a symbiotic basis. It recieves orders from customers, fulfils the orders to the letter, and in return obtains more business directly or indirectly through these customers. Advertisments do not play an important role in obtaining customers. The only way the factory maintains a steady market is through production of good quality components which should be delivered according to schedule.

Upon reciept of an order, a Sales Order Sheet (S.O.S.) will be raised. Details included in the S.O.S. will be gauging and inspection instructions together with production requirements. The S.O.S. will then be routed to the other departments. At the time the S.O.S. is being routed through the departments, information on plant efficiency and machine availability and stock level figures will be obtained from production control. This information enables the product engineering section of the sales department to design a cam for the operation of the selected semi-automatic machines. The cam design for a particular machine will produce a component from a bar stock to the specifications of the customer. To prevent disruption of flow of work in progress machine selection is vital and hence the communications with production control.

On issue of the cam design chart to production control, enquiries are made to suppliers of metal, tool, gauges and miscellaneous items. A price and delivery date will be negotiated by the sales department after which production control will again be consulted to make out the order.

At a later stage in manufacture, when the first-off inspection sample is ready for customer approval, the sales department will be notified, as it is the closest link to the customer. If the sample is acceptable, the inspection department will be notified and full production will commence. If for any reason the sample is rejected, rectification may take place and concessions might be offered. In the worse case, the machines will be reset with newly designed cams and a new 'first-off' sample is prepared.

The functions of the sales department does not end here. It maintains contact with external bodies, as mentioned earlier, right throughout the reciept and fulfilment of orders, from one order to another.

2.5 : The Finance System

The finance system is concerned with the flow of funds from the point of reciept from the customers or lending agency until they are distributed to suppliers, employees and other claimants.

In this factory the department responsible for finances is the commercial department. All paperwork pertaining to transactions within the factory and also with external bodies are retained and recorded for future reference.

The main internal flow of funds are for payment to employees i.e wages and bonus schemes. Time keeping in terms of weekly labour hours must be reported in order that wages may be calculated. A tabulation on the wages index was given in the previous sub-section of personnel system in Table 2.8. Also typical weekly hours for individual employee for the period Dec 75 to Dec 76 is given in Appendix I.

2.6 : The Material System

The material system can be considered as the flow of material from the time the raw materials or purchased parts are shipped from the supplier through receiving, storage, fabrication, and assembly until the finished product is delivered to the customer.

For this factory it would be best to have the machine shop as a central department, as this section is directly responsible for the transformation of materials from one stage to another. The schematic diagram Figure 2.4 will help visualize the interactions of the factory's departments and sub-systems.

For a clearer understanding of the material system a diagram of the entire plant, showing the actual flow of material from the stores to the stations where they are required, will be necessary. This has been done as in Figure 2.5.

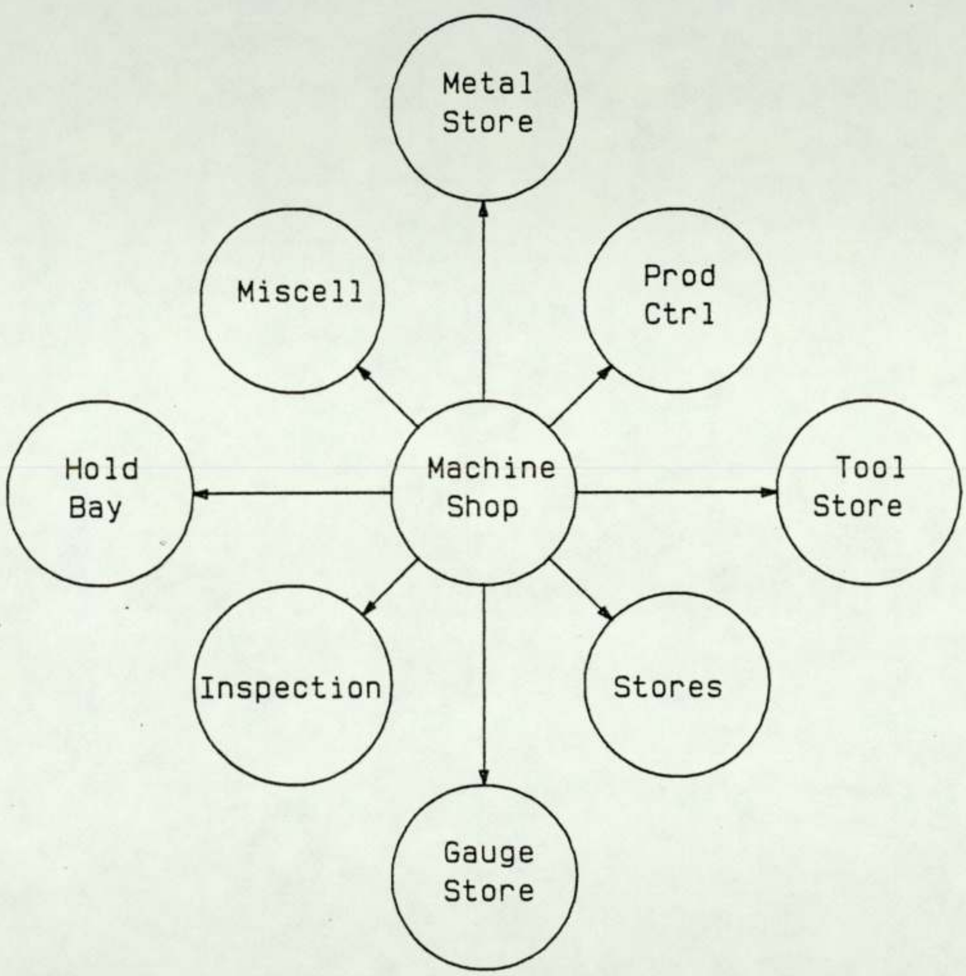


Figure 2.4 : Interactions of the Machine Shop

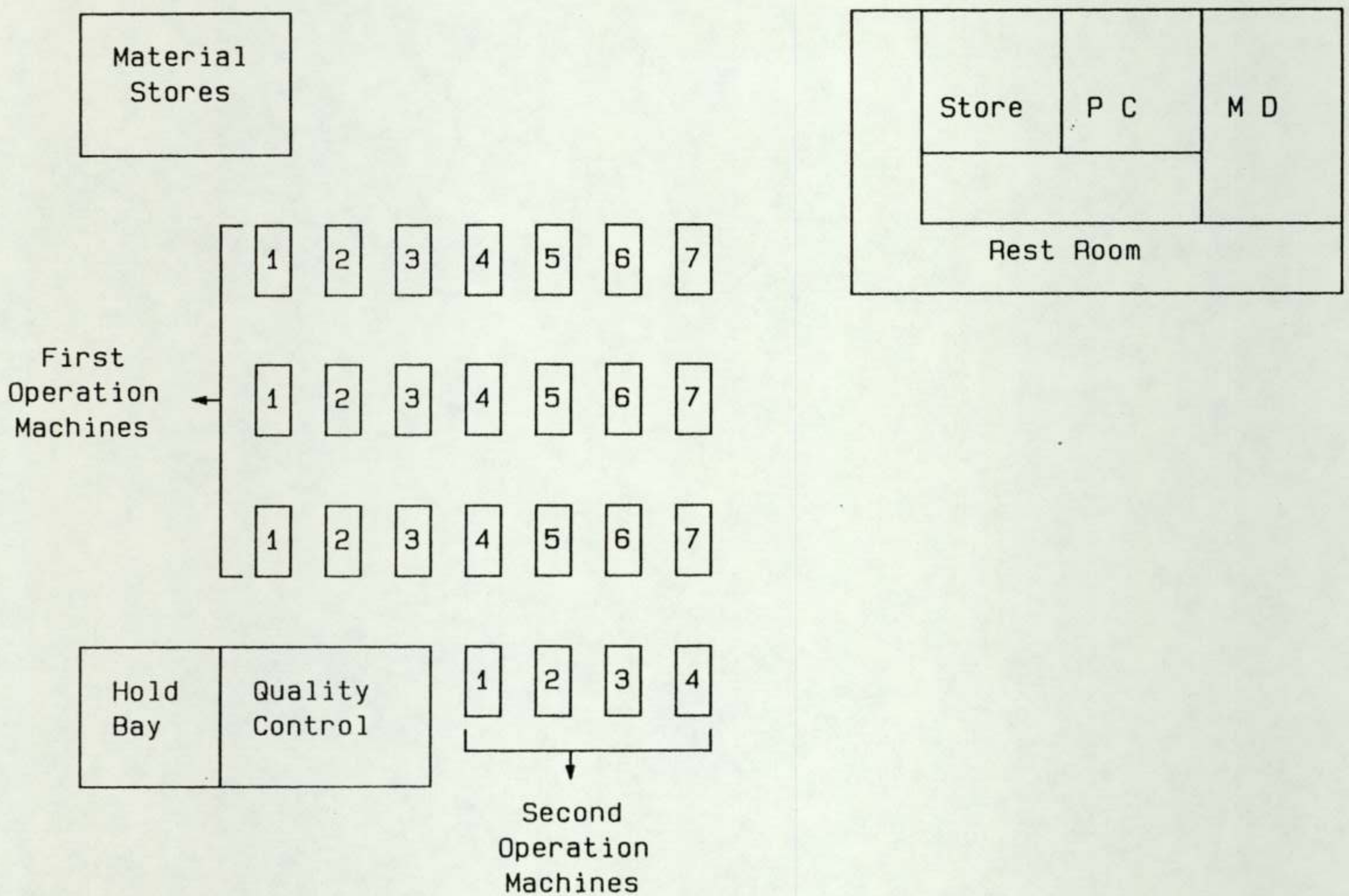


Figure 2.5 : Plant Layout (Schematic)

As the machine shop is the central department concerned with the transformation of material to component, below is a list of machines that actually perform the transformations.

B.S.A 2" - 2 off

Brown & Sharp - 1 off

B.S.A 58 2L - 2 off

B.S.A 58 3L - 1 off

C.V.A 12 - 1 off

B.S.A 98B - 2 off

B.S.A 68 - 3 off

B.S.A 48 Simp - 2 off

B.S.A 138 - 1 off

B.S.A 38 - 3 off

B.S.A 48 Lipe - 1 off

B.S.A 98BL - 2 off

Gridley - 1 off

Second Operation Machines

Drills - 4 off

Band Saws - 2 off

Grinders - 2 off

2.6.1 : Description of Stores

The job shop has a very general purpose store. It contains everything that the job shop will need in its daily functioning. The items include materials, tools, gauges, and odds and ends. These are arranged in a two dimensional array; very much like the pidgeon hole system where we get our mail in large establishments.

The stock level is not considerable, in fact it is just about the bare minimum, as the MD does not wish to tie up capital on high stock levels. Furthermore, as the job shop is in close proximity to many suppliers, the danger of running out of stock is virtually non-existent.

2.6.1 : Interactive Process of the Material System

After the sales order sheet has been received by the machine shop, limited production will begin so that a first off inspection sample can be sent to the customer for approval. This is necessary at an early stage to eliminate the risk of total batch rejection by the customer. This sample is vital, not only for customer satisfaction, but also as a check on the cam design issued by production control, and tolerances and quality of material and gauges.

The inspection department checks the first off inspection sample and requests approval from sales. Once approval has been obtained, full production will commence. In-process inspection will take place hourly in order that non-critical dimensions may be checked and hence complete control of quality established.

After completion of the batch, the consignment is sent to the hold bay where more inspection will take place i.e that of critical dimensions specified by sales department on the sales order sheet. Any defective components are either sent back to the machine shop for rectification or they are scrapped after notifying production control.

If for instance the sampling inspection technique employed implies that the whole sample is not up to standard, a D.B.A.R. form will be used to inform management so that rework, sorting, concessions or rectifications can be organized. After the final inspection stage, the transport department is queried so that delivery can be made to the customer.The above account is for the general production of a component. It must be noted that while the above is in progress, other components are also being produced. This factory produces components for many different customers at the same time, thus making the problem of analysis and solution more difficult. There is also a changing number of machines in the plant as some of the older machines are scrapped or sold, while newer ones are added to either replace or increase the existing plant.

The systems approach to factory modelling is only outlined in this section for detailed analysis would take it outside the scope of this project. The above account is merely for readers to appreciate the type of factory and systems that was encountered at the start of the project.

Chapter 3 : Definition and Solution of problems in the Factory

It has been estimated that conventional metal-cutting machine tools contribute an ammount which exceeds M£500 to the turnover of U.K manufacturing industries (1) . By technological study of the problems alone could make it possible to achieve a vast increase in production capacity and profits. Improvements can be brought about by basic analysis of efficiency in the industrial plant in question.

This chapter deals with the main problems encountered in the job shop as outlined as research areas in chapter 1. Initially a discussion based on sheer logic gives an insight as to the areas to be investigated concerning the actual problems. In the later parts, the actual problems are discussed and solved.

3.1 : Analysis of Efficiency in the Factory

At the start of the project, the factory was running at an efficiency of between 30 and 40%. For all purposes, the efficiency of the entire plant is based on the following equation. This equation was provided by the managing director.

Efficiency = N * CT * 100 / TM --- E 3.1

Where N , the number of acceptable components

CT , the estimated cycletime

TM , the total time available from all machines

The above equation holds true for a time span of 1 month and the time unit to be used is 1 second.
Consider for example a plant consisting of 3 machines and producing 3 products.

Product	CT	N
X	10	1500
Y	7.5	8500
Z	2.5	40000

Assume a full working month as being 20 days and a working day of 8 hours.

Therefore,

Efficiency = 50000 * 20 * 100 / (20 * 8 * 60 * 3)

= 57.87

Mathematically, to increase the resultant, one must either decrease the denominator or increase the numerator, numerically. The time available from all machines is a constant for a time span of one month. Any machines added to the existing plant will be disregarded in the evaluation of the efficiency in the particular period, but will be included in the next period. Hence the only way in which the resultant of equation E 3.1 can be increased is to numerically increase the value of the numerator.

To increase this value, either the number of components accepted at the end of the month has to incresed or a more accurate estimation on the cycletime is needed. For the present we will assume that the estimate is close enough to the actual cycletime obtained during production. This is a fair assumption as recalculation of efficiency can easily be made if significant differences between estimated and actual exist. The actual cycletime is usually obtained after a short production run (usually the first off sample). This is now narrowed down to the number of components accepted in a month.

3.2 : Discussion on problems and possible solutions

Basically, low efficiency is due to a significant proportion of the components that are produced being rejected at the end of the month. A component is not accepted due to one or both of the following :

i) The component is dimensionally inadequate.

ii) The component is materially inadequate.

A components dimensional inadequacy is usually due to the following :

a) Inaccurate cam design.

b) Inacurate machine setup at initial or reset stage.

A components material inadequacy is usually due to the following :

a) Supply of poor quality material (chemically)

b) Wrong material used

c) Poor standard obtained on return from external machining (e.g anodizing).

Looking at the above factors, we find that accuracy improvements on design of cams is the only area where human error or insufficient skill could be eliminated by computer aided design. There are no straight forward methods to eliminate these other factors except by greater care and extensive inspection and supervision. Undoubtedly this will help increase the number of acceptable components at the end of the month, though it must be mentioned that scrap rate is not a major problem in this factory.

A more serious problem that management faces is that of the low machine utilization factor. A machine can be in only 2 states, i.e either it is running or it is idle. Ideally for 100% efficiency, all machines must be running at all time, which for obvious reasons, is impossible. To study idle machines, it is best to split them into 2 categories. They are :

a) Those machines that have not been assigned a job.

b) Those machines that have been assigned a job but are still idle.

A machine that has not been assigned a job will be idle due to :

i) Insufficient orders

ii) Machine not selected due to poor work load distribution

Nothing apart from advertisment and maintenance of goodwill with customers can overcome the problem of insufficient orders. Poor work load distribution, resulting in machines being underloaded or idle while others are overloaded is a problem concerned with scheduling and shop loading. Vast improvements can be made in this area since no scientifically feasible method is being used to distribute work load in the factory at present.

A machine though assigned a job may be idle due to :

i) Waiting for metal, tools, or gauges

ii) Machine breakdown

iii) Changeover of jobs

iv) Maintenance setting

v) Bar feeding and running empty

vi) Absenteeism of operator

vii) Questionable cam design

viii) Labour attitude

Good inventory control and systematic scheduling of both stores and suppliers will help reduce waiting time or perhaps even eliminate it altogether. Machine breakdown is inevitable but could be reduced to a minimum by periodic maintenance. The problem of waiting for service is more damaging to the flow of the system than actual breakdown. A suitable maintenance crew could be employed to reduce waiting for service time. Changeover time from one job to another is directly propotional to the complexity of the component to be produced. Good scheduling and advanced notice as to which job is to be assigned to specific machines could help minimize changeover time. Absenteeism was an area of interest as the managing director had indicated inconsistency in the labour force. A simulation based on past data could help clarify the requirements of the manufacturing system. The analysis could also suggest a reserve crew for minimizing production loss on the likelyhood of absenteeism.

The cam that the semi-automatic machines require have to be designed accurately and feasibly. The method of design is straight forward and could quite easily be handled by a digital computer. All necessary information required for the design could be computed in an incredibly short time and any queries could be rechecked instantly.

Apart from the above mentioned possible improvements, the factory had a special order that required attention. They produced a particular component each and every month of varying batch sizes for a food processing company. The problem they had was in the selection of resources required to produce this component at minimum cost, given all the different possible ways. This presented a multi-stage machining process optimization problem and could be solved by various computational techniques.

Hence, work was done in accordance to the gravity of the problem. A decision was made after consulting the managing director and work was carried out as outlined below :

1) Multi-stage machining process optimization for a specific order using the 'Dynamic Programming' technique.
2) Simulation and economic study of labour.
3) Economics of machining theory and optimization of machining data
4) Computerised design of cams for use with the semi-automatic machines
5) Scheduling

3.3 : Multi-stage machining process optimization

Optimal process design in multi-stage machining process can be considered as a multi stage decision making process in which the best method of component manufacture should be chosen for each stage of the process taking into account the interrelations between the process in the complete system. The basis of this technique is that any decision or method considered to be the best for a particular process might well not result in an optimal path when the entire manufacturing is considered.

Hence, dynamic programming, a mathematical technique for solving certain types of sequentiol decision problems, characterizes a sequential decision problem as a problem in which a sequence of decisions must be made, with each decision affecting future decisions. The theory of multi-stage decision process is a branch of applied mathematics and is not a branch of computer programming as the name dynamic programming might suggest, though the format of the technique is best solved using a digital computer.

The theory of dynamic programming as applied in this section must not mislead its potential. Dynamic programming can be useful in a variety of fields, viz the economist, engineer, systems analyst and to anyone who may be concerned with the application of mathematics to real world problems.

3.3.1 : The Dynamic Programming Technique

A complete explanation of dynamic programming would not be necessary as it can be found in most good mathematics books. Most books by the pioneer of dynamic programming, Bellman (2), and Knuth (3). Only a general instructive account is given below to understand the method of implementation of dynamic programming.

In order to make the procedure that should be followed in the general case as clear as possible, consider a very simple case first. Consider the map below in Figure 3.1 where the nodes represent the starting and finishing point of a machining process, and the parameters, and the cost of the operations.

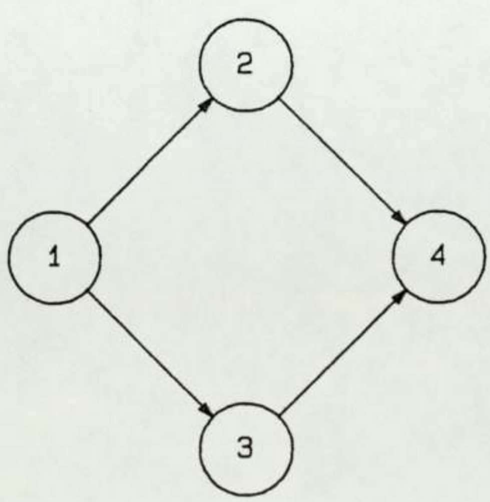


Figure 3.1 : Example of a Simple Network Map

Let function Fi represent the minimum cost of going from node i to the final destination, in this case, node 4. Thus,

$$f1 = \min(C12 + f2 , C13 + f3)$$

$$f2 = c24$$

$$f3 = c34$$

$$f4 = 0$$

The above equations for f2 and f3 are simplified for the above model since reversal of the process is not possible i.e we cannot go from node 2 to node 1. It is also obvious that f4 is equal to zero in a model where there are only 4 nodes since the cost of going from node 4 to the destination, which is node 4, is zero.

The above equations can be condensed to the following general equations :

$$fi = \min(Cij , i = 1 , 2 , ... , n - 1$$

$$fn = 0$$

which are satisfied by the minimal cost (or time) associated with a map containing N intersection points or vertices.

For any map, the equations yield a set of N-1 non-linear simultaneous equations for the desired quantities fi. These equations can then be solved using a method of sucessive approximations.

To use such a method, the vertices of the given map must be divided into sets e.g S1 , S2 , S3 , ... , Sn. In the set Si, the single vertex N (the destination) will be placed. In S2, the vertices placed in S1 will be placed together with all vertices from which vertex can be reached in one step (on a one link path). In S3, all vertices in S2 and all vertices from which a vertex in S2 can be reached in one step.

In general, place in Sk all vertices (including all those already present in Sk-1) from which a vertex in Sk-1 can be reached in one step.

Now by setting for set S1

$$fn(1) = 0 \qquad \text{--- E 3.2}$$

and for each vertex in S2 compute

$$fi(2) = \min(Cij \;,\; fj(1)) \qquad \text{--- E 3.3}$$

where the minimum is taken over all j in S1.

At the next step

$$fi(3) = \min(Cij \ , \ fj(2))$$

where i is a subset S3 --- E 3.4

Note that values of Fi for i is a subset of S2 are recomputed (and probably charged) when the set S3 is considered, since if i belongs to S2, it also belongs to S3.

The general recurance relation is as follows :

$$fi(k) = \min(Cij + fj(k-1))$$

where i is a subset of Sk

The procedure is repeated until the last set S, which contains all the vertices in the map, has been considered.

Hence, at least one value of fi would have been calculated for each vertex. Now, these values can be used to find further approximations using the same recurrance relation. The values of fi that satisfy the system of equations will occur when

$$fi(m) = fi(m-1)$$

By storage of the j values, we will know exactly which vertex to proceede to in order to maintain optimality. To utilize the computer to solve the equations in the procedure outlined above, information will have to be presented in a special format.

To present the network itself with all its connections, consider the matrix below, which has been set out in the required format for the map in Figure 3.1

Connection	Link 1	Link 2
Vertex 1	2	3
Vertex 2	1	4
Vertex 3	1	4
Vertex 4	2	3

Interpretation of the above matrix leads to the fact that vertex 2 and vertex 3 can be reached from vertex 1. Also vertex 1 and vertex 4 are connected to vertex 2. This does not mean that vertex 2 is connected to vertex 3 or that vertex 1 is connected to vertex 4.

To present the penalty, that is the cost or time associated with the change of stage i.e from vertex 1 to vertex 2 for example, the matrix below has been set out in the special format for the same map of Figure 3.1

Penalty	Link 1	Link 2
Vertex 1	2.00	4.00
Vertex 2	98.76	3.00
Vertex 3	98.76	2.00
Vertex 4	98.76	98.76

The above can be interpreted as costing 2.0 units to go from vertex 1 to vertex 2 and 4.0 units to go from vertex 1 to vertex 3. Since it is impossible to go from vertex 2 to vertex 1, the rogue value of 98.76 is inserted for notification. This matrix is used in conjunction with the matrix for connection points described previously.

3.3.2 : The problem in the factory

The factory produces an average of about 8000 in quantity of a particular type of component to an unspecified customer. The customer places orders for these components throughout the year, though on certain months of the year (or seasons), the demand is higher. A detailed drawing of the component is given in Figure 3.2.

For the production of the batch of components, the complete operational procedures are carried out from the odering of material to final inspection of components. There is more than one feasible set of operations that could be employed to produce the batch, the set selected may not be of an optimal policy. The dynamic programming technique outlined in the previous sub-section could be applied to select the optimal policy from the many available policies.

For the succesful implementation of dynamic programming, both the variables and the optimal policy has to be defined. Once the mathematical statements have been written and the recursive equations obtained, an iterative procedure may be executed via a computer program to obtain results.

The program in Appendix II reads in the matrices that give information in the particular problem i.e the connection matrix and the penalty matrix and the number of links. The program obtains the first approximation, then a second approximation, compares the two for convergence. If discrepencies exist on comparison the whole procedure continues while keeping track of the next node of transfer in a matrix called the optimal path. If the comparison proves true, the results are printed out.

This program has vast improvements over those written by DA Milner (4) and CK Yong (5). In their techniques, a large array is used to store the succesive approximations as comparisons are being made. Hence if a problem of say 100 nodes were considered and an assumption that the approximations converged after 50 iterations, then storage of 100 X 50 elements would be required. In this program, storage for only 100 X 2 elements would be required. This is possible because if no convergence occured, the contents of the first approximation could be overwritten by that of the second approximation and the whole procedure can be repeated with no further requirements on storage.

So, the problem is to find the optimal policy on producing the component shown in Figure 3.2. The job shop produces on average of 8000 units of this component, and produces it according to the processes given in the Table 3.1 and the sequencing is given by the map following the drawing Figure 3.3. The data required in the form of connection matrix and penalty matrix together with the actual program is given in Appendix II.

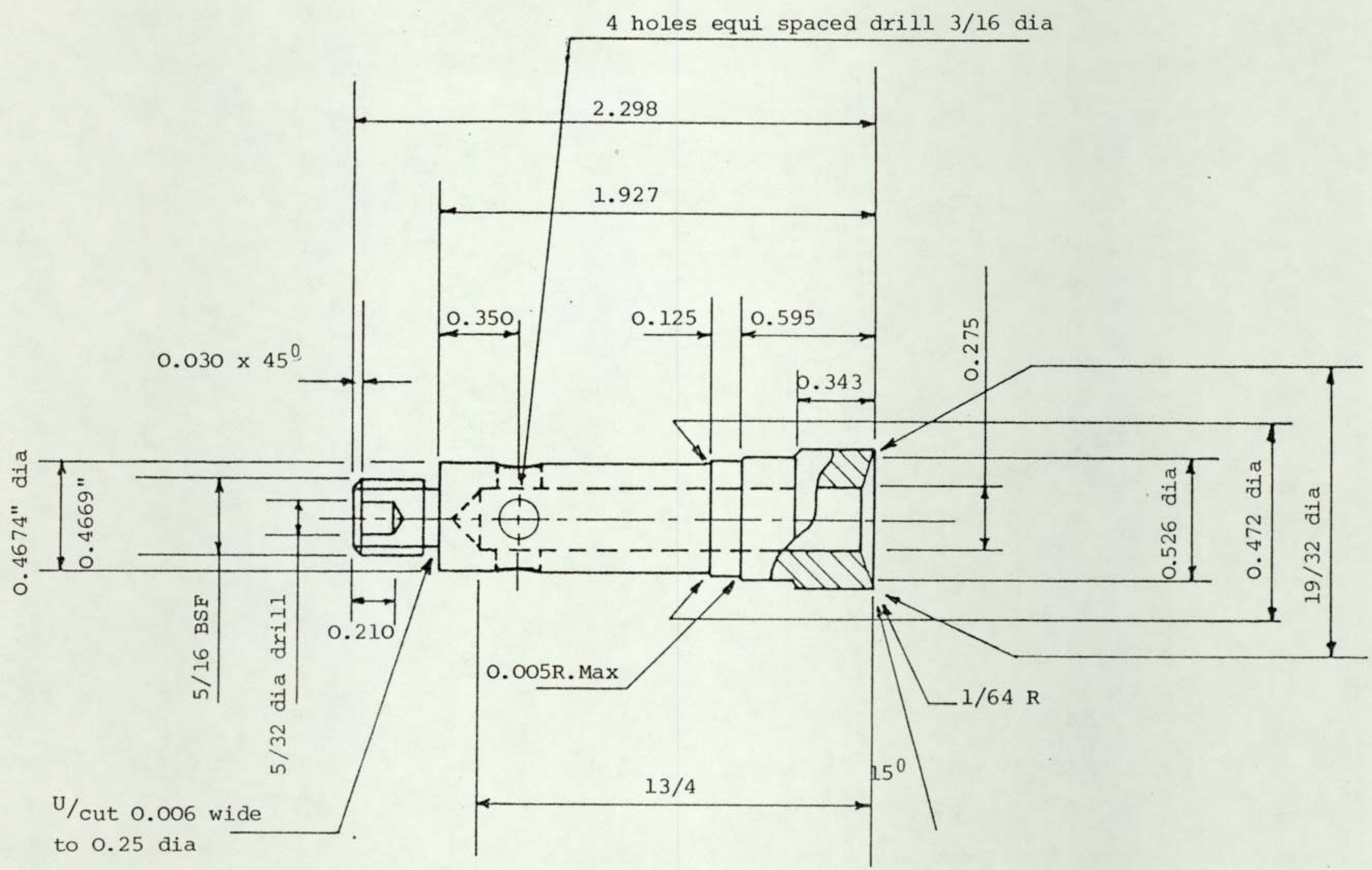


Figure 3.2 : Drawing of Component 1

Table 3.3 : Components machining operations

Nodes	Machining operations	Cost Index
1 - 2	ordering of brass	12.50
1 - 3	ordering of aluminium	4.50
1 - 4	ordering of mild steel	4.00
2 - 5	on small auto-lathe (brass)	1.36
2 - 6	on large auto-lathe (aluminium)	1.50
2 - 7	factured (mild steel)	2.00
3 - 5	on small auto-lathe (brass)	1.36
3 - 6	on large auto-lathe (aluminium)	1.50
3 - 7	factured (mild steel)	2.00
4 - 5	on small auto-lathe (brass)	3.00
4 - 6	on large auto-lathe (aluminium)	3.50
4 - 7	factured (mild steel)	4.00
5 - 8	auto-drilling (brass)	0.84
5 - 9	manual drilling (aluminium)	1.60
5 - 10	factured (mild steel)	2.40
6 - 8	auto-drilling (brass)	1.20
6 - 9	manual drilling (aluminium)	2.40
6 - 10	factured (mild steel)	3.00
7 - 8	auto-drilling (brass)	2.00
7 - 9	manual drilling (aluminium)	2.50
7 - 10	factured (mild steel)	3.50
8 - 11	auto-drilling (brass)	0.60
8 - 12	manual drilling (aluminium)	0.84
8 - 13	factured (mild steel)	1.00
9 - 11	auto-drilling (brass)	0.60
9 - 12	manual drilling (aluminium)	0.84
9 - 13	factured (mild steel)	1.00
10 - 11	auto-drilling (brass)	0.80
10 - 12	manual drilling (aluminium)	1.20
10 - 13	factured (mild steel)	1.50
11 - 14	Centreless grinding (brass)	1.25
11 - 15	in-house grinding (aluminium)	2.00
11 - 16	between centres grinding (mild steel)	2.50
12 - 14	Centreless grinding (brass)	1.25
12 - 15	in-house grinding (aluminium)	2.00
12 - 16	between centres grinding (mild steel)	2.50
13 - 14	Centreless grinding (brass)	1.25
13 - 15	in-house grinding (aluminium)	2.00
13 - 16	between centres grinding (mild steel)	2.50
14 - 17	Inspection of component (brass)	0.20
15 - 17	Inspection of component (aluminium)	0.20
16 - 17	Inspection of component (mild steel)	0.10

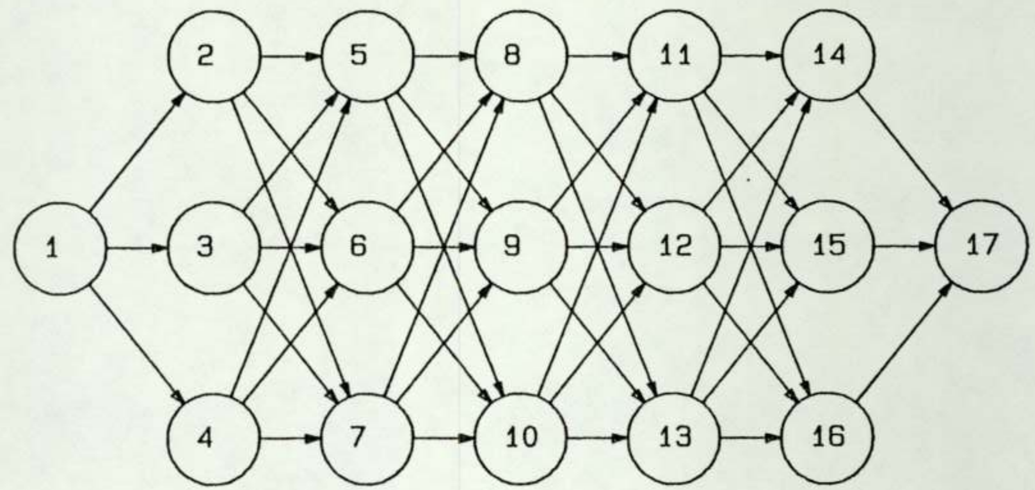


Figure 3.3 : Network Map for Component

Below are the results as output by the dynamic program. A brief description on how to interprate the output from the programme follows.

GO	FROM	TO	COST
	1	3	8.7500
	2	5	4.2500
	3	5	4.2500
	4	5	5.8900
	5	8	2.8900
	6	8	3.2500
	7	8	4.0500
	8	11	2.0500
	9	11	2.0500
	10	11	2.2500
	11	14	1.4500
	12	14	1.4500
	13	14	1.4500
	14	17	0.2000
	15	17	0.2000
	16	17	0.1000
	17	0	0.0000

The above optimal policy can be interprated as follows.

The path 1,3,5,8,11,14,17 will yield best overall optimization of the manufacturing system. If for example node 5 is not available due to machine breakdown or importance elsewhere, the optimal path would be 1,3,6,8,11,14,17.

The optimal path given above resulted in a minimum cost of 8.75 cost units. Now the component had been produced on a 'different' policy, resulting in a cost of 9.5 cost units. Thus the saving per component is 0.75 cost unit and when considering the whole batch, this yield a staggering saving of 6000 cost units a month.

3.4 : Simulation and Economic study of Labour

A brief description of the Personnel System and its interactions with other sub-systems was given in section 1. On discussing the personnel system with the managing director, particular aspects of the system appeared to negatively affect the manufacturing system as a whole. The managing director was aware of the fact that absenteeism had caused disruptions in the production system in the past, though the extent of the disruptions had no significant consequences. It was suggested that a policy to control absenteeism be implemented to try and eliminate future disruptions. Before considering or implementing any policy to control absenteeism, the following questions came to mind :

a) How much absence is there now ?

b) What form does it take ?

c) How many employees are involved ?

Failure to consider the above could lead to ineffectual policy decisions and hence a detailed analysis of the labour situation was proposed. Graphical output too would be useful as visual representation of the labour situation would create a clearer picture of the situation with respect to time.

On completion of the analysis, the natural conclusion led to the Economic study of the workforce. An optimum number of workers could be selected by study of the workforce in conjunction with the workload at hand. Using a simulation technique to represent the economic model, fairly accurate predictions on the size of the work crew required as well as a reserve crew to maintain production flow and to minimize loss caused by Absenteeism.

3.4.1 : Analysis of the labour situation

3.4.1.1 : Absenteeism - The Concept

The terms 'absenteeism', 'absence from work' and 'non-attendence' have all been frequently used as alternatives when refering to the same condition. It has been usual to confine the word 'absenteeism' to situations in which malingering is involved (i.e 'absenteeism' has connotations of anti-social behaviour) (6). The distinction between 'absence', which is self-explanatory, and 'absenteeism', which is described as voluntary absence is related to the distinction between permissible or excuseable non-attendence, and avoidable non-attendence, which does not stem from what the management considers as sufficient justification.

Absenteeism as defined by the British Institute of

Management is

> 'that kind of absence which a reasonable, having regard to all the existing circumstances of the employee concerned, may regard as avoidable'.

It is doubtful whether the identification of causes of non-attendence implied by the above distinctions can be relied upon to reflect the real reasons behind the decision to be absent from work. There is no method by which non-attendence, however defined, can be distinguished from malingering. Therefore such a distinction can be of no assitance in helping management to solve the problem of nan-attendnece. Consequently it makes little difference whether we speak of absence, absenteeism or non-attendence.

For all intents and purposes of this analysis, we shall define absenteeism to be the manifestation by an employee not to present himself at his place of work at a time when it has been planned by mamangement that he should be in attendence, and when he has been notified of such an expectation. No emphasis will be placed on the reason for absence and all that is necessary is that fact at absenteeism exists and that the problem of absence is, in short, intimately linked with the problem of production.

3.4.2 : Measurement and Cost of Absenteeism

Absenteeism means different things to different companies and there are also a number of ways of measuring it. The two basic methods of measurement are in terms of duration (severity) and in terms of episodes (frequency). The following 4 measurements are of interest :

a) Inception rate (spells of absence)

Number of episodes of absence in a year multiplied by a hundred and divided by the average population at risk during the year.

b) Inception rate (person)

Number of persons having an episode in a year multiplied by a hundred and divided by the average population at risk.

c) Annual duration per person

Number of calendar days lost in a year multiplied by a hundred and divided by the average population at risk during the year.

d) Average length of spell

Number of days lost divided by the number of episodes.

One type of measurement alone can be misleading i.e Frequency type or Sensitivity type. The frequency type of measurement represents the average number of absences per worker in a given period. A long absence (due to sickness) would carry the same weight as an absence of one day. It tells us nothing of course about the total time lost because of non-attendence.

The severity type of measurement is a usual concept, as it is easily calculated and allows comparison with the same measure for other units or departments within units. It can be used to indicate trends in absence rates and is relatively easily translated into losses in production, given the validity of certain assumptions. There is one serious defect inherent in the severity rate as an indicator of the degree of seriousness of an absence problem. It is open to distortion through the influence of a few prolonged absences, which may cause the rate to appear alarmingly high, when the situation is basically sound. So a fair representation of absentee rate can be obtained using both types of measurement.

3.4.3 : Cost of Absenteeism

The problem of absenteeism is intimately linked to the question of production. Unless an organization is over-staffed or under-loaded, the absence of an employee will result either in a loss of production (output), or else impose an additional strain on those present. If output is not lessened, it is due to the remaining labour force compensating by producing more. This usually involves producing at higher cost, if overtime has to be introduced, or if strain is increased in human or capital factors, which may result in breakdown in the future.

The extent of the loss of production will not necessarily be in proportion to the numbers involved. A great deal will depend upon the nature of the work process and upon the position of the particular person who is absent. The non-attendence of key workers in linked process industries will be the most damaging, because losses will include not only the potential production of the individual but also of that of the other workers dependant on the key worker. Replacement workers can be held in reserve, but these may be less efficient than workers who are absent. In capital-intensive industries efficient production depends on achieving a high degree of capital utilization, and in these circumstances the implications for costs of absenteeism can be far reaching.

The gravity of absenteeism can be appreciated from the equation below, which used statistics published by the Department of Employment (7). In June 1969, the total number of male employees in manufacturing industry was 6,124,000. Let us assume that the typical working year is fifty weeks. Therefore, the potential number of weeks worked in a year is 306,200,000. If we assume an average absence rate of three percent, the number of weeks lost amounts to 9,186,000 which, assuming a five day week equals 45,930,000 days lost per annum. If we assume a five percent absence rate, the number of weeks lost rises to 15,310,000 (or 76,550,000 days lost per annum). This is for male employees in manufacturing industry alone, approximately 26 percent of the total employees in employment'.

These estimates of time lost through absence from work are the end product of crude calculations. As such they must not be misused as evidence of the losses caused by absenteeism. What the calculations do emphasis strikingly is the very substantial amount of time which can potentially be lost as a result of very small amounts of absence - the rates assumed were low - over an extended period of time.

3.4.4 : Control of absenteeism

If it were decided that some action to control absenteeism was possible, then the relevant question would be to find the least cost method and to ensure that the costs of enforcing this policy does not outweigh the benefits from increased attendence. The goal of absence control should not be an unblemished attendance record, but the elimination of defficiencies created by absenteeism.

A senior member of management should be given the task to coordinate policy for absence control. His first task would be to measure the size of the problem and to identify the worst areas. A cost analysis, including not only the cost of company sick pay but also cost of overtime and overmanning will help obtain support for a full scale investigation. Loss of production and sometimes of sales may also be caused by absence and should also be considered.

Adequate and meaningful records and analysis are necessary to indicate problem areas, to identify the worst offenders and to measure the effect of control techniques. Considerations should be given to incentiive schemes, with or without financial rewards. Such schemes are usually successful in smaller organizations and their benefits may only be shortlived. A programme of other measures should be considered, since any one technique may only be successful for a limited time and a second should be planned to replace the first.

It may also be valuable to study the small proportion of employees who take a lot of time off and also the proportion who are never absent. Such analysis can indicate how hours or conditions of work, incentives, or the degree of supervision may be improved.

3.4.5 : Information from the Managing Director

To carry out a detailed analysis, information was obtained from the MD, pertaining to the personnel system. Unfortunately, the only information available was that of employee names and weekly hours done by the individual employee between the period of Dec 75 and Dec 76. The data supplied was unformatted as no plans were made in the past to analyse the labour situation. A routine had to be devised to tabulate the data not only in a more readable form but also in a form that the computer programs would require. For details on employee names and weekly hours please refer to Appendix I.

From the Appendix we see that a total of 48 were employed during the span of 1 year. This is due to the nature of employement and job description i.e part timers and unskilled labour. It must be mentioned here that at any given time no more than 15 in total are employed. This gives a rough idea as to the frequency of hiring and firing of employees and also to the availability of labour to do the necessary jobs.

3.4.5.1 : The Labour Information Program

With the information provided it is possible to obtain information on individual employees. By doing so, management could be informed of individual employees performance. Also by rewarding those who deserved it, there would be a greater incentive for others to strive towards a better individual record, thus improving the whole workforce.

The following information was required -

a) Most industrious worker

b) Least industrious worker

c) Least absent worker

d) Most absent worker

e) Hours worked by each worker

f) Hours absent by each worker

As the data provided included weekly hours over a period of 1 year, a program was written to read the data sequentially and keep a record of the above mentioned. Computations were carried out where necessary i.e percentages ect. Part of the output of the labour information program follows.

LEAST ABSENT WORKER IS M.KHAN AND WAS ABSENT FOR 0.00 HOURS

MOST ABSENT WORKER IS R.KHAN AND WAS ABSENT FOR 429.25 HOURS

LEAST INDUSTRIOUS WORKER IS S.CHALLENGER AND HE WORKED 24.00 HOURS

MOST INDUSTRIOUS WORKER IS F.MORTON AND HE WORKED 201.25 HOURS

HOURS WORKED IN YEAR IS 21951.25

HOURS ABSENT IN YEAR IS 2375.50

TOTAL EPISODES OF ABSENCE IS 92

LOST TIME RATE = 0.0976497

AVERAGE LENGTH OF SPELL = 25.82

SPELL OF ABSENCE = 0.0037818

P A I A O P = 0.0009455

3.4.5.2 : Interpretation of the results

The results obtained was merely for recording purpose only and indicated that a problem did exist. Stricter measures would be employed to ensure lightening of the problem.

The results were split into -

a) Episodes of absence

b) Hours employed

c) Absence in hours

d) Percentage absent

Of the above items, employees with the highest and lowest figures were noted. Also employees who put in the most and least hours in the year were printed out.

On general information, the lost time ratio is 0.1 ot 10%. This is a fairly high figure as mentioned earlier under the section 'Cost of Absenteeism'. The total episodes of absence was 92 and the average spell of absence lasted for 25.82 hours which is an extremely high figure. This could be interpreted that when a person was absent, it usually lasted 3 days.

To enable management to visualise the extent of absenteeism a program was written to graphically display the information in the form of histograms. 'Gino Graph' routines of the ICL (later transferred to the HPGL Plotter) together with the data provided enabled plotting of the graphs. The plots for the frequency and cumulative frequency of the different categories are given in the next section.

3.4.5.4 : Simulation and Economics of Labour

In many technological situations or systems, it would be valuable to know accurately the effect of altering process variables, such as modifying the sequence of operations or changing the design of the component, in advance of actually carrying out such a change. Most of the situations lead to investigations by the construction and operation of physical models of the system.

A model is required as it is unwise to experiment with a production plant itself to study the effects of change. Similarly, it is undesireable that a new and probably expensive plant should have to be built and operated before it is known whether the operational characteristics of individual machines are compatible with each other.

Physical model building is impractical in such instances but it is possible to build models by using mathematical expressions. The main aspect of this approach is to describe the operating characteristics of each part of the system by simple probability distributions. Models of this type are known as simulation models and they are one of the most powerful techniques available to management.

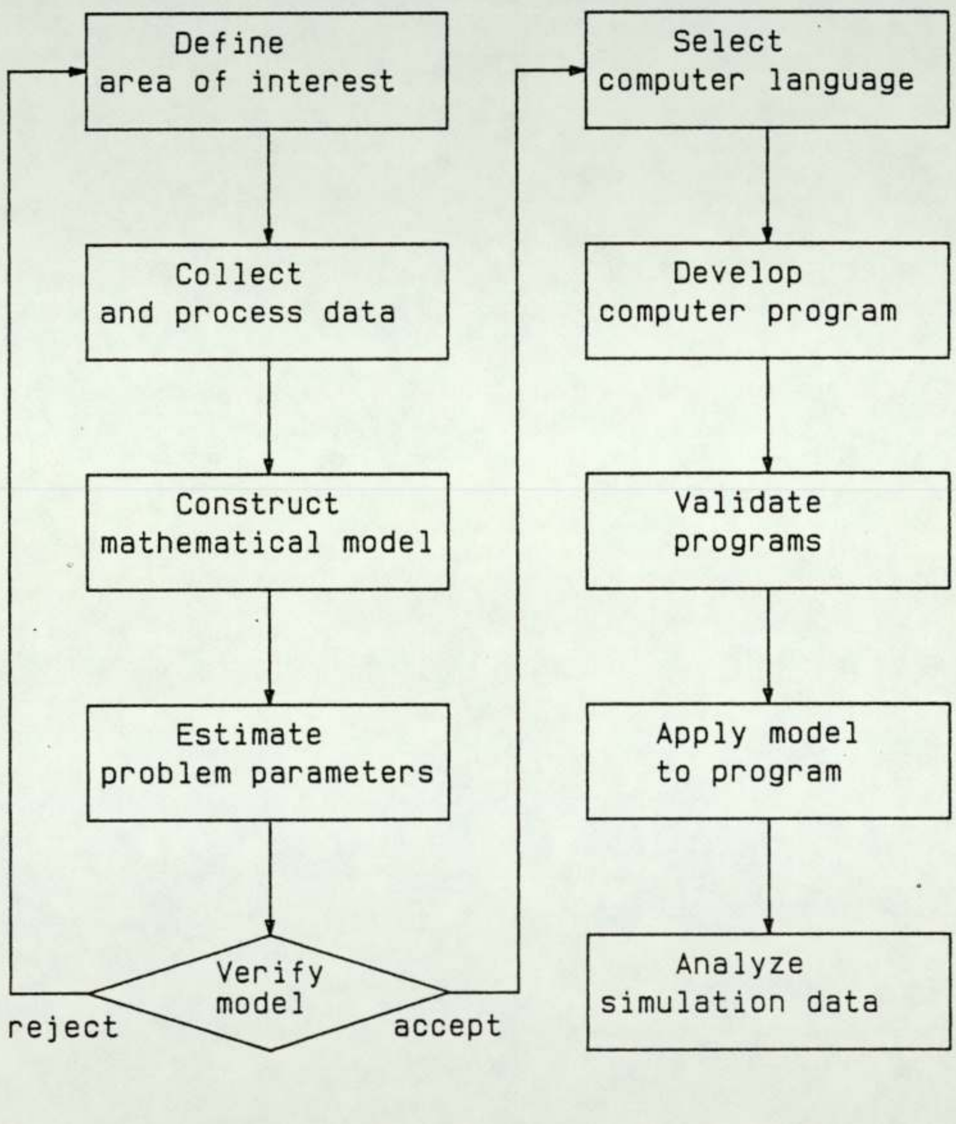


Figure 3.4 : Simulation Flowchart

Simulation may be defined as 'A quantative technique used for evaluating alternative courses of action based on facts and assumptions, with a mathematical model representing actual decision making, under conditions of uncertainty'.

A complete study and description of simulation techniques would take it outside the scope of this project. More information on simulation models and Monte-Carlo methods, a common simulation technique, can be obtained in most management books by (12), (13), (14).

The actual use of the technique will now be outlined in direct relation to the problem at hand i.e the labour situation, after a brief description of the economics involved.

3.4.5.5 : Economics of the situation

When we have decided that the labour force is inadequate to handle the average workload of the production system, we must consider the following.
Could the workload be handled if all the workmen were present for the whole day, every day of the week, every week of the year? If the answer is yes, then we must increase the workforce, on the assumption that this increase is a backup or reserve crew, as absenteeism is

the prime cause of the low production. If the answer is no, then we must increase the workforce logically as the workload increases, i.e the turnover of the company is increasing.

There is of course the possibility that the workforce is not being allocated efficiently, but that is a different matter entirely. We must assume that the best man for the job is already assigned to the job.

The MD was quite happy to say that the answer to the above question is a definate yes, and we must hence consider a reserve crew.

Many methods by analogy could be applied to solve the problem of an optimum reserve crew. Methods used by J R King (16), Palm (17), Ashcroft (18) on determination of a reserve crew to service automatic machines gave an insight to the method to use. To solve this particular problem at the factory, a simulation was preferred.

Now the costs pertaining to a reserve crew is fairly straightforward. If there is absenteeism, the cost due to production loss is going to go up. If we have a large standby crew, the cost to employ the crew is going to be high. So the solution now is clearer. We have to try and offset the reserve crew so that cost due to production loss plus cost of having a reserve crew is kept a minimum.

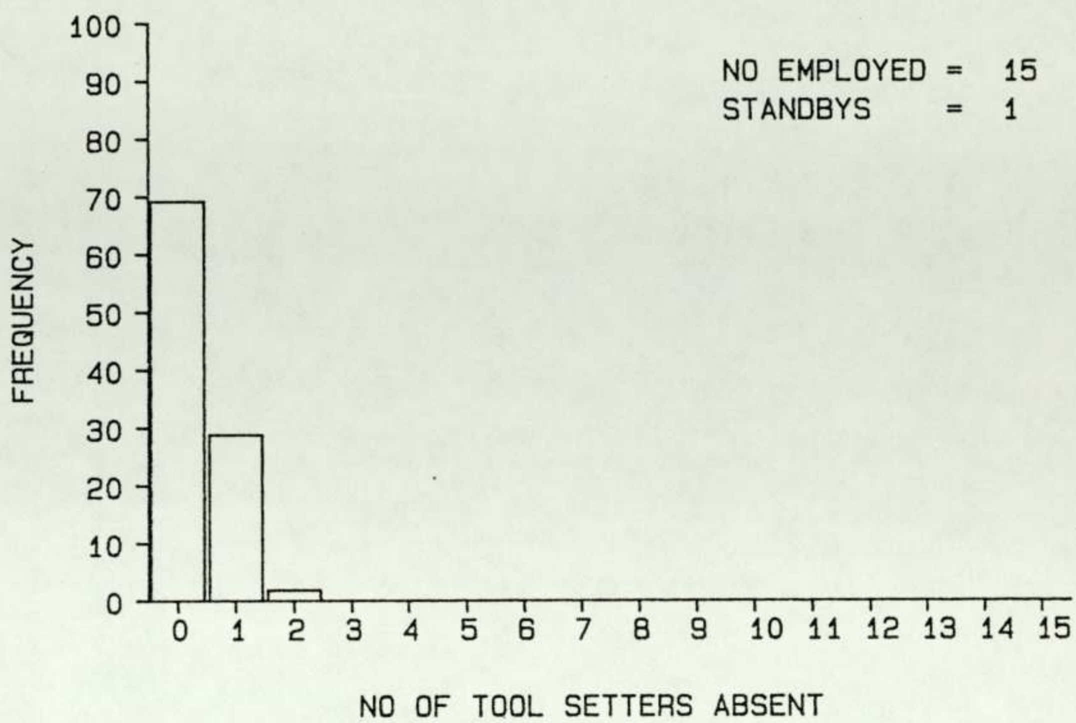
Total Cost = Cost of lost Production + Standby Cost

3.4.5.6 : The theory behind the model

Let us start by considering the frequency of absenteeism i.e the occurances of episodes of absence, which naturally leady to a frequency distribution chart as described in Figure 3.5.

The frequency chart shows how often a particular number of employees are absent in a day. For example it shows that for aproximately 30% of the time at least 1 occurance of absence was noted in that particular category of employees. We only consider episodes of absence rather than individuals because it is immaterial as to who is absent within a working category, since the damage done will be the same.

Now we have to look at the cumulative frequency chart to obtain the probability of absentee. Refering again to Figure 3.5. we can tell that 1 episode of absence per week is going to be the highest due to the vast difference in the heights of the histogram between 1 and 2 opeartors.



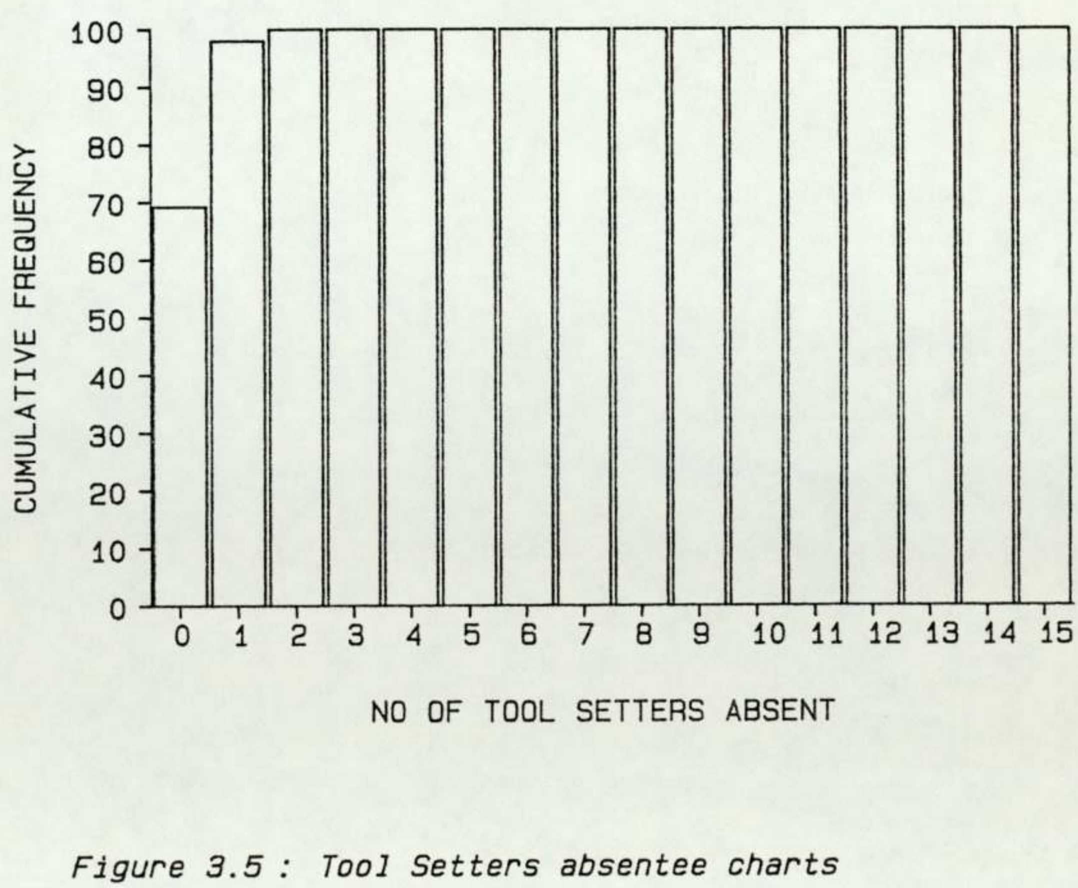


Figure 3.5 : Tool Setters absentee charts

To employ the simulation technique we obtain random numbers using a digital computer (psuedo random actually) and compare them against the cumulative frequency. Every time a random number falls within the range of the cumulative frequency, we increase the probability of that number of episodes happening by one divided by the number of simulations that we are considering. The larger the number of simulations the more accurate the probability.

Once the probability distribution is obtained, an economic theory as outlined earlier can be used i.e

Total cost = Cost of lost Production + Standby Cost

This can be done by assuming first that we have one standby, and one operator absent. We calculate the total cost, then increase the number of operators absent by one, calculating at each stage the total cost, until the maximum case of absentee. We then repeat the whole process using 2 standbys, and keep increasing the number of standbys by 1 until the total workforce is doubled, noting at each stage the total cost. The optimum number of standbys is obtained obviously at the stage when the total cost is at a minimum, i.e the number of standbys that contribute the lowest total cost.

3.4.5.7 : Results and Conclusions

From the output we find the following

Category	Standbys	Cost Associated
Tool Setters	1	3.61
Bar Feeders	1	2.05
Machine Operators	0	0.00
General Workers	0	0.00
Inspectors	1	1.50

It is evident that the simulation was fairly accurate as we see that if even one of the tool setters, bar feeders, or inspectors were absent, production loss would be high. We must also note that the numbers involved in each of the 3 categories are low so that 1 absentee will have a considerable effect.

It is surprising that the simulation portrayed such a picture since the data obtained for the simulation was inadequate for 2 reasons. One, the employees were generally of type temporary and secondly , historical data for only one year was used. It would be interesting to utilise this simulation program on a company where the workforce was consistent and with available historical data spanning a few years.

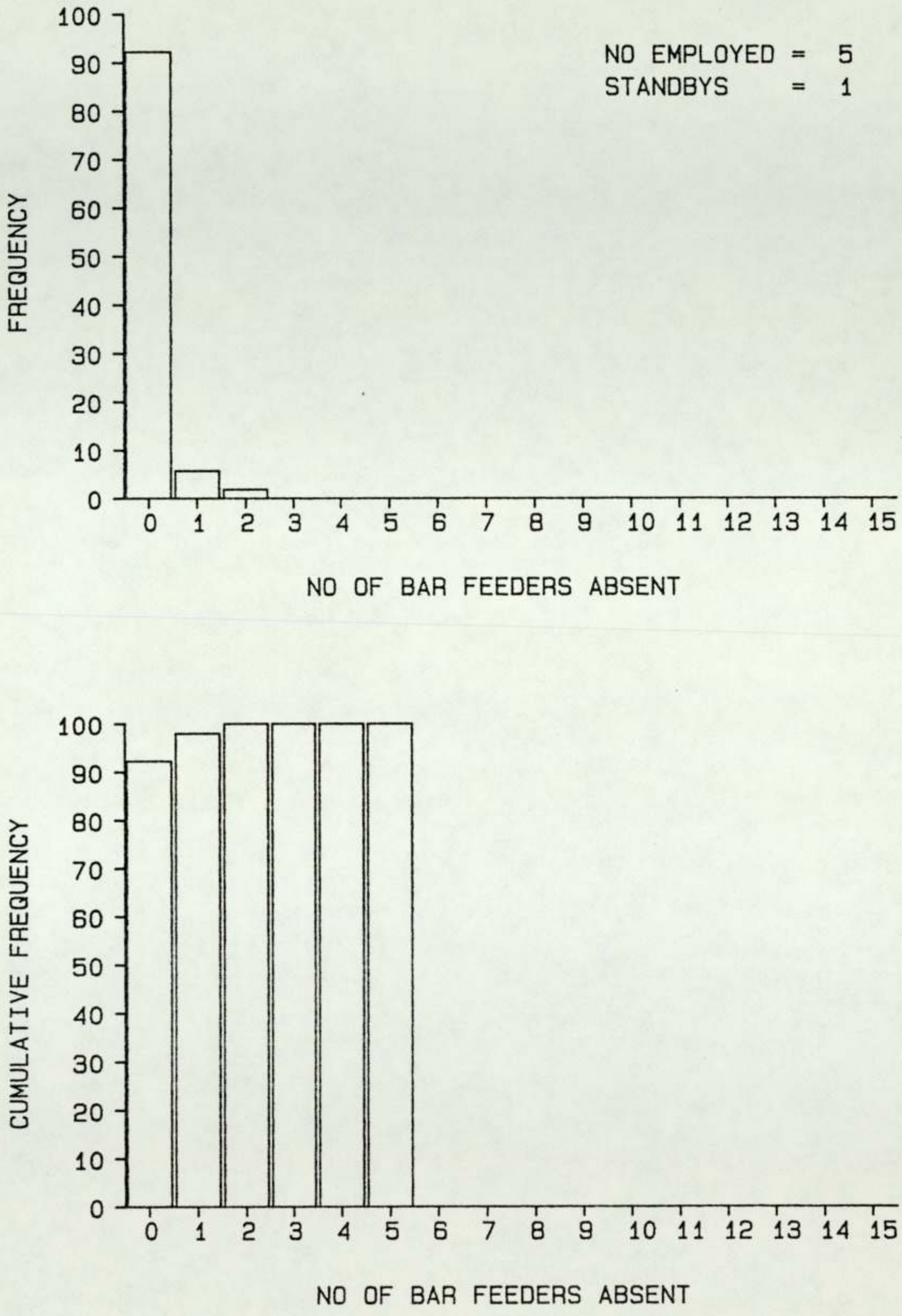
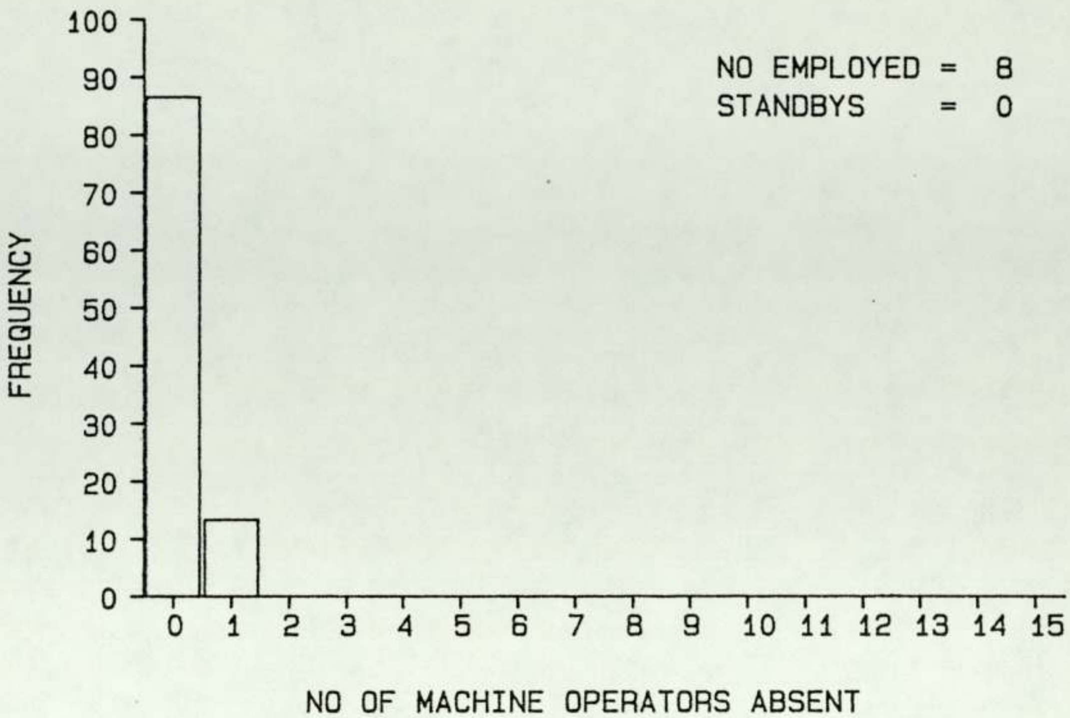


Figure 3.6 : Bar Feeders absentee charts



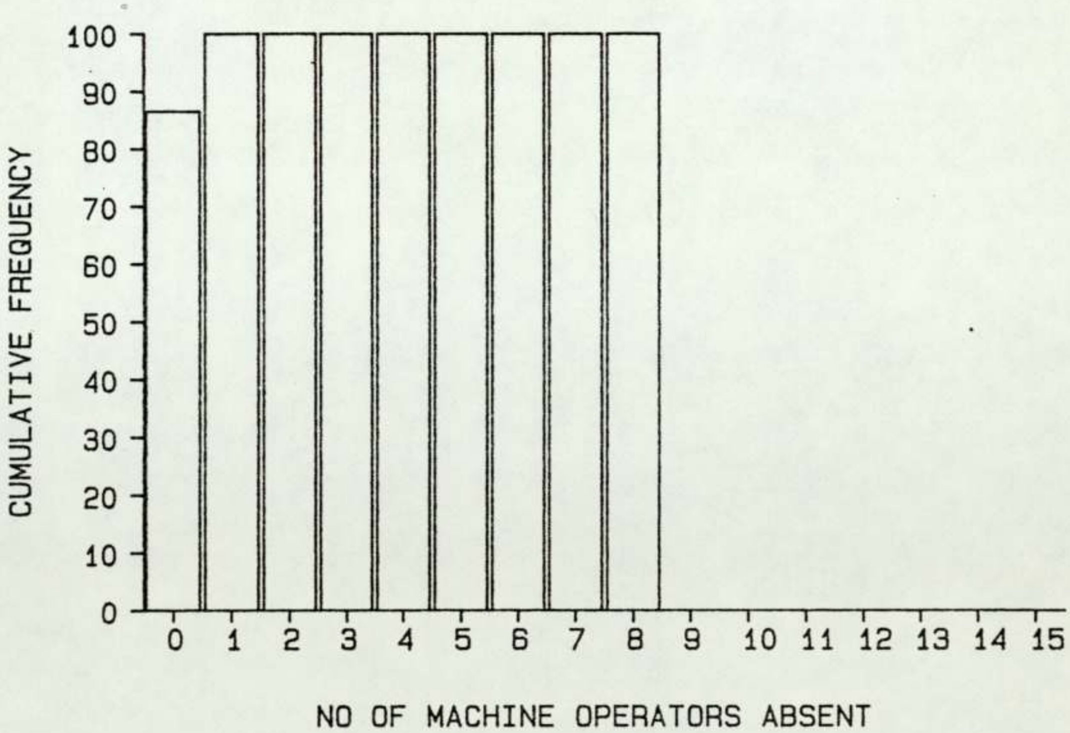
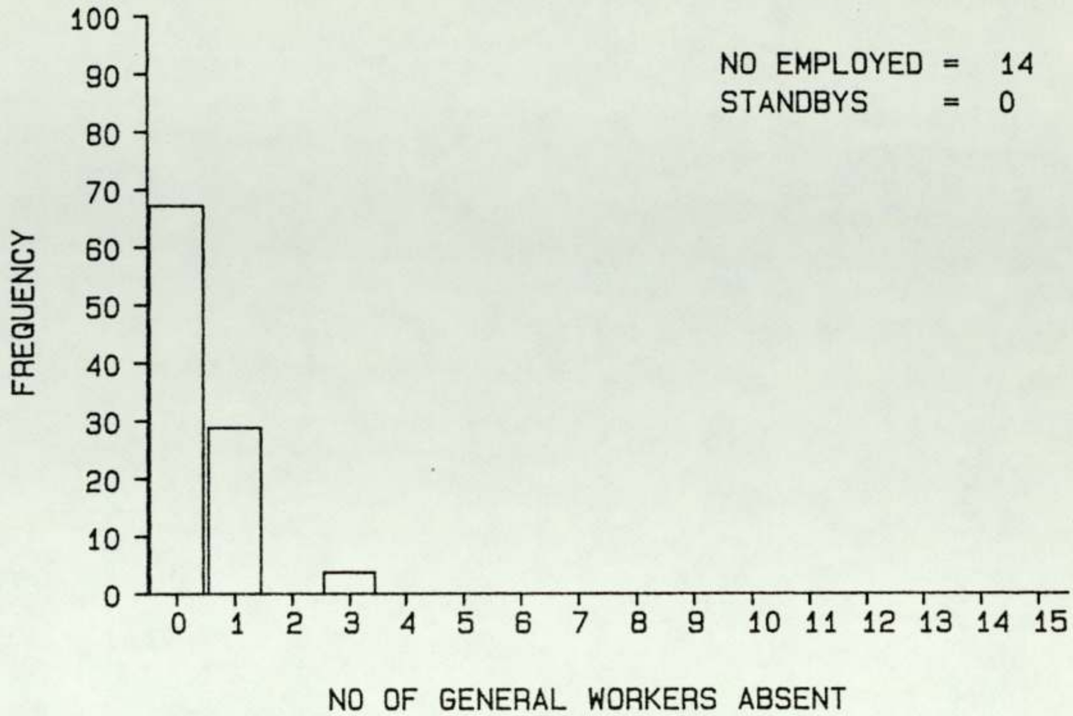


Figure 3.7 : Machine Operators absentee charts



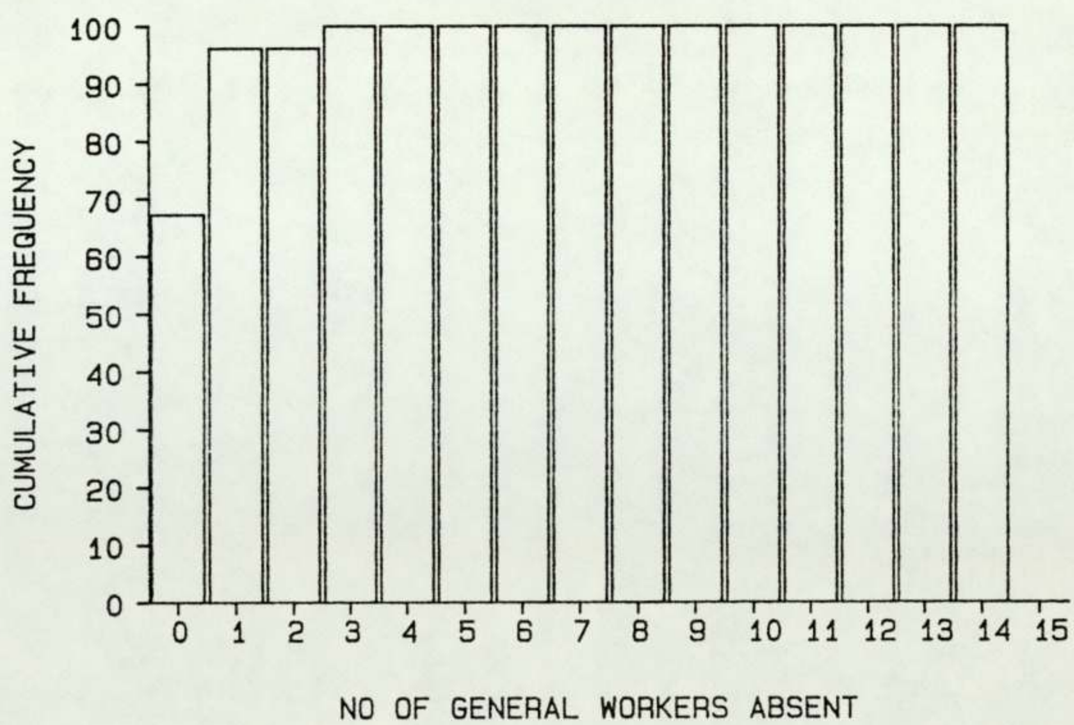
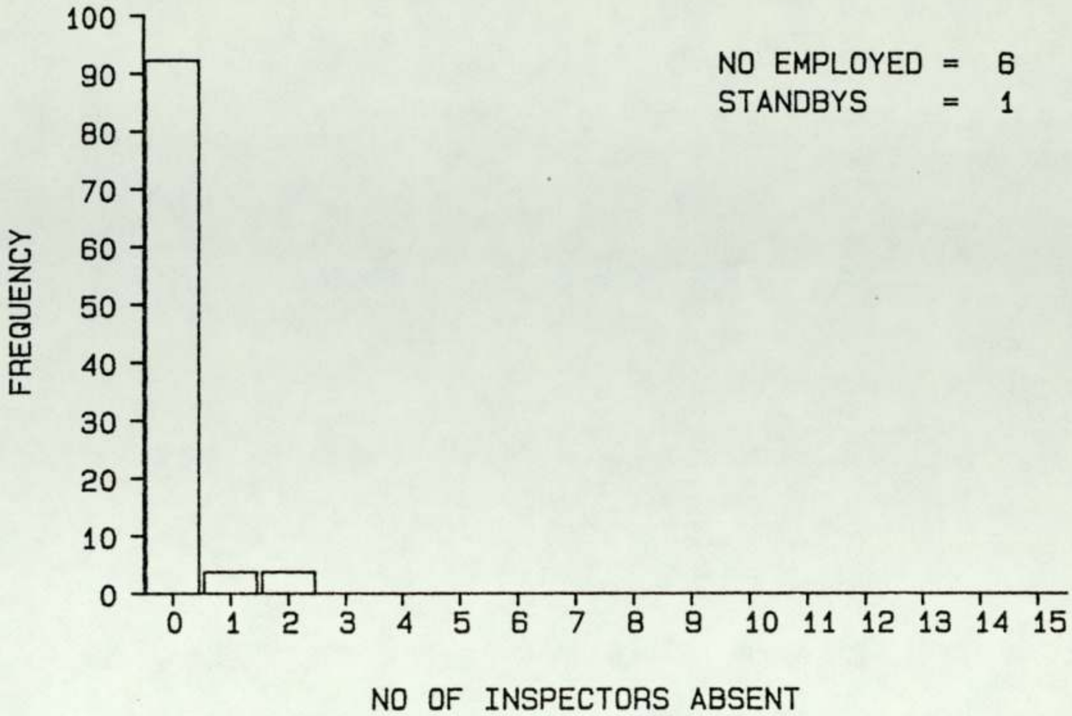


Figure 3.8 : General Workers absentee charts



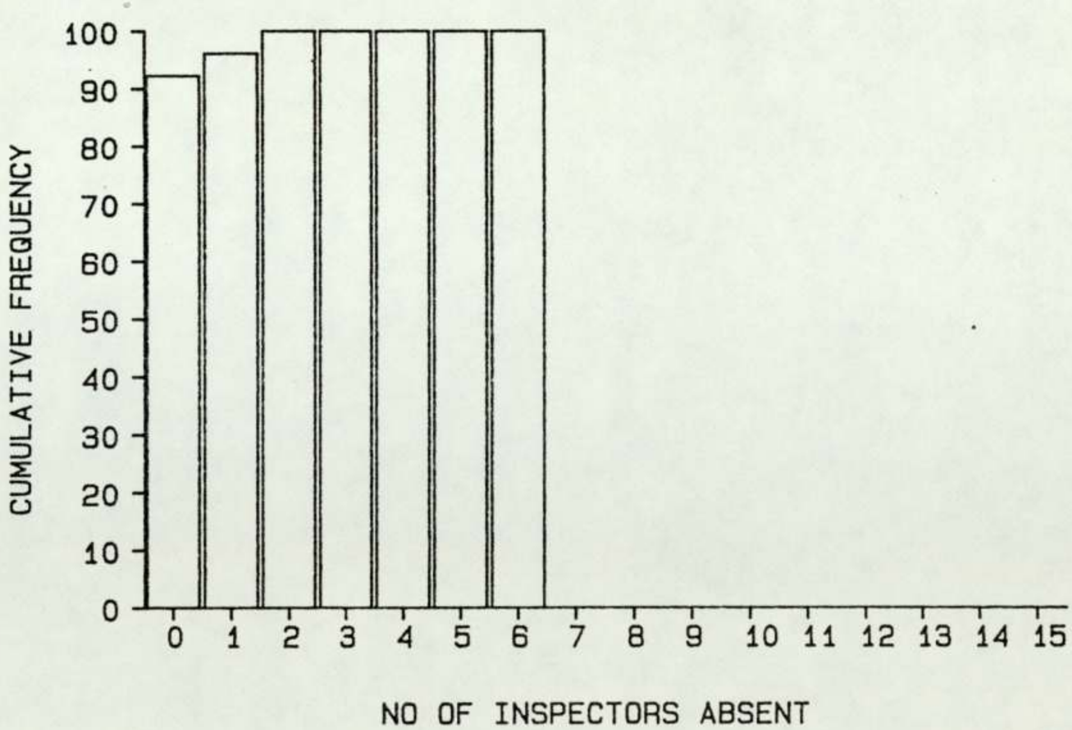


Figure 3.9 : Inspectors absentee charts

Chapter 4 : Machining Theory - Economics and Optimization

One of the factors that contribute to a low efficiency figure is that of poor or ill-designed cams, which are vital for the working of the semi- automatic machines (see chapter3 section1)

A poor or ill designed cam is one that could lead to one or all of the following -

1) poor dimensional quality of the component.
2) low production rate
3) low profitability

Apart from human aspects or 'labour attitude' the above mentioned can be overcome by detailed study of the economics involved in machining. As conventional machine tools spend 20% of their working life idle and no more than half the time spent non-idle is devoted to metal cutting on job. Even when cutting, a machine is generally working at less than 2/3 of its maximum possible production rate (26).

Hence, optimum selection of machining data through a detailed study is of vital importance. It must be noted that poor dimensional quality of the component is not a direct consequence of the machining data used. Such inaccuracies are a direct consequence of the designing technique used in the preparation of the cam chart and will be dealt with in the next chapter.

In this chapter, the economics of machining are studied and the basic mathematical model that describes the cost/time relationship is developed. The 3 main criteria of 'minimum cost production', 'maximum production rate', and 'maximum profit production' are investigated to appreciate the technicalities involved when designing the cam to optimum specifications. Graphs are plotted where relevant to help visualize and explain the effectiveness of parameters of the criteria.

4.1 : The Basic Mathematical Model

For the analysis of the economics of machining theory, let us first consider the unit cost to produce a workpiece by means of a simple rough turning operation. The unit cost is the sum of 3 principal costs -

1) Machining Costs
2) Tooling Costs
3) Handling Costs

The material costs are not considered at this stage as we are not referring to a particular component.

4.1.1 : Machining costs

The machining cost is the cost involved, relating to the time required to machine a component (time being money).

By defination -

Machining cost = Co*Tm

Where Co is the operating time cost,
Tm is the total machining time.

4.1.2 : Tooling costs

The tooling cost is that part of the unit cost that relates to tools, which includes both the actual cost of the tool as well as the cost of the time wasted while changing tools (if necessary).

By defination -

Tooling cost = Ct*Tm/T+Co*Tc*Tm/T

or = (Tm/T)*(Co*Tc+Ct)

Where Ct is the tool cost, (# / tool),

Tc is the tool changing time,(mins),

T is the tool life,(mins),

The ratio Tm/T is the number of tool edges required per workpiece. The tool life can be obtained from Taylors Tool Life equation (27).

VT^nF^m = K

where V is the cutting speed,(ft./min)

F is the feed rate,(ins/rev)

n, m and K are constants.

4.1.3 : Handling costs

The handling cost is the cost that is directly related to the time involved in handling, be it tools, material, stocks etc.

By defination -

$$\text{Handling cost} = Co*Th$$

Where Th is the handling time,(mins).

Now the unit cost, the sum of the 3 principal costs, is

Unit cost = Machining cost + Tooling cost + Handling cost

$$Cu = Co*Tm+Co*Th+(Tm/T)*(Co*Tc+Ct) \quad \text{--- E4.1}$$

Now the total time that the tool is actually cutting i.e the machining time, for a constant depth of cut is

$$Tm = (PI*D*L)/12*V*F$$

Where D is the diameter of the workpiece,(ins),

L is the length of the workpiece,(ins).

To obtain the unit time model from the unit cost model, we must disregrd all the costs.

We hence obtain,

$$Tu = Tm+Tc*(Tm/T)+Th \qquad \text{--- E4.2}$$

From this unit time model we could obtain the equation for production rate Q, as the production rate is the reciprocal of the unit time, i.e

$$Q = 1/Tu$$

Now that the basic model has been fully described we can alter parameters of the model to find its effect on perhaps the time or cost. But before we attempt this we must outline our economic policy and this will lead us to our optimization policy.

4.2 : Machine cutting criteria

We must briefly describe the available methods before deciding on a particular type of policy. The main conventional criteria are -

1) Maximum production rate.
2) Minimum cost production.
3) Maximum profit production.

4.2.1 : Maximum production rate

This criteria is basically for production sequences where 'bottlenecks' exist in the production line. The objective is to produce the most number of components in the shortest time possible. We can obtain the speed Vmax by setting the derivative of Tu with respect to V to zero in the equation for unit time as developed in section 4.1. So we have

$$Vmax = K/(F^m*((1/n-1)*Tc)^n) \qquad \text{-- E4.3}$$

4.2.2 : Minimum cost production

This criteria is based on the assumption that operating at minimum cost conditions tends to increase profits in the long run. Vmin is obtained by setting the derivative of Cu with respect to V to zero, in the equation for Cu as developed in section 4.1. So we have

$$Vmin = K/(F^m*((1/n-1)*(Tc+Ct/Co))^n) \quad \text{-- E4.4}$$

Between these two conventional criteria of minimum cost and maximum production rate, it is known that there exists a range of cutting speeds from which one could be selected to yield maximum profit. On this fact the maximum profit criteria has been brought about by many well known authors such as Wu and Ermer (28), Okushima and Hitomi (29) and Armarego and Russel (30). The more logical criteria of maximum profit production as outlined by Wu and Tee (31) appears to be an ideal starting point for selecting parameters for machine cutting. The following section is devoted to the principles of the maximum profit criteria as this would set up a basis from which to plan out the methods needed for the design of cams in the next chapter.

4.3 : The maximum profit concept

The profit concept involves 7 basic parameters that affect the overall profitability of the production. They are Iu, Co, n, C, Ct, Th and Tc. Iu is the income value of a particular component, while the remaining 6 parameters have already been described in the previous sections.

The profit concept can be subdivided into 3 main aspects

a) The profit region.

b) The profit response.

c) The profit contour.

Any machining operation can be expressed on a per unit basis or on an operating time basis, pertaining to the profit response. The minimum cost criterion expects a maximum profit value based on a per unit item basis i.e

$$P = Iu - Cu$$

Where P is the profit, (£/piece),

Iu is the income per piece, (£/piece),

Cu is the cost per piece, (£/piece).

It is obvious that when Cu is a minimum, the profit, P, is a maximum. But a lengthy operating time could lead to a maximum profit response and is evidently not optimum. The profit response of the machining operation should be based on a time basis rather than on a ti me basis. This leads to

$$P = (Iu-Cu)/Tu$$

Where Tu is the total time per piece,(min/piece).

On substitution -

$$P = (Iu-Co*Tm+Tm*(Co+Ct)/T+Co*Th)/(Tm+Tc*Tm/T+Th) \quad \text{--- E4.5}$$

Using the above equation and assigning a zero profit, a boundary for specific feeds and cutting speeds can be obtained. Obviously we can substitute for Tm,Ct,Tc etc. as they have been defined in section 4.1.

For specific values of profit, there exists a lowest feed rate and a corresponding highest cutting speed which satisfy the equation, and any feed above this lowest feed rate there exists two cutting speeds that satisfy the equation.

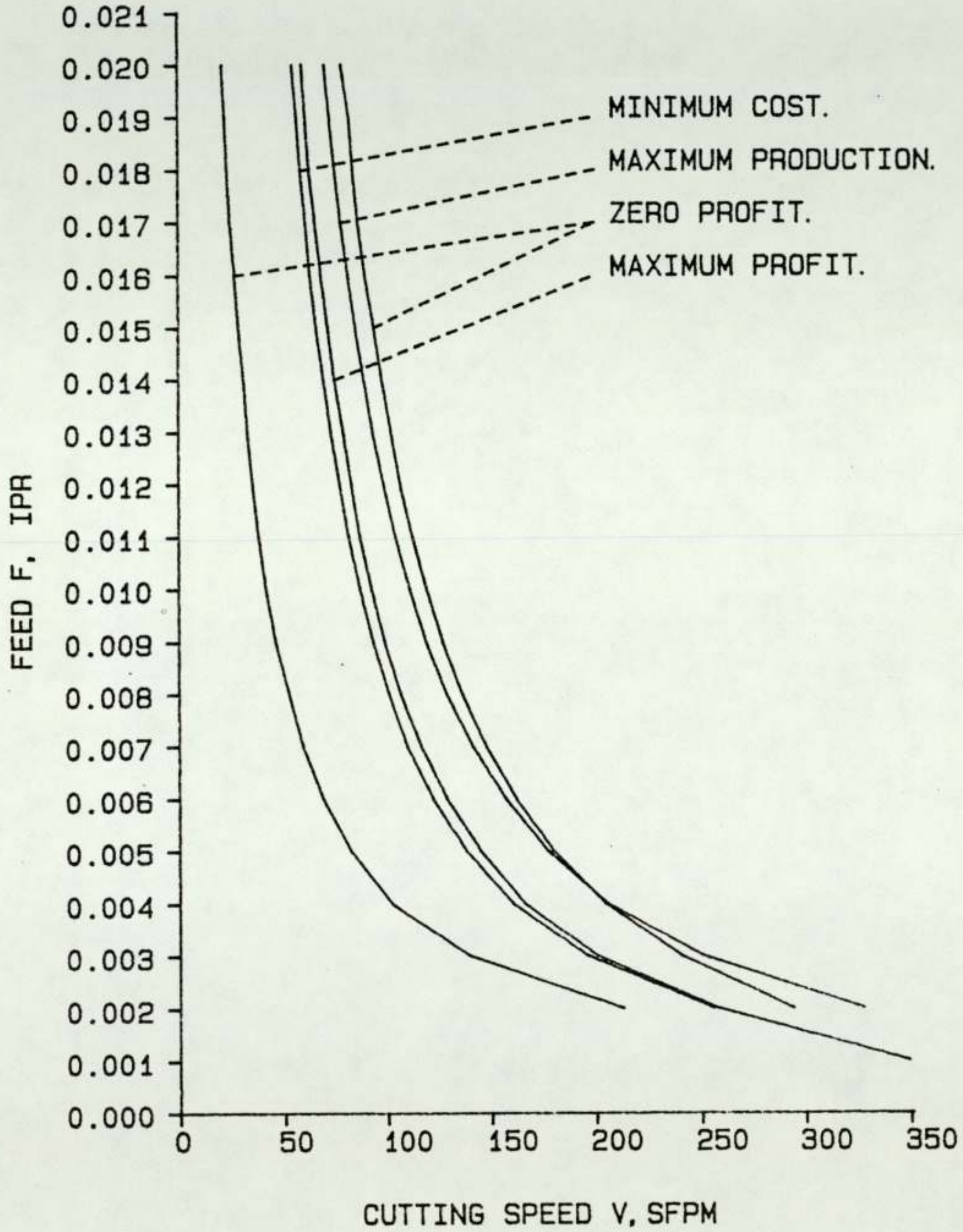


Figure 4.1 : Profit region

Similarly, for any speed lower than this highest cutting speed, there exists two feed rates that satisfy the equation. This whole region bounded by the zero profit boundary is termed the profit region. In figure 4.1, the profit region is plotted togather with the maximum production rate locus and the minimum cost locus.

Three dimensional profit curves have been plotted using the Calcomp 1051 Plotter on the ICL. The profit plots, figure 4.5 show the same profit region but viewed from different angles. The ridge like structure is that part of the profit region that contributes to gain, positive profit response. The flat plane on either side of the ridge shows loss, negative profit response. The changeover from negative profit to positive profit which is a fine line where the surface starts into the third dimension, is termed the zero profit boundary. Also in Figure 4.5 is the ariel view of the same profit response and is termed the profit contour. The zero profit boundary is more clearly defined here.

Similar plots have been produced for cost and demand. In the case of cost, figure 4.6, the minimum cost will be in the lowest part of the ditch. And in the case of demand, figure 4.7, the maximum demand will be at the top most part of the ridge.

Now from equation 4.5 a new equation can be derived that will give the cutting speed at maximum profit. This is obtained by differentiating equation 4.1 with respect to the cutting speed (dP/dV). The new equation is

$$Iu - ((Vp^{1/n} * F^{m/n}) / k^{1/n}) *$$
$$((Ct*Th+Iu*Tc)(1/n-1) + PI*D*L*Ct/12F\ Vp^{n}) \quad — E4.6$$

This equation represents the maximum profit response locus.

To fully study the effect of parameters on the profit response, let us consider the following operation, a single pass, single point turning operation.

Assume the following

Tc = 1 min. / Tool edge

Th = 1 min. / piece

Co = 0.04 £ / min.

Ct = 0.25 £ / Tool edge

D = 3 inches

L = 6 inches

d = 0.2 inches (depth of cut)

Use the generalized Taylors Tool Life equation

$$VT^n F^m = K$$

Where $n = 0.125$

$m = 0.67$

$K = 6.5$

Assume a constant selling price of 0.50 £ per piece i.e

Iu = 0.50 £ / piece (excluding material cost)

By substituting these above values in equation 4.5 and varying the feed F and cutting speed V, zero profit boundaries are obtained as in figure 4.1.

By substituting these above values in equation 4.6, the maximum profit locus can be obtained at different feeds and speeds. We include this final curve in figure 4.1 to obtain an overall picture of the 3 main criteria. The locus of the minimum cost is obtained using equation 4.4 and that of maximum production rate, using equation 4.3. We can see that the minimum cost locus lies to the left of the maximum profit locus and that the maximum production rate locus lies to the right of the maximum profit locus. The minimum cost locus intersects the maximum profit locus at a zero profit point.

Now that we have considered the profit region, we must look at the profit contours which will show the change of profit with respect to feed and speed. In figure 4.6 the profit contours are more widely separated at the left of the maximum profit locus than in the right. This means that any change in cutting speed at the left of the maximum profit locus will cause a smaller change in the profit response than at the right, i.e the closer the contours to each other, the greater the change.

From the contours we also learn that the two loci of maximum production rate and minimum cost can be used as an approximation to the maximum profit locus. For example, at any feed rate the choice of the minimum cost speed Vmin, is prefferred to that of the maximum production rate speed Vmax due to the fact that production at Vmin yields higher profit and any slight deviation from Vmin will not cause a severe change in the profit response. This is important in cases where speeds cannot be selected accurately or that standard speeds available on machines do not cover that of the calculated speed.

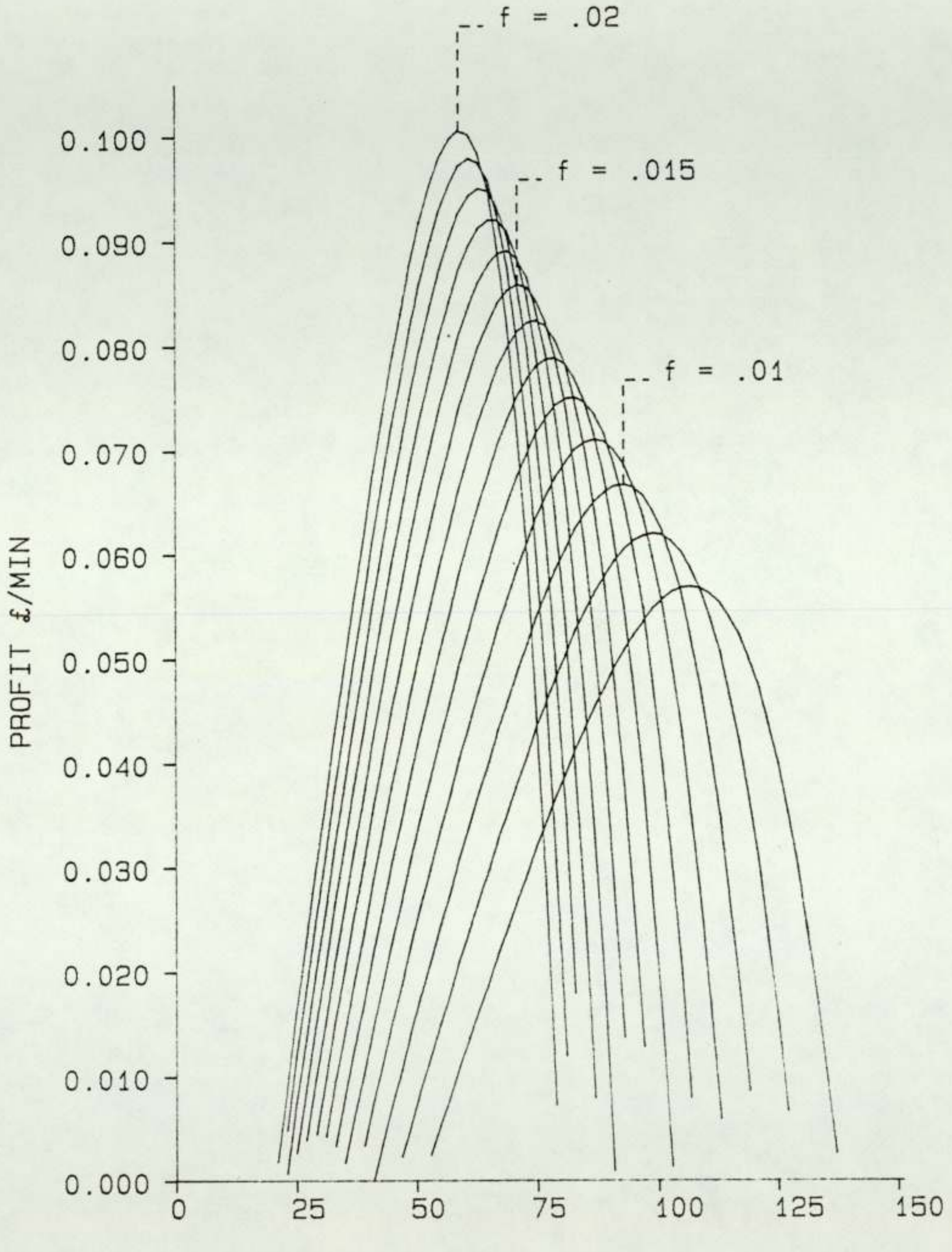


Figure 4.2 : Profit response

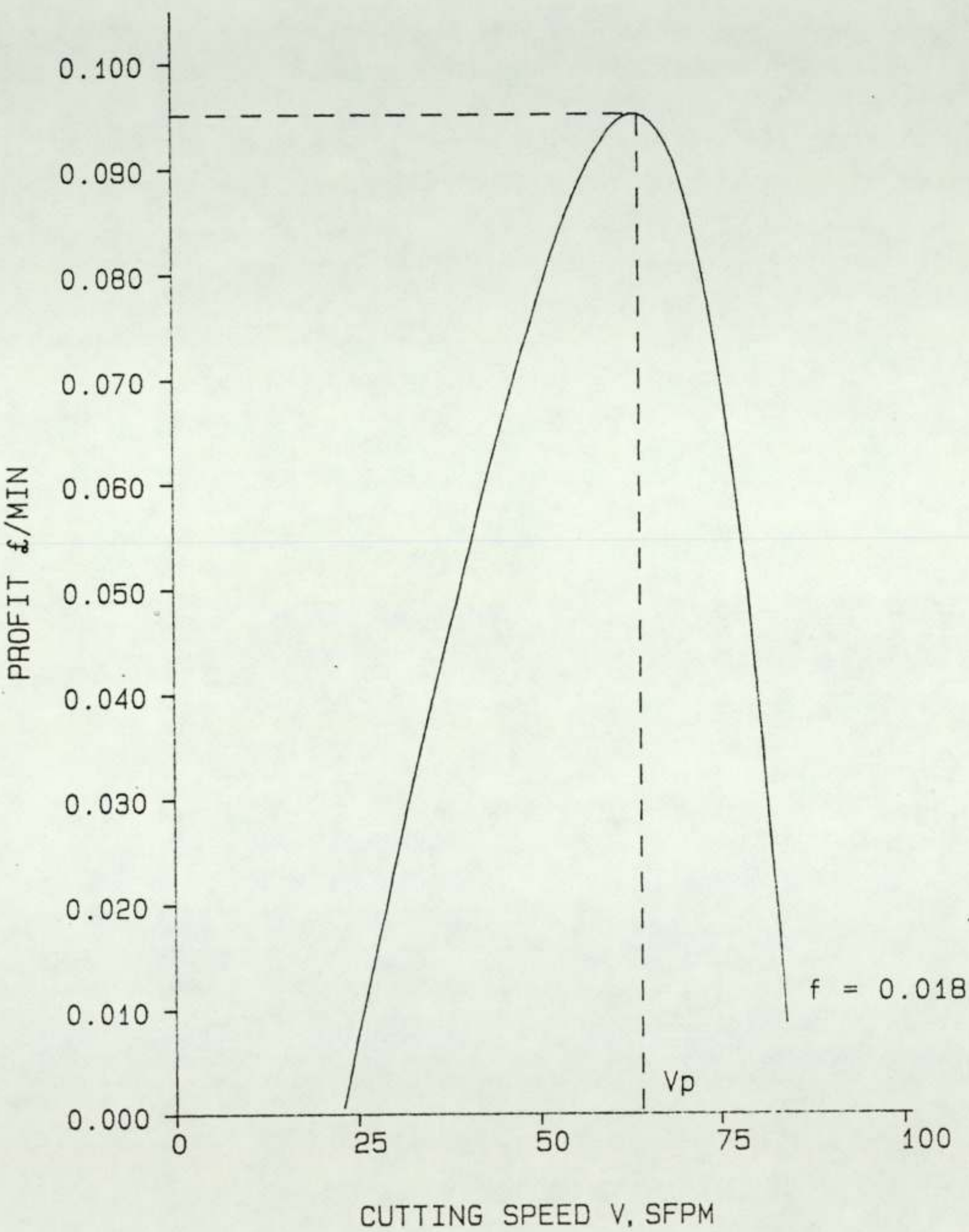


Figure 4.3 : Profit sensitivity

The other important aspect of the maximum profit criteria is the 'profit response'. Referring to figure 4.2, where profit is plotted against cutting speed V, while changing the feed in steps, we see that the maximum profit increases as the feed rate increases, while the maximum profit cutting speed Vp decreases (see figure 4.3 for profit sensitivity). Also the range of the cutting speeds that describe the profit region decreases with increasing speed.

As was discussed with the profit contours, it can be seen that the right portion of the profit response curve is steeper than the left portion indicating that a change in speed which is higher than Vp will cause a greater change of profit response, than will a change in speed which is lower than Vp. We must note the sensitivity of the maximum profit criteria to increases in either speed of feed. We now know that the sensitivity of profit response increases with increasing feed due to the increasing steepness of the response curve, as the feed is increased.

We have developed equations to find the cutting speed that would yield maximum profit but we have not considered the problem of demand. It is pointless setting up a production system to work at the maximum profit cutting speed and later discover we have either underproduced or overproduced within the time span allocated. This problem only exists if there is a specific demand, that is, a certain number of components have to be produced in a specific span of time.

For simplicity let us consider the case of fixed demand. If we have a demand of say 5000 pieces per month and have an allocated time of 20 working days a month and 8 hours a day, then the demand will be

W = 5000 / 20 * 8 * 60

= 0.26 pieces / min.

The fixed demand volume, W, can be expressed in terms of 1/Tu (pieces/min.)

1/W = Tu

= PI*D*L / 12*V*F + (PI*D*L*V^(1/n-1) * F^(m/n-1)*Tc)/12*K^(1/n) + Th -- E4.7

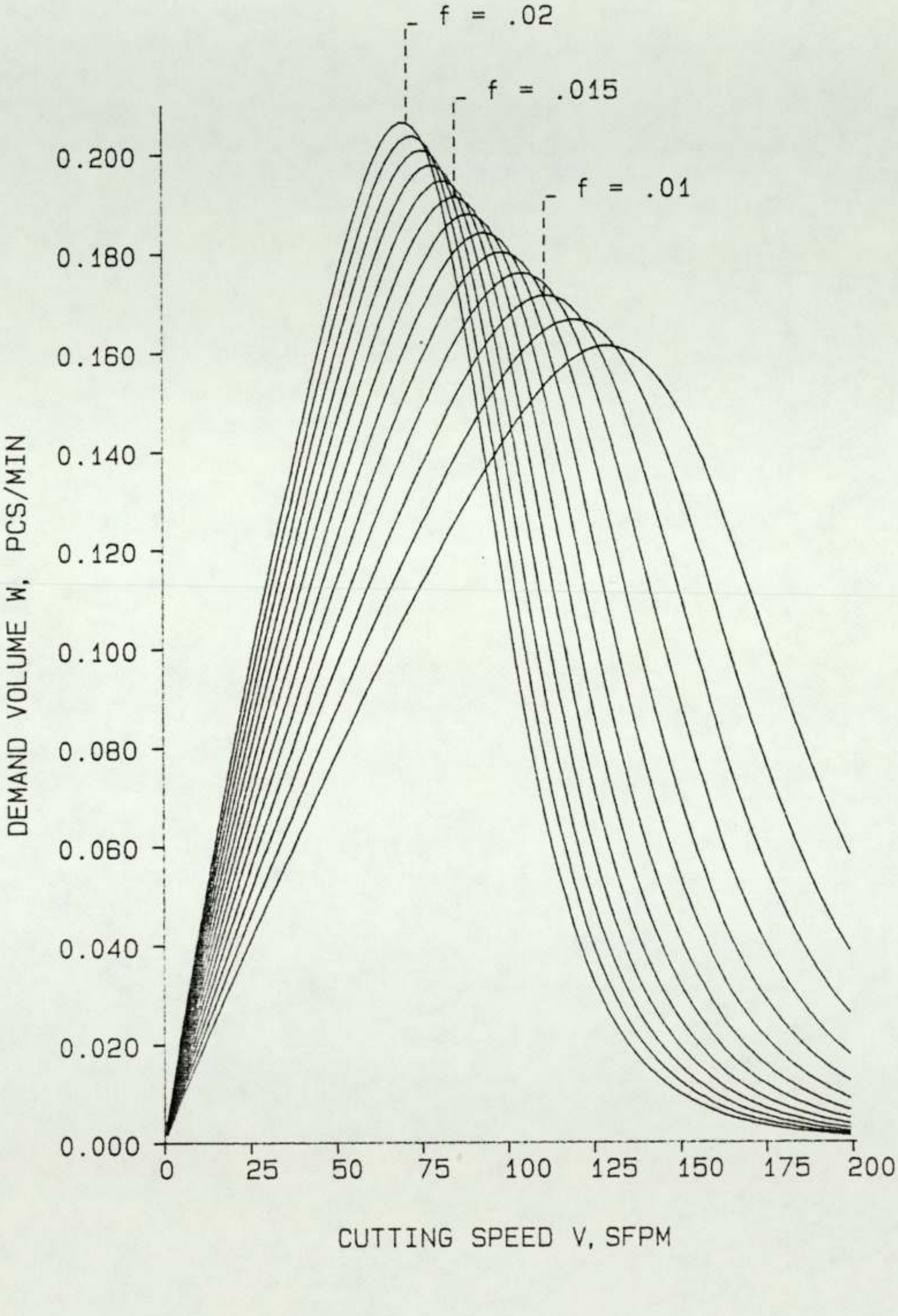


Figure 4.4 : Demand volume

By using this above equation, we can again produce contours and plots that show the effect of cutting speed on demand volume.

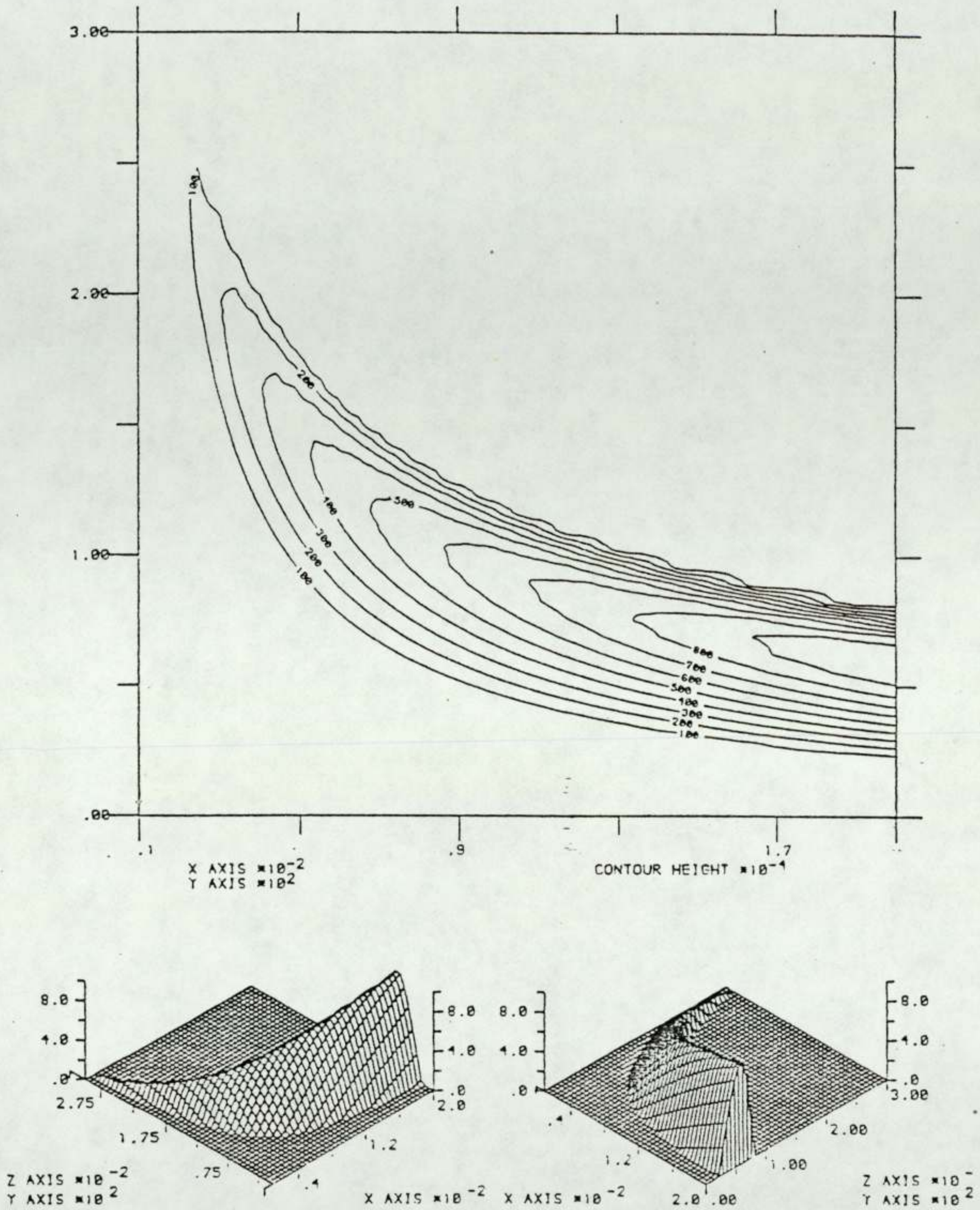
By looking at plots in figure 4.4 we can see that for a particular demand volume there are many feeds and speeds that could be utilized to fulfil the demand. It can also be seen that below certain feeds the demand cannot be met and so we must chose a feed above this lower limit. If we use speeds that do not lie on the demand volume line, we will end up with either overproduction or underproduction depending on which side of the line the speed lies on.

Under the assumption of fixed demand volume, it is clear that the maximum profit cutting speed yields maximum profit but may not be able to fulfill the demand. On the other hand, if we operate at the speeds that will fulfil demand, we may not obtain reasonable profit. Therefore a more realistic optimum cutting speed will exist at an interval between the two speeds of Vd and Vp. This means that our search for an optimum is confined to a smaller region as compared to a search within the whole profit region.

4.4 : Conclusion

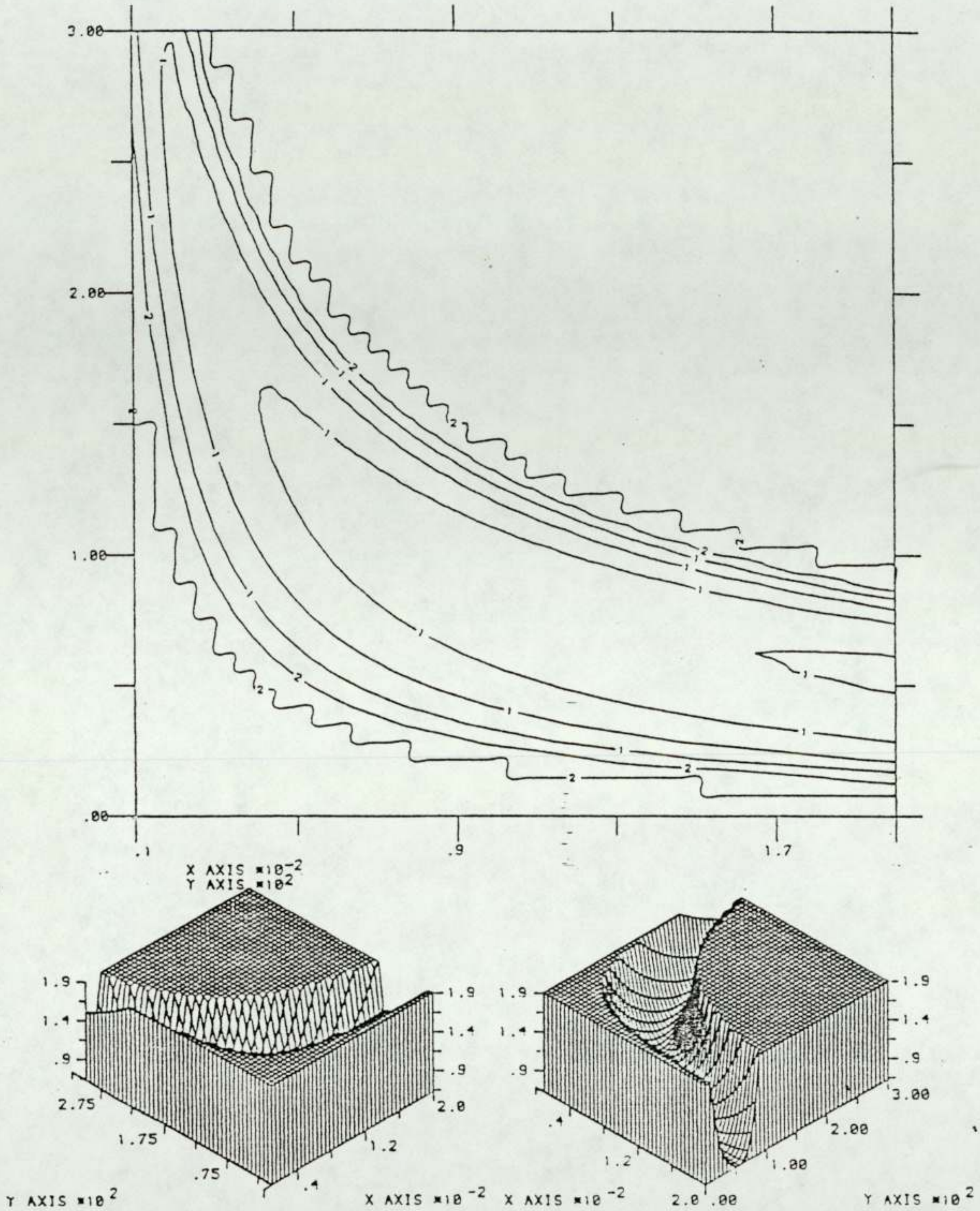
It is true that to maximize the profit we must utilize the maximum profit cutting speed Vp. But this may lead to problems when we consider the demand of the batch i.e to prevent over or under production. Hence, the optimum cutting conditions that will yield maximum profit for a case of fixed demand will lie in the range of speeds between that of Vp (maximum profit cutting speed) and Vd (cutting speed to fulfill demand). The highest allowable feed rate is always used to achieve maximum profit. So a combination of the above two facts would result in cutting conditions that would yield overall maximum profit. Obviously if no specific demand were present we would utilize Vp, but normally there would be a fixed demand.

From the theory of machining we are now in the position to proceed to the next chapter which deals with the design of cams for semi-automatic machines. The selection of feeds and speeds can now be made with confidence as we can also compute the exact profits to be expected. So we should select cutting speeds between Vd and Vp.



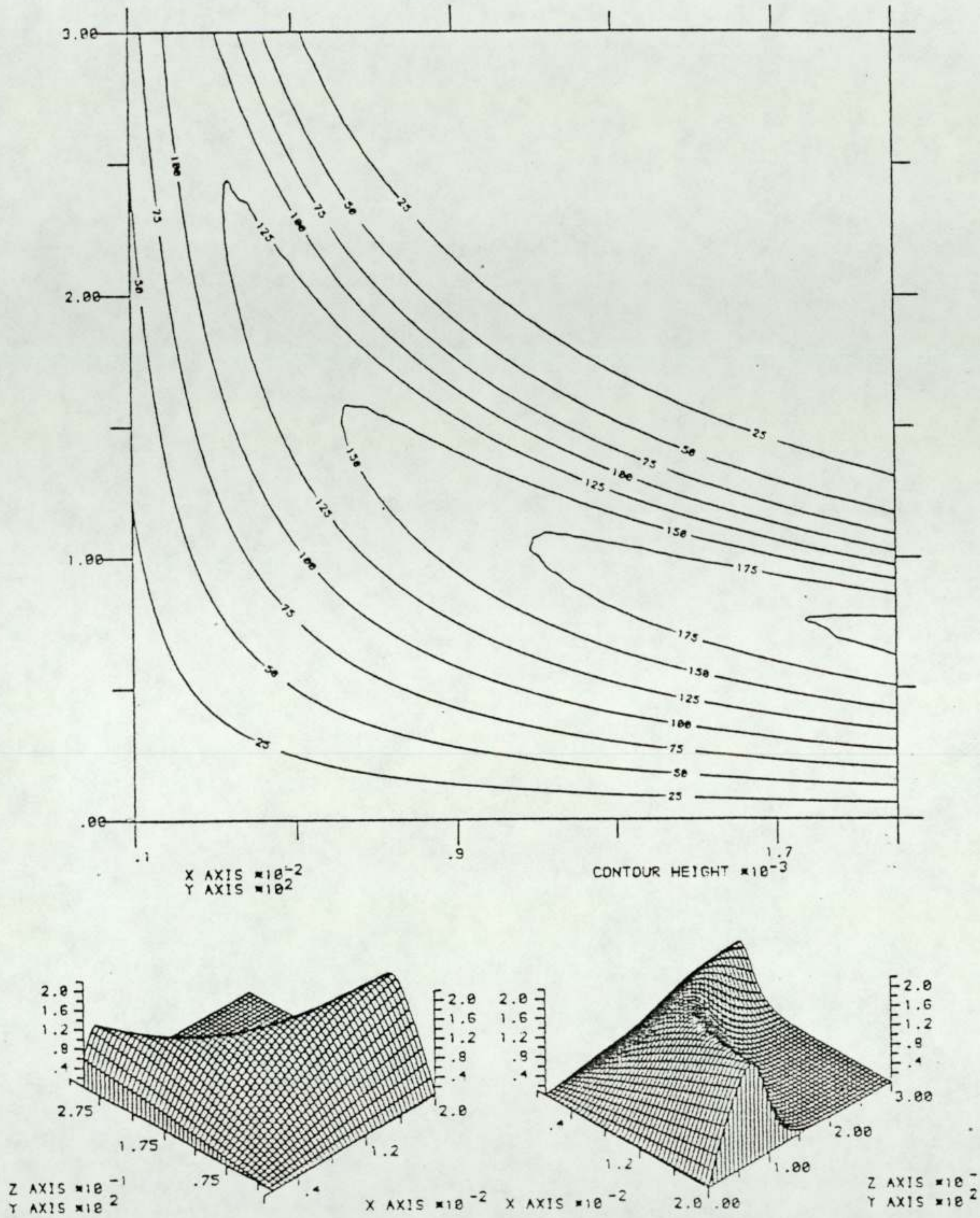
X = FEED RATE
Y = SPEED
Z = PROFIT

Figure 4.5: 3D Profit response and contours



X = FEED RATE
Y = SPEED
Z = COST

Figure 4.6: 3D Cost response and contours



X = FEED RATE
Y = SPEED
Z = DEMAND

Figure 4.7: 3D Demand response and contours

Chapter 5 : Computerized design of cams

This chapter deals with the cam design package. The types of cams dealt with are those required for the working of semi-automatic machines. The work carried out is related specifically to Herbert machines of 'Alfred Herbert Ltd' (51).

Initially a brief description of conventional design of cams is outlined. From the outline it is possible to find areas where improvements can be made. The later section deals with the actual design by computer, implementing the improvements envisaged earlier. It is assumed that the reader is familiar with the terminology used in the design of cams.

5.1 : Conventional Design of Cams

No fixed rules can cover all points in cam design as so much depends upon the requirements of each individual component. We will hence, not deal with unusual tooling layouts but illustrate the procedure usually adopted by means of a typical layout.

The various tool slides are operated by means of a flat disc cam, supplied in blank form, having a centre hole to suit the cam shaft. One side of the blank is divided into hundredths for convenience when marking out.

In order to obtain the time for the production of a component, it is necessary to consider the actual cutting time and the idle movement time, which can be represented by revolutions of the work spindle or hundredths of the cam surface. The 'actual cutting time' depends entirely upon the 'permissible' cuting speeds and feeds for the material used and the depth or length of cut togather with the type of operation at hand. The 'idle movements' represent the time taken by non cutting operations like feed stock, index turret, changing spindle speeds and for clearance between tools whenever it is necessary to withdraw a tool before another can advance into position. Whenever possible, all idle movements should be overlapped with cutting times.

5.1.1 : Design Procedure

The following outlines the general procedure when designing cams.

1) Calculate rpm of workspindle. Note restrictions of permissible feeds and speeds.
2) Set out the sequence of operations to produce the component.
3) Determine travel for each tool operation.
4) Determine feed per revolution of workspindle for each cutting tool.
5) Calculate number of revolutions of workspindle for each operation.
6) Consider at this stage the possibility of overlapping idle movements with cuuting operations.
7) Calculate number of revolutions of workspindle for idle movements.
8) Determine hundredths of cam surface required for clearance and dwells.
9) Calculate number of revolutions of workspindle to finish one component.
10) Calculate hundredths of cam surface for each operation.

The various aspects and considerations above are best described with an example.

From the above it can be seen clearly that the task of replacing the manual method of cam design by a computerized method is fairly straight forward. As long as the steps are clearly defined and one stage leads to another, the programming required is not too complicated.

An Example

Consider the component in figure 5.1 of a brass nipple of overall diameter 0.375 inches and length 0.75 inches.

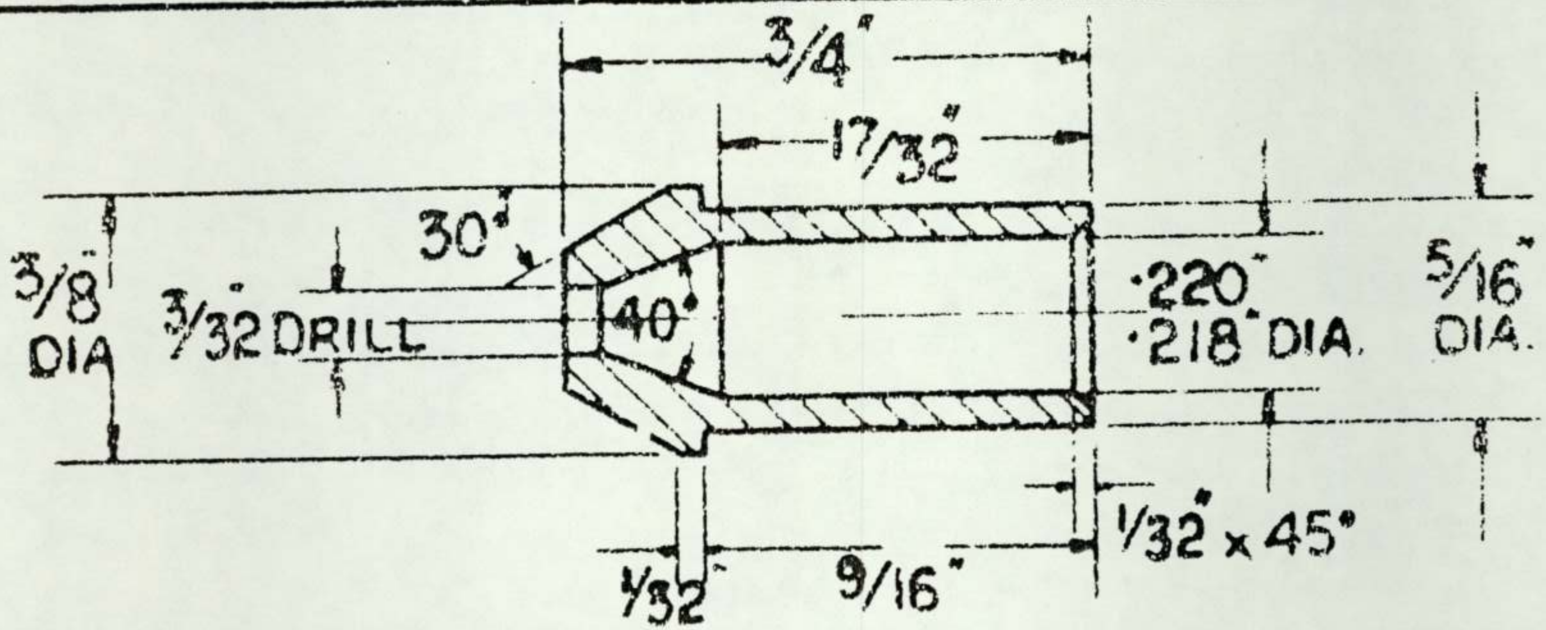


Figure 5.1 : Drawing of Component 2

Step 1 : To determine the revolutions per minute of workspindle.

As the diameter of the component is 0.375 inches and the material is brass, the maximum allowable cutting speed is 600 feet/minute. Tables that give maximum cutting speeds for different materials are readily available. Now working from 600 ft/min we must try to obtain an rpm of the workspindle that would result in a cutting speed not exceeding 600 ft/min. Hence,

$$(PI * D * N) / 12 = V$$

Where D - Diameter (inches)

N - Revolutions per minute (rpm)

V - Cutting speed (ft/min)

Now the above procedure is repeated for each and every cutting operation necessary to produce the component. If it is found that for a particular operation, the cutting speed is exceeded while using the preselected rpm, the whole procedure will have to be restarted; this time selecting a lower rpm. This is done so that all cutting operations are within the safety limits as specified by the cutting speed tables.

Step 2 : Sequence of Operations

For this component the turret and cross slide tools will be used in the following order:

1) Feed stock to turret stop.
2) Centre drill end of bar.
3) Turn 5/16 inch diameter and drill 0.209 inch diameter hole.
4) Drill 3/32 inch diameter hole.
5) Cone bottom of 0.209 inch diameter hole.
6) Ream 0.218 inch diameter and cone.
7) Cut off piece from rear cross slide.

The forming operation from the front cross slide is overlapped on the drilling and reaming opeartions. The above steps are logically dependant on the component.

Step 3 : Travel for each tool.

The travel is the total length of the cutting operation for each cutting operation. Total because the travel includes any extra approach to avoid damage to the tools or component.

Operation 1 : Centre Drill

Total travel is equal to travel by centre drill plus 0.010 inch approach plus 0.005 inch residual hole left by 3/32 inch dialeter drill and is equal to 0.120 inch.

Operation 2 : Turn and drill

The travel of the turning tool exceeds that of 0.209 inch diameter drill therefore the total travel will be 0.562 inch plus 0.008 inch approach which equals to 0.570 inches.

Note : 0.209 inch diameter hole is closest to 0.218 inch hole required before reaming.

Operation 3 : Drill 3/32 inch diameter

The 3/32 inch diameter drill is allowed to cut its full diameter 0.005 inch past the rear edge of the cut off tool and allowing an approach of 0.010 inch, the total travel amounts to 0.280 inches.

Operation 4 : Cone

The bottom of the 0.209 inch diameter drilled hole must be rough coned in preparation to reaming. The travel for this operation will be 0.125 inch plus 0.010 inch approach and is equal to 0.135 inches of total throw.

Operation 5 : Ream

To ream the 0.218 inch diameter hole, the throw will be 0.531 inch (17/32) minus 1/32 chamfer plus 0.010 inch approach which equals to 0.510 total throw.

Operation 6 : Cut off

In this example, the throw of the cut off tool has been reduced by the forming operation behind the head of the nipple, thereby reducing the diameter and by allowing the 3/32 inch diameter drill to cut past (0.005 inch into the next piece).

Operation 7 : Form

This operation is overlapped with the drilling and reaming operation and hence will not enter into the cycle time claculation. The throw, feed, revolutions etc. will still be required for drawing the cam , nevertheless. The total throw in this case is 0.095 inch plus 0.005 inch approach wich equals 0.100 inches.

Step 4 : Feed per revolution of workspindle

Similar to cutting speeds, feeds too have to be selected according to limits as set out in tables. The feeds depend again on properties of the material, the type of component and the type of operation. There are no hard and fast rules in feed selection for optimum cutting conditions.

The selected feeds can be seen in Appendix IV. It must be mentioned that since the higher the feed rate, the shorter the cycletime, so operations that are overlapped need not high feeds. This will help increase the life of the tool without increasing the cycletime.

Step 5 : Revolutions of workspindle per operation

To obtain the revolutions of workspindle per operation, simply divide the feed per revolution into the throw :

Travel or Throw of tool/Feed per revolution = Revolution per operation

The results for each operation is entered in the appropriate column in table 5.1.

Note : Overlapped operations have a special column for revolutions for each operation

To obtain revolutions of workspindle per operation for idle operations, the specifications chart provided by the manufacturers of the semi-automatic machine must be consulted. For example for a B.S.A 48 short stroke it takes 1/4 second to index or feed stock. Therefore the number of revolutions required is equal to

RPM * Time / 60 + 5

And in this case works out to be 26 i.e

4995 * 0.25 / 60 + 5

The complete list of revolutions of workspindle per operation is given in Table 5.1

Step 6 : Hundredths required for dwells and clearances

When the cutting tool completes its cutting operation it is allowed to dwell to ensure a clean surface and a uniform length or diameter. Usually 1 hundredth of the cam surface is sufficient for this purpose.

Before an approximate cycletime can be obtained a chart giving the number of hundredths required for a range of clearances must be consulted.(provided by the machines manufacturers)

Step 7 : Deciding approximate cycletime

Disregard overlapped operations and add together all the revolutions required for cutting and indexing, which for this example is 5/16 revs. Add togather hundredths required for dwells and clearances, in this case 7.5 hundredths, deduct from 100 to obtain the percentage that 5/16 revs represents of the total revs to make one piece.

Total cutting and indexing revs = 516

Total hundredths for dwells and clearances = 7.5

Therefore, 100 - 7.5 = 92.5

92.5 percent represents 516 revs

So approximate revolutions per piece

516/92.5 * 100 = 558 revs per piece

Now revolutions of workspindle to make 1 piece is the time to make 1 piece mutilplied by the rpm and divided by 60. i.e

Cycletime = 60 * 558 / 4995

Which is approximately 6.7 seconds

Step 8 : Converting revolutions per operation into hundredths

To find the number of hundredths of the cam surface devoted to each operation, divide the total revs per piece by 100 to obtain revs per hundredth. Then use this value and divide the revolutions of each operation by this value. The answers are entered in table 5.2

To illustrate consider the centering operation.

Total revs per piece / 100 = 583 /100 = 5.83

Revs for centering / revs per hundred

= 29 / 5.83 = 5 hundredths

Note that the total in the column for hundredths must add up to a hundred.

Step 9 : Drawing the cam

In order to get a clear idea of the sequence and overlapping of the different tools, it is advisable to draw all the disc cams required for the job, superimposed on a common centre as shown in Fig 5.3

The steps are :

1) Vertical and horizontle lines drawn on the circle
2) Divide circle into 100 equal parts with the verticle line being 0.

3) Moving in a clockwise direction count 4.5 hundredths (for feed stock) and draw a radial line. Continue in same direction for all operations.

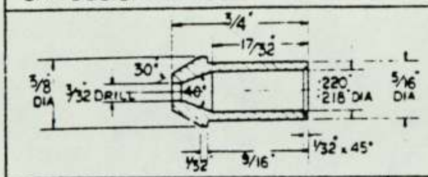
4) Add cut down for each turret to total throw i.e for centre drilling (5/8 + 0.12) 0.745. Mark a point on the 9th hundredth line at a total distance of 0.745 inches from the circumference of the blank. Continue to do so for all operations.

5) Join all such points in a neat manner, avoiding sharp rises and falls whenever possible. The form and cut off cam are developed in the same manner as the lead cam.

The method outlined is only a general guide and depicts a typical case for a brass nipple. Detailed explanations on complex components would take the matter outside the scope of the project but it must be mentioned that detailed explanations can be obtained from many books on semi-automatic machines. The above outline is the main basis for the logic involved in the computer programme.

EXAMPLE No. 1

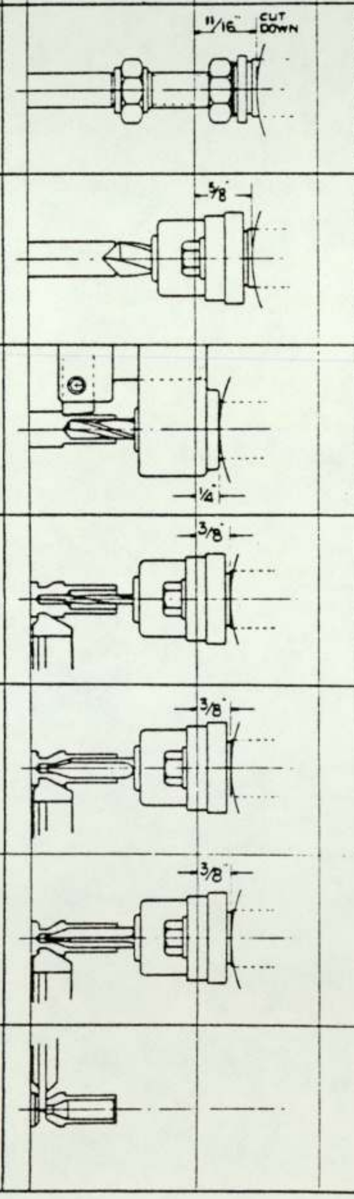
CAM DESIGN WORK SHEET — No. 48 S.S. AUTOMATIC SCREW M/C — SHORT STROKE



WORKSPINDLE SPEEDS R.P.M.			CUTTING SPEEDS F.P.M.	
FAST	R.H.	4995	TURNING	490
	R.H.		DRILLING .209" DIA	273
			DRILLING .093" DIA	121

PRODUCTION TIME – 7 SECS — MATERIAL – BRASS 3/8" DIA

DISTANCE BETWEEN TURRET AND CHUCK: MINIMUM 1 13/16" — TRAVEL 1 3/8"



Station	SEQUENCE OF OPERATIONS	THROW IN INCHES	FEED INCHES PER REV	REVOLUTIONS OF WORKSPINDLE: FOR EACH OPERN	REVOLUTIONS OF WORKSPINDLE: OVER-LAPPED OPERN	CORRECTED	HUNDREDTHS
1	FEED STOCK	DWELL		26		26	4½
	INDEX TURRET (ONE STATION)			26		26	4½
2	CENTRE END OF BAR	.120"	.006"	20		29	5
	DWELL TO CLEAN UP					6	1
	INDEX TURRET (ONE STATION)			26		26	4½
3	TURN 5/16" DIA & DRILL .209" DIA	.570"	.005	114		117	20
	DWELL TO CLEAN UP					6	1
	INDEX TURRET (ONE STATION)			26		26	4½
4	DRILL 3/32" DIA HOLE	.280"	.003"	93		93	16
	DROP BACK TO CLEAR					9	1½
	FORM (OVERLAPPED)	.100"	.0005"		200	(204)	(35)
	DWELL TO CLEAN UP				6		(1)
	INDEX TURRET (ONE STATION)			26		26	4½
5	CONE - BOTTOM OF .209" DIA HOLE	.135"	.003"	45		50	8½
	DWELL TO CLEAN UP					6	1
	INDEX TURRET (ONE STATION)			26		26	4½
6	REAM .218" DIA & CONE	.510"	.010"	51		53	9
	DWELL TO CLEAN UP					6	1
	INDEX TURRET (OVERLAPPED)				26		(4½)
7	CUT OFF REAR CROSS SLIDE	.060"	.002"	30		32	5½
	CUT OFF REAR CROSS SLIDE	.020"	.003"	7		9	1½
	CLEAR CUT-OFF TOOL					12	2
	TOTALS BEFORE CORRECTION	92½%		516			7½
	TOTALS AFTER CORRECTION	92½%		539		583	100

Table 5.1 : Cam Design Chart

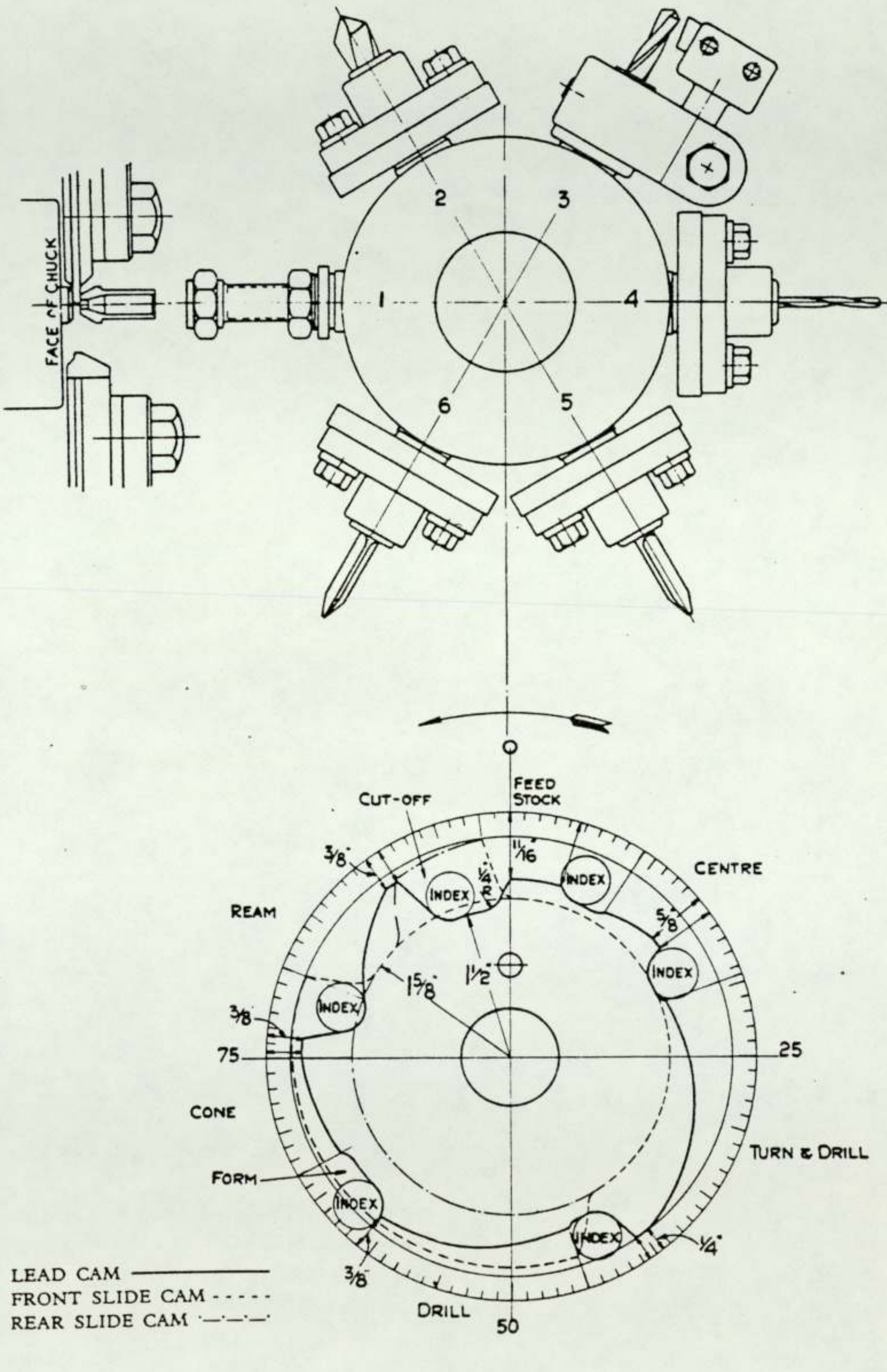


Figure 5.2 : Tool Layout and Drawing of Cam

5.2 : Why Computerize

As mentioned in chapter 3 a component may suffer poor dimensional quality due to poor cam design. Therefore this aspect must be investigated to better the situation if possible.

Another major reason is that of 'time'. When the MD recieves an enquiry from a prospective customer, it would be ideal if a prompt quotation could be made. The situation was such that the MD had to make a rough guess as to how much the component would cost and also how long it would take to complete the batch. More often than not, his estimates were over optimistic and thus led to instability within the system. Instability in the sense that his quotation has to be met to maintain customer goodwill.

To computerize the system thereby obtaining fairly accurate quotes on cost and times would be of great help. Furthermore we must not forget that we are keeping in time with technology when we consider computerization in any field. Much work has already been done in the metal cutting field especially with numerical control utilising computers. Applying computer aided design to semi-automatic machines is one step further into the age of computers.

Hence, a package was written in Fortran that would enable the Managing Director to type in information pertaining to the product and the computer will produce a listing of the necessary information needed for quotation.

5.3 : The Cam Design Package

The cam design package was written mainly to replace the manual task of designing cams that would be 'superior' all round. A 'superior' cam design is one that not only produces the component to specification but one that operates at feeds and speeds and on machines that contributes maximum profit.

The aspect of profit has been considered in the previous chapter and all that has been learnt will now have to be applied in this package. Concerning interaction with the user, the package was written with the intention of minimum input and maximum useful output.

Before designing the package we must decide what output is required. From there we can loyout the steps necessary to produce these outputs. Finally by analysis we can determine the minimum inputs necessary for manipulation. The package was designed with the following outputs in mind :

1) Complete Cam Design
2) Selection of Machines
3) Optimum feeds and cutting speeds
4) Estimated completion time of the batch
5) Estimated Metal requirement for the batch

6) Estimated Tool requirement for the batch

7) Decision Making information

From the above and a knowledge on how to obtain the above outputs, inputs were categorised as follows :

1) Component information

2) Batch information

3) Machine information

4) Tooling information

5) Material information

Note that the inputs are divided into 2 main types -

a) those pertaining to a particular component

b) those pertaining to any component (global)

Details of the inputs and outputs will be given in the next section. Summerising the package we can say that it is a suit of programs that, when given information on a particular component, designs the cam for semi-automatic machines using global data that have been filed. Also available is general information for management.

5.3.1 : The Program

The program written in Fortran, consists of a master segment and 42 subroutine segments. Though written in Fortran, it was set out in a very 'English' like manner so that anyone who understands Fortran will be able to follow the logic of the program. For a listing of the program refer to Appendix V.

The program is heavily overlayed and requires a core size of 6k words. As the program had to be run on an interactive basis on a multiple online programming ICL 1904S machine, the overall size of the program had to be a minimum, as the main memory of the computer is shared amongst several users. Initially the program was 34K words in size and could not be run online, but after extensive analysis and segmentation together with overlays, the program was brought down in size.

5.3.2 : Running the Package

Running the program is a fairly complicated issue to describe. Figure 5.3 should help envisage the interactions involved in the package. A macro was written to help run the program so that we did not have to do repetative things like assigning card readers and lineprinters.

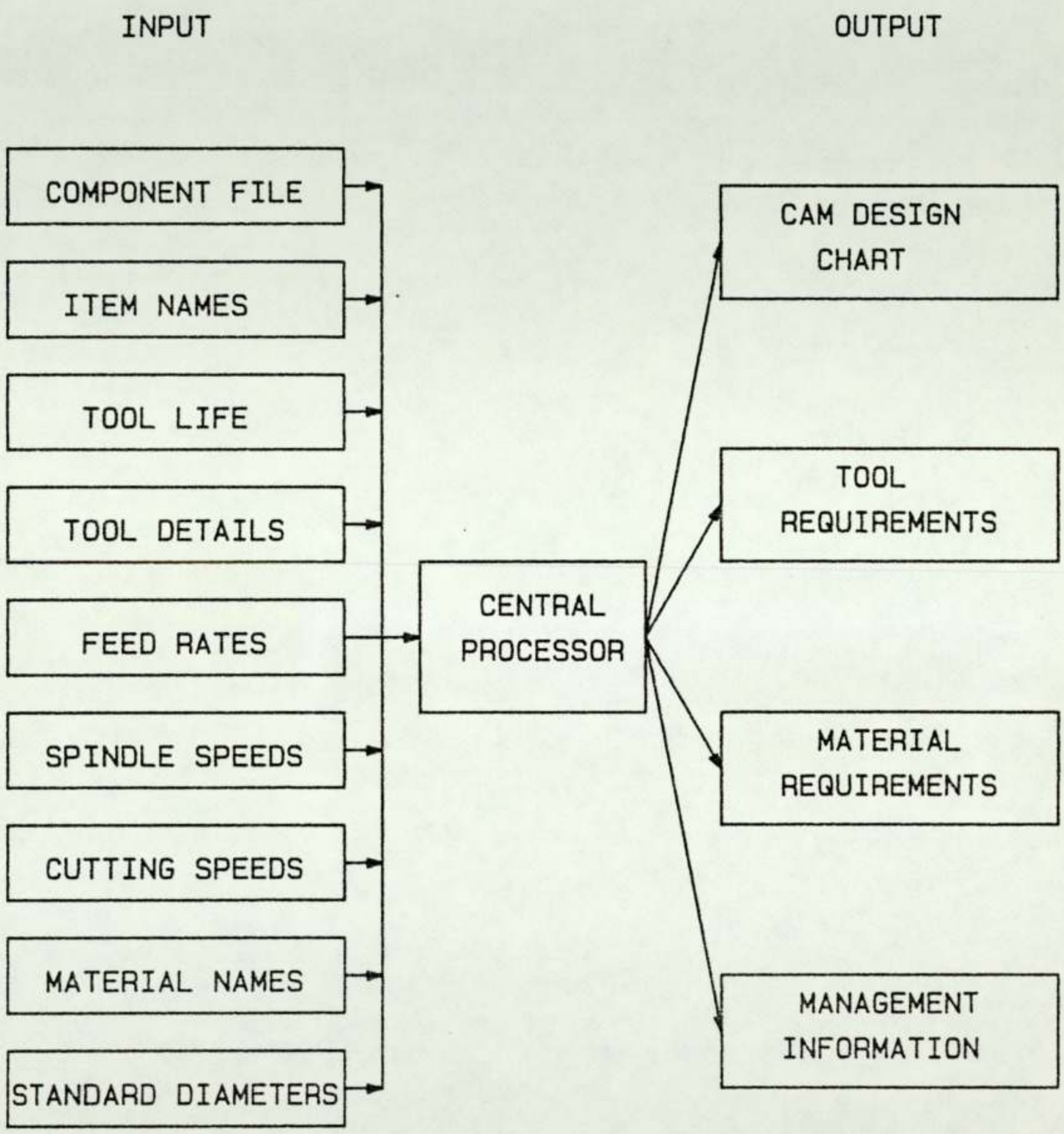


Figure 5.3 : Cam design package transput

The macro requires the following files to be online:

1) Component file (workfile)
2) Item names
3) Tool life
4) Tool names
5) Feed rates
6) Spindle Speeds
7) Cutting Speeds
8) Material Names
9) Standard diameters

Since all the above mentioned files will have to be read, the logical step would be to assign 9 card readers to the files. This would push the size of the program up as each card reader requires 255 words of core space. So to eliminate this problem the macro assigns and releases the card readers as necessary.

From the names of the above files we can guess what it contains, except perhaps for 'Item names'. This file contains the names of the operations that can be handled by this package. File numbers 2 to 9 are global data files and always necessary whatever the component to be produced. Obviously these global data files are dependant on the job shop itself and new ones will have to be set up for different job shops. File number 1, the component file is the file that is different for each component.

The component file contains all the data necessary and is listed out below :

1) Throws
2) Sequence of operations
3) Overall length of the component
4) Overall diameter of the component
5) The batch size
6) The type of speed selection required
7) The type of machine required (override)

Points 1 to 6 refers to the order recieved from the customer and is particular to the component. Point 7 the selection of speeds, is where extensive work has been done to obtain an optimum speed that would result in maximum profit. The 3 method by which speeds can be selected are

1) Assignment
2) Optimization using maximum profit criteria
3) Herberts Technique

Speed selection by assignment is where the user supplies the speed. This speed ideally would be provided by the design department through their experience. This option is basically an override included in the program so that checks can be made to see the effect of a change in speed.

Speed selection by optimization is where the program computes a speed going through the routines based on the maximum profit criteria as discussed in chapter 4. All possible speeds that are available on all possible machines that could produce the component are tested and the optimum speed that produces the maximum profit is used to design the cam. The demand is also considered to ensure that it is met under the operating speeds.

Speed selection by Herberts Technique is where an approximate speed is specified usually by experience. This speed is used as a starting point and speeds are scanned in steps to obtain speeds that are within the cutting range of all operations. No consideration is placed on profit. This technique has been outlined in the earlier, the design by conventional method.

Now for machine selection we can either select a particular machine through previous experience or availability of machines, or we could leave the task to the computer program. If left to the program, it will first select all possible machines that could handle the physical size of the component. Then it would design the cam as if it was going to be run on all the machines i.e all machines are going to be used to produce the batch.

Special notes in the priority listings will be produced so that the M.D can be the final judge as to which machine is to be assigned the job. The priority listing is basically an ascending ordered list of the machines that could handle the job according the following 3 criteria

1) Maximum Profit

2) Earliest Due Date

3) Shortest Time

If for example the MD wanted to maximize his profit on a particular batch, he would chose the cam design that is associated with the machine that is at the top of the list given under criteria 1. If he wanted to finish a batch in the shortest possible time than he would pick the top one from the list given under criteria 3. Now the top one in the list given under criteria 2 should be used if there is a large batch of components to be produced and the completion time is critical. Given in this format taking into consideration the scrap rate, non production rates etc. this list could be very useful.

The other two main items that are not input directly is that of tools and material diameter selection. The tools required is listed easily since the types of operations are known. The quantity required to complete the batch can be computed using tool life equations or substituting actual experimental tool life values. The description on running the program can best be illustrated using an example. Consider the component in figure 5.1.

Data for the component

```
  2.0  0.375  1  21
1  23  6   24  23  2   24  23  8  25
4  24  23  19  24  23  11  23  5  5
25
0.0  0.0  0.25  0.0  0.0  0.375  0.0  0.0  0.09375  0.0
0.3125   0.0   0.0  0.209   0.0   0.0  0.218   0.0  0.3125
0.3125
0.0
  2  0.81
F       4995.0
0.0  0.0  0.120  0.0   0.0  0.570  0.0  0.0  0.280  0.0
 0.100   0.0   0.0  0.135   0.0   0.0  0.510   0.0   0.06
0.020
0.0
0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0
0
                    5000   15.0  25.0
```

Result of Cam Design Program using data from previous page

CAM-DESIGN-CHART

MACHINE NAME : B.S.A 48
MATERIAL NAME : BRASS-RO
MATERIAL SIZE : 0.3750 DIA.
SPINDLE SPEED -
FAST : 4995.0000 RPM.
SLOW : 4995.0000 RPM.
CYCLE TIME : 6.2500 SECS.

FEED-STOCK	0.000	0.000	26.000	0.000	0.000	4.997
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
CENTRE-DRILLING	0.120	0.007	17.000	0.000	0.000	3.267
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
TURNING	0.570	0.013	46.000	0.000	0.000	8.841
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
DRILLING [3-5/16]	0.280	0.005	56.000	0.000	0.000	10.763
CLEAR-STOCK	0.000	0.000	0.000	0.000	0.000	1.000
FORMING	0.100	0.001	71.000	1.000	0.000	13.646
DWELL	0.000	0.000	0.000	1.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
CONING	0.135	0.003	45.000	0.000	0.000	8.649
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
REAMING	0.510	0.013	39.000	0.000	0.000	7.495
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
PARTING-OFF	0.060	0.003	20.000	0.000	0.000	3.844
PARTING-OFF	0.020	0.003	7.000	0.000	0.000	1.345
CLEAR-STOCK	0.000	0.000	0.000	0.000	0.000	1.000

More detailed results are in Appendix V.

5.4 : Conclusion

The package was primarily designed and written to enable management to obtain quick accurate information at the customer enquiry stage. This has been achieved as the program uses 1 millisecond of the 1904s computer time to execute. Furthermore, relevant information is available to make the required committments to the customer to ensure a sale. The basic benefits at this level are :

1) approximate cost per component
2) total production time (for due date specification)

The benefits to the factory system itself include:

1) cam design (Production Controller)
2) Tool requirements (Tool store & management)
3) Material Requirement (Material Store & management)
4) Machine speeds and feeds (Tool setters)

These benefits will allow the staff involved to spend their time more fruitfully rather than doing mundane jobs. It would improve productivity and also improve working standards and goodwill.

Chapter 6 : Scheduling

There have been many atempts to obtain a method of schedule that would be suitable to job shops. The analysis and detailed study of scheduling would take it outside the scope of this project, as a workable system is required. The basic types of loading rules are

FIFO	- first in first out
SI	- select the job with the shortest imminent operation
SJ	- select the job with the shortest total processing time
SR	- select the job with the shortest remaining processing time
SRA	- select the job with the shortest remainihg mean operation time
W	- select the job that, by the end of the next operation, will minimize the total cumulative waiting time of the rest of the queue.
LI	- select the job with the longest imminent operation
LJ	- select the job with the longest total processing time

The method of scheduling chosen is very simple to implement and also easy to maintain. It involves a very basic method of scheduling. The most basic method that comes to mind is one where a job is given a number. This magic number is then used to determine when the job is to be started. Also once the job has started, this same number determines how much resources is made avaiable to the batch until completion.

Let us start by calling this magic number the 'priority rating' of the job. Assume that a job shop has 10 orders to be filled and that 4 of them have already been initiated. Also assume that each of the orders have been given a priority rating. All we need do now is to pick up the job with the highest priority rating and give it maximum resources towards completion. Obviously not all the resources should be spent on a single job, but proportionately divided to ensure a speedy completion. When any job has been completed, the most suitable job from a queue of orders will have to be selected to ensure that machines are not kept idle. The obvious choice would be that with the highest priority rating.

6.1 : Obtaining the Priority Rating

Any order that is accepted can be broken down into the batch size and the component. These two items determine the actual priority rating of the order. The component itself is the main contributing factor to the priority rating whereas a larger batch size only increases the total production run time and requires that much more material.

Consider the component to be produced itself. The main elements that I chose to include in the analysis of priority are as follows.

1) Due Date
2) Operations Remaining
3) Process Time Remaining
4) Critical Order
5) Setup Cost
6) Longest Operation

At this stage I must stress that the breakdown of the component into elements that contribute to priority can be extended infinetly where one could speculate that the time of day or even the weather as affecting the priority of a job.

Due Date

This is a preassigned value as to the expected date that the batch is to be dispatched and is a constant. In computer terms we can always represent time numerically. So as the difference between the due date and the current date gets smaller the 'due date rating' gets numerically larger.

Operations Remaining

This rating mainly affects jobs that have been started. The value not only tells us how close the job is to completion, but also how much time will be necessary for changeover. If there are no more changeovers then the job should be uninterrupted. This rating is useful in producing a better overall schedule in the plant by interrupting orders for complex components.

Process Time Remaining

This rating is similar to that of 'Operations Remaining' but does not consider chaneover time. So the complexity of the component is not in question but the actual time to completion.

Critical Order

This rating can be considered to be a fiddle factor. If for any reason management requires that a particular order should be completed irrespective of the computed priority rating (special customer for example), then this value can be manipulated to give the overall priority rating a boost. In other words this facilitates a manual override.

Setup Cost

This rating helps evaluate in real terms the actual cost of assigning a job. If the setup cost is going to be great then the priority will be lower. One can argue that the cost is not going to change at whatever time we chose to assign the job, but it has been noticed that many orders have similar operations to be performed. In this case considerable money and time can be saved by assigning the correct job to the correct workstation i.e a good schedule.

Longest operation rating

The SI rule (the selection of the job with the shortest imminent operation) is found to be the most effective in reducing the mean job throughput time and delay factor, when the batch size is comparatively small (65).

Hence the longer the operation, the smaller the priority rating.

For all the ratings mentioned above a numerical value of 1000 is assigned. So any order having the maximum priority will have a priority rating of 6000. Furthermore, it must be mentioned that in computing the ratings, a threshold value exists. For example if the due date rating exceeds 175 then we give it the maximum available rating of 1000, indicating that completion is imminent. These threshold values are different for each rating and by fine tuning a good threshold value can be obtained for each.

6.2 Executing the Scheduling Package

The Scheduling package consists of four programs written in Fortran and four macros written in the ICL George 3 operating system language.
They are

1) The schedule program
2) The printout program
3) The output program
4) The Gantt chart program
5) The schedule macros

Although the schedule program is all that is necessary for the scheduling of the orders that have been accepted within the jobshop, the other 3 programs are essential for checking purposes. The progress and cross check are made using these 3 utility type programs.

6.2.1 : The Schedule program

This program is run on the basis that we first decide how long a simulation is required. We input the number of weeks, months and years that we expect the simulation to run.
The program then sets up the machine status and creates an initial set of orders. This is inserted into the master order file and the status is supposed to represent the initial condition before any scheduling is to be done. In real

situations this status is obtained from the current status of the job shop and inserted into the master order file.
Now that the initial condition has been set, we proceede to schedule for a period as specified by weeks, months and years. The scheduling part on the first pass, goes through the routine task of creating a number of random orders for the day.

The random order is made up of the following elements.

1) component type
2) material code
3) batch size
4) priority
5) capable machines
6) actual machine number
7) started pointer
8) ended pointer
9) job number
10) diameter
11) length
12) cycle time
13) longest process time
14) total process time
15) due date

All necessary constraints are taken into consideration and the order generated will be acceptable to the job shop.
After creating the random jobs they are inserted sequentially into the master order file, we go through a routine of finding

out the possible machines that could handle the order. Once we have done this we must then decide the priority of the order. The process of obtaining the priority has been described in the previous section. Once the priority has been obtained we have to interrogate the state of machines. The routine that interrogates the state of machines basically scans the list of machines present and looks for an idle one. As soon as a machine is found another routine is executed to find the order with the highest priority. Obviously the order with the highest priority selected must also be an order that can be filled by the available machine. As soon as one is found, it is assigned to the machine. At this stage the status of the order in the master order file is altered accordingly, by the use of flags. Also the machine history file will be updated.

In this way utilities can later be executed to verify the accuracy and usefullness of the scheduling package.

6.2.2 : The printout program

This program basically prints out all the records in the master order file. This file contains all orders that have been generated by the scheduler. Whether the order has been initiated, completed or even unprocssed, it will remain in this file and hence will be printed out. Certain flags within the record signifies its current status.

The format of the listing produced is

1) component type

2) material code

3) batch size

4) priority

5) capable machines

6) actual machine number

7) started pointer

8) ended pointer

9) job number

10) diameter

11) length

12) cycle time

13) longest process time

14) total process time

15) due date

This master listing is always referred to for verification of the schedule.

6.2.3 : The output program

This program interrogates the master order file and outputs 3 basic lists. The 3 basic lists are

a) processed orders list

b) machine utilization list

c) unprocessed orders list

For the processed orders list, the program scans the whole master order file and selects records where the order has been fully satisfied. The format of listing is under the headings of

a) machine number

b) job number

c) priority

d) start time

e) finish time

An example is given in Appendix VI.

From this list we can confirm that the actual scheduling of the job has taken place. The job number is a sequential number of the job on entry to the master order file and this is an indicator as to the time of arrival of the order.

For the machine utilization list, the program scans the whole machine history file and prints out all the records. Records are entered in this file whenever a machine is selected to fulfil an order. The format of listing is as for the processed orders list above.

From this list we can verify that the correct machines have been assigned the correct order. Furthermore we can also draw conclusions on the actual machine utilization factor. If our simulation is representative of the types of orders that we normally get, then improvements on the machining force can be made by either buying or selling machines.

For the unprocessed orders list, the program scans the whole master order file and prints out all the records where the orders have not been completed. The format of listing is under the headings of

a) job number

b) priority

c) machines

d) due date

An example is given in Appendix VI.

From this list which not only tells us which machines are capable of handling the order, it also tells us when the order is required, i.e the due date. By cross checking we may need to update the priority of the job by utilizing the overide feature in order to get the job started.

6.2.4 : The Gantt chart program

This program produces a Gantt chart of the complete schedule so that we can visualise the actual utilization of the machine shop. The charts, like any other Gantt chart, shows the span of time spent on jobs begining from the start of the schedule for each order that has been accepted.

These Gantt charts are possible because the start and finnish times are stored in the machine history file. An example of the Gantt chart is given in Appendix VI.

6.2.5 : The schedule macro

From the explanation of the various interactions of files, together with the input of the simulation period, it can be seen that the method of running the whole package is a cumbersome task. The schedule macro's task is to simplify the whole process and by just executing the macro, all necessary files and data will be attached to the program. Instances where certain files have to be specified repeatedly can now be mechanised. Similarly, the utilities are also handled in this same manner.

6.3 : Conclusion

From the printouts of the utility programs that have been executed we can see the scheduling package does what it was expected to do. By cross checking we can pinpoint any job and aslo manually go through the process of scheduling and find finally that its expected position is exactly as scheduled by the package.

Now all that remains is to adapt the package to a real world system. This is not too difficult because the initial condition of the jobshop status, as discussed previously, can be input into the master order file and we can start the schedule under any initial condition that we require. Even holidays and other non events can be catered for by this declaration.

Obviously for the final package to be implemented, the scheduling package must be run interactively and many more utilities must be available and run indepenant of the main scheduling process. For example the alteration of an orders priority or even alterations in the basic defination of the order, be it batch size or component description, must be possible. Furthermore, the alterations must be made while the scheduling process is being carried out so that the sense of real time is not lost.

Chapter 7 : Conclusions and Future Work

It can be seen that such work as that carried out in this thesis can be of great use and importance to manufacturing systems. The work carried out in this project was mainly for Job Shops but one can see the similarities to other types of manufacturing systems. The approach used, by basic analysis, can be adhered to in respect to other systems and no doubt the fruits of the work will be equally rewarding.

All the programs were written in Fortran and based on the ICL 1904 s series initially. It would be absurd to suggest that any company requiring the facilities of such programs buy or rent an ICL or mainframe computer.

The ideal hardware for the suit of programs would be a table top micro. For simplicity let as consider the well known Commodore Pet. The minimum configuration required to run the suit of programs except for the scheduling package would be

1) CBM 8032 (Computer)
2) CBM 8050 (Dual Disk Drive)
3) Any printer

The computer and disk drive each cost £900 and a printer could cost anything bewteen £400 and £1000 depending on the quality of printouts required. So for an outlay of say £2750, one could be in the position to benefit from the ever increasing applications of the digital computer.

The problem of cam design can be easily dealt with by the CBM machines. In fact in the appendices, some of the programs are written in basic and executed on CBM machines. The processing power of these small machines are quite unbelievable and I am confident in saying that the cam design package can be handled by these and similar micros. In the March 82 issue of 'Mind Your Own Bussiness', there was a plea by a managing director for a program that would print out the cycletime of a component given the necessary parameters. No doubt that as I am writing this, someone will be in the process of preparing a software package to do the above mentioned task.

The problem of scheduling perhaps requires a more powerful processor but by clever sharing of the processor by utilizing the interrupts, one could get a scheduling type program to work on the CBM machines. The clock time on the CBM machines is 1MHz with a minimum instruction time of 2 microseconds suggests that it might be possible to implement scheduling if the programs were written in machine code. Furthermore the possibility of two seperate processors, assesing a single disk drive unit must not be ruled out. This would enable one processor to carry out the actual scheduling while the acted as a slave and did mundane jobs like order entry, master file updates and so on. This would cause the minimum configuration to include 2 CBM

8032's and would boost the overall price of the system to about £3650, say. This is by no means a high capital outlay on the consideration of hardware cost. The true cost will come in when such a package is required and a team of programmers, including an analyst is approached to provide the software.

Appendix I : Employee Names and Weekly Hours

The names of all employees are given together with the hours that they worked each week for the period between Dec 75 and Dec 76.

Note :

0.00 signifies that the employee was absent
98.76 signifies that the employee was on holiday
67.89 signifies that the employee was not employed at that time

J.FRAMPTON
40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00
40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00
40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00
40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00
40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00 40.00
40.00 40.00 40.00 40.00 40.00 40.00 40.00

C.DOWNES
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 44.00 45.00 49.00
54.50 49.00 53.00 44.00 49.00 44.00 47.00 49.00 44.00
44.00 43.75 44.00 49.00 44.00 43.00 40.00 44.00 44.00
40.75 40.00 40.00 48.00 40.00 40.00 98.76 98.76 40.00
40.00 40.00 40.00 40.00 40.50 41.50 43.25

R.CARTWRIGHT
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 16.00 25.00 38.00 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89

M.GLAZE
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 17.00 42.00 44.00 40.00 44.50 29.00 24.00
28.00 40.00 32.00 44.00 44.00 37.00 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89

J.BRISCOE
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 40.00 32.00 40.00 35.25 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89

G.ROBINSON
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 26.00 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89

R.HARPER
67.89 67.89 67.89 47.50 42.75 45.00 50.00 27.00 27.00
45.00 45.00 45.00 45.00 45.00 36.00 42.00 50.00 40.00
30.50 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89 67.89
67.89 67.89 67.89 67.89 67.89 67.89 67.89

J.MILLS

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	15.00	23.00	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

G.LUDSLY

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	43.75	46.00	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

D.ALLKINS

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	41.75	41.50	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

P.GRIFFIN

45.00	44.75	45.00	45.00	45.00	43.50	45.00	45.00	36.00
45.00	45.00	40.00	45.00	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

S.AZER

45.00	44.25	40.75	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

D.BURKE

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
36.75	46.25	41.25	48.25	43.50	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	39.50	23.75	41.00	45.25	45.75		

P.STANTON

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	23.50	45.00	39.00	20.75	38.00	35.25	38.75
31.25	32.00	67.89	67.89	67.89	67.89	67.89		

S.HAZEL

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	40.00
40.00	98.76	24.00	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

F.MORTON

44.50	44.75	49.25	40.75	35.50	36.50	49.75	45.00	46.00
45.00	44.50	44.25	50.25	44.50	45.00	43.75	49.75	48.00
52.75	44.00	46.75	43.25	43.75	42.75	46.75	48.25	43.75
33.50	43.75	43.25	47.50	42.75	41.00	40.00	41.25	39.75
40.00	37.00	37.75	45.50	29.00	40.00	40.00	37.00	38.25
38.00	39.00	31.25	39.25	38.75	39.25	39.50		

G.GORDON

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	44.00	44.00	48.00	43.00	46.00	47.00	31.00
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

R.MORGAN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
44.50	45.00	41.50	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

D.HALL

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	24.00	40.00	40.00	46.50	36.00		

D.SOLANKI

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
35.00	98.76	30.75	40.00	40.00	40.00	40.00	40.00	40.00
40.00	44.00	32.00	40.00	40.00	40.00	40.00		

D.EVANS

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	18.00	45.00	32.00	40.00
42.50	29.00	39.25	32.00	23.00	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

P.WILLIAMS

45.00	45.00	36.00	45.00	49.75	50.00	49.25	26.75	45.00
36.00	44.50	27.00	44.25	45.00	41.00	45.00	0.00	48.75
30.25	34.25	40.00	44.00	44.75	42.00	46.75	49.00	42.00
44.00	44.00	43.75	43.75	43.75	40.50	32.00	40.25	31.75
39.75	31.75	40.00	42.00	40.00	98.76	98.76	29.00	39.50
39.75	40.00	31.25	39.50	38.75	32.00	39.50		

G.DARKES

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	16.00	36.50	31.50	32.00	40.00
40.00	40.00	31.00	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

B.ST.JUSTIE

67.89	36.00	44.75	43.50	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

S.PHIPPS

44.25	45.00	45.00	45.00	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

K.GRIMMETI

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	44.00	32.00	40.00	40.00	40.00
98.76	98.76	39.25	44.00	31.75	40.00	39.50	0.00	0.00
8.00	32.00	43.00	40.25	31.75	41.00	44.00		

J.CARDEN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	32.00
40.00	40.00	42.50	40.00	0.00	0.00	0.00		

R.KHAN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
15.25	40.00	98.76	98.76	30.75	40.00	0.00	0.00	0.00
0.00	0.00	0.00	4.75	0.00	0.00	0.00		

S.AHMED

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
9.00	45.00	45.00	52.25	45.00	39.00	27.00	18.25	48.00
50.75	47.00	44.00	44.00	17.00	44.00	47.00	45.00	44.00
44.50	48.50	44.00	44.50	44.50	43.50	40.00	40.00	40.00
40.00	40.00	40.00	41.75	40.00	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

M.SHAFIQ

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
34.00	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

A.J.KHAN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
9.75	24.75	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

M.SHAIK

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	39.00	46.25	11.75
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

G.SINGH

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	27.00	2.75
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

S.HUSSAIN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	44.75	45.00	10.00	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

A.KHAN

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	40.50	44.75	45.00	45.00	36.00	0.00
42.00	17.00	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
24.00	40.00	40.00	45.00	40.00	40.00	40.00	40.00	40.00
38.50	40.00	43.00	40.00	40.00	40.00	44.00		

S.KHAN

45.00	45.00	50.00	45.00	50.00	45.00	45.25	27.00	27.00
45.00	45.00	45.00	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

K.MORTON

18.00	44.75	49.25	44.75	44.50	44.75	49.75	27.00	11.50
27.00	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

A.JEFFREY

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	24.00	35.00
40.00	32.00	40.00	24.00	0.00	40.00	40.00		

K.EAGER

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	33.00
39.75	39.75	40.00	40.00	40.00	40.00	39.75	40.00	19.75
5.00	67.89	67.89	67.89	67.89	67.89	67.89		

P.KIRBY

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	40.00	40.00	40.00	40.00	40.00	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

C.HUBBAND

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	32.00	39.00
41.00	40.75	41.00	32.00	32.00	38.50	32.00	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89		

S.EAGER

67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89	67.89
67.89	67.89	67.89	67.89	67.89	39.50	32.50	51.25	42.25
31.50	8.00	33.00	40.00	43.75	41.00	38.00	29.00	32.00
32.00	98.76	24.25	39.00	32.00	39.25	39.50	31.75	40.00
32.00	39.75	40.00	31.50	39.50	39.50	39.00		

Appendix II : Dynamic Programming and Results

```
      PROGRAM (DYNAMIC)
      INPUT  1 = CR1
      OUTPUT 2 = LP2
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 0
      END

      MASTER  RAJ
      REAL    PENALTY MATRIX(50,10),INITIAL COMPUTATION(50)
      REAL    FINAL COMPUTATION(50),NO LINK / 98.76 /
      REAL    INFINITY / 1.0 E 50 /
      INTEGER CONNECTION MATRIX(50,10),NUMBER OF LINKS(50)
      INTEGER OPTIMAL PATH(50),VERTEX
      READ(1,38) LAST NODE,MAXIMUM LINKS
      READ(1,38) (NUMBER OF LINKS(NODE),NODE = 1,LAST NODE)
      WRITE(2,41)
      DO 28 NODE = 1,LAST NODE
      READ(1,38) (CONNECTION MATRIX(NODE,LINK),LINK = 1 ,
     ^            MAXIMUM LINKS)
      WRITE(2,40) (CONNECTION MATRIX(NODE,LINK),LINK = 1 ,
     ^             MAXIMUM LINKS)
   28 CONTINUE
      WRITE(2,42)
      DO 29 NODE = 1,LAST NODE
      READ(1,38)  (PENALTY MATRIX(NODE,LINK),LINK = 1 ,
     ^             MAXIMUM LINKS)
      WRITE(2,40) (PENALTY MATRIX(NODE,LINK),LINK = 1 ,
     ^             MAXIMUM LINKS)
   29 CONTINUE
      VERTEX = LAST NODE + 1
   30 VERTEX = VERTEX - 1
      DO 33 LINK = 1,NUMBER OF LINKS(VERTEX)
      NODE = CONNECTION MATRIX(VERTEX,LINK)
      NEXT LINK = 1
   31 IF (CONNECTION MATRIX(NODE,NEXT LINK) .EQ. VERTEX) GOTO32
      NEXT LINK = NEXT LINK + 1
      GOTO31
   32 IF (PENALTY MATRIX(NODE,NEXT LINK) .EQ. NO LINK) GOTO33
      INITIAL COMPUTATION(NODE)= INITIAL COMPUTATION(VERTEX) +
     ^                          PENALTY MATRIX(NODE,NEXT LINK)
   33 CONTINUE
      IF (VERTEX .NE. 1) GOTO30
   34 DO 35 VERTEX = 1,LAST NODE - 1
      FINAL COMPUTATION(VERTEX) = INFINITY
      DO 35 LINK = 1,NUMBER OF LINKS(VERTEX)
      NODE = CONNECTION MATRIX(VERTEX,LINK)
      IF (PENALTY MATRIX(VERTEX,LINK) .EQ. NO LINK) GOTO35
      IF ((INITIAL COMPUTATION(NODE) + PENALTY MATRIX(VERTEX ,
     ^       LINK)) .GT. FINAL COMPUTATION(VERTEX)) GOTO35
      FINAL COMPUTATION(VERTEX) = INITIAL COMPUTATION(NODE) +
     ^                            PENALTY MATRIX(VERTEX,LINK)
      OPTIMAL PATH(VERTEX) = NODE
   35 CONTINUE
```

```
      DO 37 VERTEX = 1,LAST NODE
      IF (FINAL COMPUTATION(VERTEX) .EQ.
     ^INITIAL COMPUTATION(VERTEX)) GOTO37
      DO 36 VERTEX = 1,LAST NODE
36    INITIAL COMPUTATION(VERTEX) = FINAL COMPUTATION(VERTEX)
      GOTO34
37    CONTINUE
      WRITE(2,39) ((NODE,OPTIMAL PATH(NODE),FINAL COMPUTATION
     ^                (NODE)),NODE = 1,LAST NODE)
      STOP 'END OF DYNAMICITY RAJ'
38    FORMAT(10 G0.0)
39    FORMAT(T5,'GO FROM      TO',// (5X,3(G10.5,3X) /))
40    FORMAT(T5,10 (G10.5) /)
41    FORMAT(T5,'THE CONNECTION MATRIX IS AS FOLLOWS -'//)
42    FORMAT(//,T5,'THE PENALTY MATRIX IS AS FOLLOWS -'//)
      END

      FINISH
```

Connection Matrix

17	6								
3	4	4	4	6	6	6	6	6	6
6	6	6	4	4	4	3			
2	3	4	0	0	0				
1	5	6	7	0	0				
1	5	6	7	0	0				
1	5	6	7	0	0				
2	3	4	8	9	10				
2	3	4	8	9	10				
2	3	4	8	9	10				
5	6	7	11	12	13				
5	6	7	11	12	13				
5	6	7	11	12	13				
8	9	10	14	15	16				
8	9	10	14	15	16				
8	9	10	14	15	16				
11	12	13	17	0	0				
11	12	13	17	0	0				
11	12	13	17	0	0				
14	15	16	0	0	0				

Penalty Matrix

12.50	4.50	4.00	0.00	0.00	0.00
98.76	1.36	1.50	2.00	0.00	0.00
98.76	1.36	1.50	2.00	0.00	0.00
98.76	3.00	3.50	4.00	0.00	0.00
98.76	98.76	98.76	0.84	1.60	2.40
98.76	98.76	98.76	1.20	2.40	3.00
98.76	98.76	98.76	2.00	2.50	3.50
98.76	98.76	98.76	0.60	0.84	1.00
98.76	98.76	98.76	0.60	0.84	1.00
98.76	98.76	98.76	0.80	1.20	1.50
98.76	98.76	98.76	1.25	2.00	2.50
98.76	98.76	98.76	1.25	2.00	2.50
98.76	98.76	98.76	1.25	2.00	2.50
98.76	98.76	98.76	0.20	0.00	0.00
98.76	98.76	98.76	0.20	0.00	0.00
98.76	98.76	98.76	0.10	0.00	0.00
98.76	98.76	98.76	0.00	0.00	0.00

Results

GO	FROM	TO	COST
	1	3	8.7500
	2	5	4.2500
	3	5	4.2500
	4	5	5.8900
	5	8	2.8900
	6	8	3.2500
	7	8	4.0500
	8	11	2.0500
	9	11	2.0500
	10	11	2.2500
	11	14	1.4500
	12	14	1.4500
	13	14	1.4500
	14	17	0.2000
	15	17	0.2000
	16	17	0.1000
	17	0	0.0000

Appendix III : Labour Information Program and Results

```
      PROGRAM (LABOUR INFO)
      INPUT  1 = CR1
      INPUT  2 = CR2
      OUTPUT 3 = LP3
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 0
      END

      MASTER  RAJ
      COMMON / PERSON / EMPLOYEE NAME(2)
      REAL    LEAST ABSENT(2),MOST ABSENT(2)
      REAL    LEAST INDUSTRIOUS(2),MOST INDUSTRIOUS(2)
      REAL    WEEKLY HOURS(52),LOST TIME RATE
      REAL    HOLIDAY / 98.76 /,UNEMPLOYED / 67.89 /
      REAL    LOW ABSENTEE / 5.6E50 /,HIGH ABSENTEE / -5.6E50 /
      REAL    LOW HOURS / 5.6E50 /,HIGH HOURS / -5.6E50 /
      INTEGER WEEK,EPISODES,TOTAL EPISODES,GUILTY
      WRITE(3,36)
      DO 35 MAN = 1,48
      READ(1,37) (EMPLOYEE NAME(I),I = 1,2)
      READ(2,38) (WEEKLY HOURS(WEEK),WEEK = 1,52)
      HOURS,ABSENCE = 0.0
      EPISODES = 0
      DO 30 WEEK = 1,52
      IF (WEEKLY HOURS(WEEK) .EQ. HOLIDAY  .OR.
     ^     WEEKLY HOURS(WEEK) .EQ. UNEMPLOYED  ) GOTO30
      IF (WEEKLY HOURS(WEEK) .LT. 40.00) ABSENCE = ABSENCE +
     ^   (40.00 - WEEKLY HOURS(WEEK))
      IF (WEEKLY HOURS(WEEK) .LT. 32.00) EPISODES = EPISODES+1
      HOURS = HOURS+WEEKLY HOURS(WEEK)
   30 CONTINUE
      PERCENTAGE ABSENCE = ABSENCE / (HOURS+ABSENCE) * 100.0
      WRITE(3,39) (EMPLOYEE NAME(I),I = 1,2),EPISODES ,
     ^              HOURS,ABSENCE,PERCENTAGE ABSENCE
      IF (LOW ABSENTEE .LT. ABSENCE) GOTO31
      LOW ABSENTEE = ABSENCE
      CALL TRANSFER NAME (LEAST ABSENT)
   31 IF (HIGH ABSENTEE .GT. ABSENCE) GOTO32
      HIGH ABSENTEE = ABSENCE
      CALL TRANSFER NAME (MOST ABSENT)
   32 IF (LOW HOURS .LT. HOURS) GOTO33
      LOW HOURS = HOURS
      CALL TRANSFER NAME (LEAST INDUSTRIOUS)
   33 IF (HIGH HOURS .GT. HOURS) GOTO34
      HIGH HOURS = HOURS
      CALL TRANSFER NAME (MOST INDUSTRIOUS)
   34 IF (EPISODES .GT. 1) GUILTY = GUILTY+1
      TOTAL ABSENCE = TOTAL ABSENCE+ABSENCE
      TOTAL HOURS = TOTAL HOURS+HOURS
      TOTAL EPISODES = TOTAL EPISODES+EPISODES
   35 CONTINUE
      LOST TIME RATE = TOTAL ABSENCE / (TOTAL ABSENCE +
     ^                                 TOTAL HOURS)
```

```
    AVERAGE LENGTH OF SPELL = TOTAL ABSENCE / TOTAL EPISODES
    SPELL OF ABSENCE = TOTAL EPISODES / (TOTAL ABSENCE +
   ^                                            TOTAL HOURS)
    P A I A O P = GUILTY / (TOTAL ABSENCE+TOTAL HOURS)
    WRITE(3,40) (LEAST ABSENT(I),I = 1,2),LOW ABSENTEE
    WRITE(3,41) (MOST ABSENT(I),I = 1,2),HIGH ABSENTEE
    WRITE(3,42) (LEAST INDUSTRIOUS(I),I = 1,2),LOW HOURS
    WRITE(3,43) (MOST INDUSTRIOUS(I),I = 1,2),HIGH HOURS
    WRITE(3,44)   TOTAL HOURS,TOTAL ABSENCE,TOTAL EPISODES
    WRITE(3,45) LOST TIME RATE,AVERAGE LENGTH OF SPELL ,
   ^            SPELL OF ABSENCE,P A I A O P
    STOP ' END OF LABOUR INFO RAJ'
36  FORMAT(//,1X,'INFORMATION FOR MANAGEMENT',/// ,
   ^             1X ,'EMPLOYEE NAME    ',T18,'EPSDS' ,
   ^             T25,'HOURS',T34,'ABSENCE' ,
   ^             T43,'%TAGE' //)
37  FORMAT(20X,2A8)
38  FORMAT(10G0.0)
39  FORMAT(T1,2A8,T18,1X,I2,T25,F7.2,T34 ,
   ^         F7.2,T43,F5.2)
40  FORMAT(/,T2,'LEAST ABSENT WORKER IS ',2A8,
   ^'AND WAS ABSENT FOR',2X,F8.2,' HOURS')
41  FORMAT(/,T2,'MOST ABSENT WORKER IS ', 2A8,
   ^'AND WAS ABSENT FOR',2X,F8.2,' HOURS')
42  FORMAT(/,T2,'LEAST INDUSTRIOUS WORKER IS ',2A8 ,
   ^         2X,'AND HE WORKED ' ,F8.2,' HOURS')
43  FORMAT(/,T2,'MOST INDUSTRIOUS WORKER IS ',2A8 ,
   ^         2X,'AND HE WORKED ',F8.2,' HOURS')
44  FORMAT(/,T2,'HOURS WORKED IN YEAR IS ' ,F9.2 ,/,
   ^          T2,'HOURS ABSENT IN YEAR IS ' ,F9.2 ,/,
   ^         T2,'TOTAL EPISODES OF ABSENCE IS ' ,I8)
45  FORMAT(1X,'LOST TIME RATE = ',F10.7,/ ,
   ^         1X,'AVERAGE LENGTH OF SPELL = ',F7.2,/ ,
   ^         1X,'SPELL OF ABSENCE = ',F10.7,/ ,
   ^         1X,'P A I A O P = ',F10.7)
    END

    SUBROUTINE TRANSFER NAME (NAME)
    COMMON / PERSON / EMPLOYEE NAME(2)
    REAL   NAME(2)
    DO 30 N = 1,2
30  NAME(N) = EMPLOYEE NAME(N)
    RETURN
    END

    FINISH
```

INFORMATION FOR MANAGEMENT

EMPLOYEE NAME	EPSDS	HOURS	ABSENCE	%TAGE
J.FRAMPTON	0	2080.00	0.00	0.00
C.DOWNES	0	1541.25	0.00	0.00
R.CARTWRIGHT	2	79.00	41.00	34.17
M.GLAZE	4	465.50	73.00	13.56
J.BRISCOE	0	147.25	12.75	7.97
G.ROBINSON	1	26.00	14.00	35.00
R.HARPER	3	662.75	39.50	5.62
J.MILLS	2	38.00	42.00	52.50
G.LUDSLY	0	89.75	0.00	0.00
D.ALLKINS	0	83.25	0.00	0.00
P.GRIFFIN	0	569.25	4.00	0.70
S.AZER	0	130.00	0.00	0.00
D.BURKE	1	411.25	20.00	4.64
P.STANTON	3	303.50	61.50	16.85
S.HAZEL	1	104.00	16.00	13.33
F.MORTON	2	2201.25	50.75	2.25
G.GORDON	1	303.00	9.00	2.88
R.MORGAN	0	131.00	0.00	0.00
D.HALL	1	186.50	20.00	9.69
D.SOLANKI	1	581.75	22.25	3.68
D.EVANS	3	300.75	66.75	18.16
P.WILLIAMS	8	1982.75	145.25	6.83
G.DARKES	3	267.00	53.00	16.56
B.ST.JUSTIE	0	124.25	4.00	3.12
S.PHIPPS	0	179.25	0.00	0.00
K.GRIMMETI	5	630.50	145.75	18.78
J.CARDEN	3	194.50	128.00	39.69
R.KHAN	12	130.75	429.25	76.65
S.AHMED	4	1302.50	89.75	6.45
M.SHAFIQ	0	34.00	6.00	15.00
A.J.KHAN	2	34.50	45.50	56.87
M.SHAIK	1	97.00	29.25	23.17
G.SINGH	2	29.75	50.25	62.81
S.HUSSAIN	1	99.75	30.00	23.12
A.KHAN	3	904.75	84.50	8.54
S.KHAN	2	514.25	26.00	4.81
K.MORTON	4	361.25	76.50	17.48
A.JEFFREY	3	275.00	85.00	23.61
K.EAGER	2	377.00	63.00	14.32
P.KIRBY	0	200.00	0.00	0.00
C.HUBBAND	0	328.25	34.50	9.51
S.EAGER	6	1000.75	137.50	12.08
M.KHAN	0	1005.00	0.00	0.00
K.STYLES	2	28.50	51.50	64.37
C.BROOKE	1	266.50	35.00	11.61
P.SCANLON	2	245.00	71.00	22.47
S.ROSE	0	879.50	46.50	5.02
S.CHALLENGER	1	24.00	16.00	40.00

LEAST ABSENT WORKER IS M.KHAN AND WAS ABSENT FOR 0.00 HOURS
MOST ABSENT WORKER IS R.KHAN AND WAS ABSENT FOR 429.25 HOURS
LEAST INDUSTRIOUS WORKER IS S.CHALLENGER AND HE WORKED 24.00 HOURS
MOST INDUSTRIOUS WORKER IS F.MORTON AND HE WORKED 2201.25 HOURS
HOURS WORKED IN YEAR IS 21951.25
HOURS ABSENT IN YEAR IS 2375.50
TOTAL EPISODES OF ABSENCE IS 92
LOST TIME RATE = 0.0976497
AVERAGE LENGTH OF SPELL = 25.82
SPELL OF ABSENCE = 0.0037818
P A I A O P = 0.0009455

```
10000  REM=== PROGRAM TO OBTAIN THE FREQUENCY
10020  REM=== AND CUMULATIVE FREQUENCY AND
10040  REM=== FINALLY THE PROBABILITY OF
10060  REM=== ABSENTEE BY THE LABOUR FORCE.
10080  REM=== AN OPTIMUM RESERVE CREW CAN
10100  REM=== THEN BE SPECIFIED.
10120  REM
10140  REM===   WRITTEN : 22/10/80, R NARAYANAN
10160  REM===   AMENDED : 27/04/82, R NARAYANAN
10180  REM

11000  REM=== DECLARATIONS
11020  REM
11040  CT=5:
       TP=8:
       SM=100:
       C=0:
       WK=0:
       NA=0:
       CS=0:
       PS=0:
       IN=1:
       SI=0
11060  DEF FNXA(X)=INT(FNX(XO)+X*10*XS):
       DEF FNX(X)=INT(X*10)
11080  DEF FNYA(Y)=INT(FNY(YO)+Y*10*YS):
       DEF FNY(Y)=INT(Y*10)
11100  MO=16:
       PL=255:
       PE=254:
       OPEN PL,4:
       OPEN PE,6
11120  XS=120/16:
       YS=75/100
11140  DIM AH(52),PH(MO),FQ(MO),CF(MO),CA(MO),PR(MO),DF(MO)
11160  OPEN IN,8,2,"0:lab-econdata,s,r"
11180  ES$=CHR$(27)
11200  CZ$=ES$+".b"
11220  PL$=ES$+".(":
       GOSUB 14000

11240  REM--PLOTTER INITIALIZATION
       PL$="in;vs12,7;sp5;":
       GOSUB 13000:
11260  PL$="ip0,0,16000,11400;":
       GOSUB 13000
```

```
11280  REM--SCALE TO MILLIMETRES
       PL$="sc80,4080,50,2900;":
       GOSUB 13000:

11300  REM--HEIGHT
       PL$="si.167,.25;":
       GOSUB 13000:
11320  GOTO 20000

13000  REM==SEND INSTRUCTION TO GRAPH PLOTTER
13020  REM
13040  PRINT #PL,CZ$;:
       INPUT #PE,BZ:
       IF BZ<LEN(PL$)+10 GOTO 13040
13060  PRINT #PL,ES$+".e":
       INPUT #PE,EM:
       IF EM<>0 THEN
          PRINT "error ="EM

14000  REM==SEND INSTRUCTION WITHOUT CHECKING SIZE
14020  REM
14040  PRINT #PL,PL$;:
       RETURN

15000  REM==TERMINATION
15020  REM

15040  REM--KEEP PEN
       PL$="pu;sp0;pa4080,2900;":
       GOSUB 13000:

15060  REM--PLOTTER OFF
       PL$=ES$+".)":
       GOSUB 14000:
15080  CLOSE PL:
       CLOSE PE:
       PRINT "[3bell]":
       RETURN

16000  REM=== PLOT THE AXES
16020  REM
16040  REM  AXES
16060  PL$="pa"+STR$(FNX(X0))+","+STR$(FNYA(100))+";pd;
16080  PL$=PL$+"pa"+STR$(FNX(X0))+","+STR$(FNY(Y0))+","
16100  PL$=PL$+STR$(FNXA(16))+","+STR$(FNY(Y0))
16120  PL$=PL$+";pu;":
       GOSUB 13000
16140  REM  X TICKS
16160  PL$="tl0,.5;":
       GOSUB 13000
16180  FOR V=1 TO 16:
          PL$="pa"+STR$(FNXA(V-.5))+","+STR$(FNY(Y0))+";xt;"
16200     PL$=PL$+"pa"+STR$(FNXA(V-.5)-30)+","+
          STR$(FNY(Y0-5))+";"
```

```
16220       PL$=PL$+"lb"+STR$(V-1)+"[stop]pu;"
16240       GOSUB 13000:
        NEXT V
16260   REM  Y TICKS
16280   PL$="tl0,.375;":
        GOSUB 13000
16300   FOR V=0 TO 100 STEP 10:
            PL$="pa"+STR$(FNX(X0))+","+STR$(FNYA(V))+";yt;"
16320       PL$=PL$+"pa"+STR$(FNX(X0-10))+","+
            STR$(FNYA(V)-15)+";"
16340       PL$=PL$+"lb"+RIGHT$("   "+STR$(V),3)+"[stop]pu;"
16360       GOSUB 13000:
        NEXT V
16380   REM  LABELS
16400   PL$="pa"+STR$(FNX(X0)+(FNXA(16)-FNX(X0)-
        LEN(XL$)*25)/2)+","
16420   PL$=PL$+STR$(FNY(Y0-15))+";lb"+XL$+"[stop]pu;":
        GOSUB 13000
16440   PL$="pa"+STR$(FNX(X0-15))+","
16460   PL$=PL$+STR$(FNY(Y0)+(FNYA(100)-FNY(Y0)-
        LEN(YL$)*25)/2)+";"
16480   PL$=PL$+"di0,1;lb"+YL$+"[stop]pu;di1,0;":
        GOSUB 13000
16500   RETURN

17000   REM=== DRAW HISTOGRAM BOX
17020   REM
17040   IF Y=0 THEN
            RETURN
17060   PL$="pa"+STR$(FNXA(X-.95))+","+STR$(FNY(Y0))+";pd;"
17080   PL$=PL$+"pa"+STR$(FNXA(X-.95))+","+STR$(FNYA(Y))+","
17100   PL$=PL$+STR$(FNXA(X-.05))+","+STR$(FNYA(Y))+","
17120   PL$=PL$+STR$(FNXA(X-.05))+","+STR$(FNY(Y0))+";pu;":
        GOSUB 13000
17140   RETURN

20000   REM=== MAIN PROGRAM
20020   REM
20040   PRINT "[scr.clr]Operator"TAB(20)"Number of";
20060   PRINT TAB(40)"   Cost"TAB(60)"Number"
20080   PRINT "  Type"TAB(20)"Reserves";
20100   PRINT TAB(40)"Associated"TAB(60)"Employed[cur.dwn]"
20120   FOR C=1 TO CT
20140       REM READ LABOUR DATA
            GOSUB 21000:
20160       REM CALCULATE FREQUENCY
            GOSUB 22000:
20180       REM CALCULATE CUMULATIVE FREQUENCY
            GOSUB 23000:
20200       REM SIMULATE LABOUR SITUATION
            GOSUB 24000:
20220       REM ECONOMICS OF LABOUR
            GOSUB 25000:
20240       REM PLOT THE CURVES
```

```
          GOSUB 26000:
20260     IF C=CT GOTO 20340
20280     PRINT #PL,CZ$;:
          INPUT #PE,BZ:
          IF BZ<927 GOTO 20280
20300     PRINT "[3bell]Press any key when ready
20320     POKE 158,0:
          WAIT 158,15,0:
          PRINT "[cur.up][srl.dwn][cur.up]
20340  NEXT C
20360  DCLOSE
20380  GOSUB 15000:
       END

21000  REM===  READ LABOUR DATA
21020  REM
21040  REM NUMBER OF OPERATORS
       READ NO:
21060  FOR WK=1 TO 52:
          FOR PS=1 TO NO:
             INPUT #IN,PH(PS):
          NEXT :
          NA=0
21080     FOR PS=1 TO NO
21100        IF PH(PS)=67.89 THEN 21140
21120        IF 40-PH(PS)>TP THEN
                NA=NA+1
21140     NEXT PS:
          AH(WK)=NA:
       NEXT WK:
       RETURN

22000  REM=== CALCULATE FREQUENCY
22020  REM
22040  FOR NA=1 TO MO:
          FQ(NA)=0:
       NEXT NA
22060  FOR NA=1 TO NO+1:
          FOR WK=1 TO 52
22080        IF AH(WK)=NA-1 THEN
                FQ(NA)=FQ(NA)+((1/52)*100)
22100     NEXT WK:
       NEXT NA:
       RETURN

23000  REM=== CALCULATE CUMULATIVE FREQUENCY
23020  REM
23040  CS=0:
       FOR NA=1 TO NO+1:
          CS=CS+FQ(NA)
23060     CF(NA)=CS:
       NEXT NA:
       RETURN
```

```
24000  REM=== OBTAIN PROBABILITY FUNCTION
24020  REM
24040  FOR NA=1 TO MO:
          PR(NA)=0:
       NEXT NA
24060  FOR SI=1 TO SM:
          FOR NA=1 TO NO+1
24080        IF CF(NA)=>RND(TI)*100 THEN 24120
24100     NEXT NA:
          GOTO 24140
24120     PR(NA)=PR(NA)+(1/SM)
24140  NEXT SI:
       RETURN

25000  REM=== ECONOMIC RESERVE CREW
25020  REM
25040  FOR NA=1 TO MO:
          DF(NA)=0:
       NEXT NA
25060  REM OPERATOR TYPE, OPERATOR COST, IDLE MACHINE LOSS
       READ OP$,OT$,OC,IL:
25080  HU=1E38
25100  FOR NS=1 TO NO+1:
          FOR NA=NS TO NO+1
25120        CA(NA)=(NA-NS)*IL*TP+DF(NA)
25140        CL=CL+(CA(NA)*PR(NA)):
          NEXT NA
25160     SC=OC*(NS-1)
25180     TC=CL+SC
25200     IF TC<HU THEN
             NR=NS-1:
             HU=TC
25220     CL=0:
          SC=0:
          TC=0
25240     FOR NA=1 TO MO:
             CA(NA)=0:
          NEXT NA
25260  NEXT NS
25280  PRINT OT$;TAB(23)NR;TAB(43)HU;TAB(63)NO
25300  RETURN

26000  REM=== PLOT THE CURVES
26020  REM
26040  X0=60:
       Y0=175
26060  YL$="frequency":
       XL$="no of "+OP$+" absent ":
       GOSUB 16000
26080  FOR X=1 TO NO+1:
          Y=FQ(X):
          GOSUB 17000:
       NEXT
26100  X0=60:
```

```
      Y0=65
26120 YL$="cumulative frequency":
      XL$="no of "+OP$+" absent":
      GOSUB 16000
26140 FOR X=1 TO NO+1:
         Y=CF(X):
         GOSUB 17000:
      NEXT
26160 Y0=175:
      PL$="pa"+STR$(FNXA(10))+","+STR$(FNYA(90))+";"
26180 PL$=PL$+"lbno employed = "+STR$(NO)+"[stop]pu;":
      GOSUB 13000
26200 PL$="pa"+STR$(FNXA(10))+","+STR$(FNYA(90)-FNY(5))+";"
26220 PL$=PL$+"lbstandbys    = "+STR$(NR)+"[stop]pu;":
      GOSUB 13000
26240 PL$="sl.2618;pa"+STR$(FNX(50))+","+STR$(FNY(30))+";"
26260 PL$=PL$+"lbfIGURE 3."+MID$(STR$(5+C),2)+" : "
26280 PL$=PL$+OT$+" ABSENTEE; =CHARTS[stop]pu;sl0;":
      GOSUB 13000
26300 RETURN

63000 REM=== DATA SECTION
63020 REM
63040 DATA 15,"tool setters","tOOL sETTERS",2,14
63060 DATA 5,"bar feeders","bAR fEEDERS",1.25,15
63080 DATA 8,"machine operators","mACHINE oPERATORS",1,0
63100 DATA 14,"general workers","gENERAL wORKERS",1,0
63120 DATA 6,"inspectors","iNSPECTORS",1.5,5
63999 REM SCRATCH"lab-econ",D0:DSAVE"lab-econ",D0:VERIFY"*",8
```

CATEGORY	EMPLOYED	RESERVES	COST
TOOL SETTERS	15	1	3.59
BAR FEEDERS	5	1	2.15
MACHINE OPERATORS	8	0	0.00
GENERAL WORKERS	14	0	0.00
INSPECTORS	6	1	1.50

Appendix IV : Metal Cutting Theory programs

```
10000  REM=== PROGRAM TO PLOT THE CURVES
10020  REM===  PROFIT AGAINST VELOCITY
10040  REM=== AND
10060  REM=== DEMAND AGAINST VELOCITY
10120  REM
10140  REM===   WRITTEN : 28/04/82, R NARAYANAN
10160  REM===   AMENDED : 28/04/82, R NARAYANAN
10180  REM
11000  REM=== DECLARATIONS
11020  REM
11040  TC=1:
       TH=1:
       CO=0.08:
       CT=.5:
       D=3:
       L=6:
       N=0.125:
       M=0.67:
       K=6.5:
       IU=1
11060  PI=3.14159263
11080  DEF FNXA(X)=INT(FNX(XO)+X*10*XS):
       DEF FNX(X)=INT(X*10)
11100  DEF FNYA(Y)=INT(FNY(YO)+Y*10*YS):
       DEF FNY(Y)=INT(Y*10)
11120  PL=255:
       PE=254:
       OPEN PL,4:
       OPEN PE,6
11140  P1=0.98571429:
       P2=0.05714286:
       P3=-0.08571429:
       P4=0.77142857
11160  P5=0.34285714:
       P6=-0.22857143:
       P7=0.48571429:
       P8=-0.01428571
11180  DIM X(350),Y(350),X1(350)
11200  ES$=CHR$(27)
11220  CZ$=ES$+".b"
11240  PL$=ES$+".(":
       GOSUB 14000

11260  REM--PLOTTER INITIALIZATION
       PL$="in;cs2;vs12,7;sp1;":
       GOSUB 13000:
11280  PL$="ip0,0,16000,11400;":
       GOSUB 13000

11300  REM--SCALE TO MILLIMETRES
       PL$="sc80,4080,50,2900;":
       GOSUB 13000:

11320  REM--HEIGHT
```

```
       PL$="si.167,.25;":
       GOSUB 13000:
11340  GOTO 20000

13000  REM==SEND INSTRUCTION TO GRAPH PLOTTER
13020  REM
13040  PRINT #PL,CZ$;:
       INPUT #PE,BZ:
       IF BZ<LEN(PL$)+10 GOTO 13040
13060  PRINT #PL,ES$+".e":
       INPUT #PE,EM:
       IF EM<>0 THEN
          PRINT "error ="EM

14000  REM==SEND INSTRUCTION WITHOUT CHECKING SIZE
14020  REM
14040  PRINT #PL,PL$;:
       RETURN

15000  REM==TERMINATION
15020  REM

15040  REM--KEEP PEN
       PL$="pu;sp0;pa4080,2900;":
       GOSUB 13000:

15060  REM--PLOTTER OFF
       PL$=ES$+".)":
       GOSUB 14000:
15080  CLOSE PL:
       CLOSE PE:
       PRINT "[3bell]":
       RETURN

16000  REM=== PLOT THE AXES
16020  REM
16040  REM  AXES
16060  PL$="pa"+STR$(FNX(X0))+","+STR$(FNYA(YI))+";pd;
16080  PL$=PL$+"pa"+STR$(FNX(X0))+","+STR$(FNY(Y0))+","
16100  PL$=PL$+STR$(FNXA(XI))+","+STR$(FNY(Y0))
16120  PL$=PL$+";pu;":
       GOSUB 13000
16140  REM  X TICKS
16160  PL$="t10,.5;":
       GOSUB 13000
16180  FOR V=1 TO XI STEP TX:
          PL$="pa"+STR$(FNXA(V-1))+","+STR$(FNY(Y0))+";xt;"
16200     PL$=PL$+"pa"+STR$(FNXA(V-1)-30)+","+
          STR$(FNY(Y0-5))+";"
16220     PL$=PL$+"1b"+STR$(V-1)+"[stop]pu;"
16240     GOSUB 13000:
       NEXT V
16260  REM  Y TICKS
16280  PL$="t10,.375;":
```

```
      GOSUB 13000
16300 FOR V=0 TO YI STEP TY:
         PL$="pa"+STR$(FNX(X0))+","+STR$(FNYA(V))+";yt;"
16320    PL$=PL$+"pa"+STR$(FNX(X0-17.5))+","+
         STR$(FNYA(V)-15)+";"
16340    PL$=PL$+"lb"+STR$(INT(V))+"."+MID$(STR$(V-INT(V)+1)+
         "000000",4,3)+"[stop]pu;"
16360    GOSUB 13000:
      NEXT V
16380 REM  LABELS
16400 PL$="pa"+STR$(FNX(X0)+(FNXA(XI)-FNX(X0)-
      LEN(XL$)*25)/2)+","
16420 PL$=PL$+STR$(FNY(Y0-15))+";lb"+XL$+"[stop]pu;":
      GOSUB 13000
16440 PL$="pa"+STR$(FNX(X0-20))+","
16460 PL$=PL$+STR$(FNY(Y0)+(FNYA(YI)-FNY(Y0)-
      LEN(YL$)*25)/2)+";"
16480 PL$=PL$+"di0,1;lb"+YL$+"[stop]pu;di1,0;":
      GOSUB 13000
16500 RETURN

18000 REM=== PLOT A CURVE GIVEN NP
18020 REM
18040 NP=NP-1
18060 X1(0)=P1*X(0)+P2*(X(1)+X(3))+P3*X(2)+P8*X(4)
18080 X1(1)=P2*(X(0)+X(4))+P4*X(1)+P5*X(2)+P6*X(3)
18100 FOR V=2 TO NP-2:
         X1(V)=P3*(X(V-2)+X(V+2))+P5*(X(V-1)+X(V+1))+P7*X(V):
      NEXT V
18120 X1(NP-1)=P2*(X(NP-4)+X(NP))+P6*X(NP-3)+P5*X(NP-2)+
      P4*X(NP-1)
18140 X1(NP)=P8*X(NP-4)+P2*(X(NP-3)+X(NP-1))+P3*X(NP-2)+
      P1*X(NP)
18160 PL$="pu;pa"+STR$(X(0))+","+STR$(Y(0))+";pd;":
      GOSUB 13000
18180 FOR V=1 TO NP:
         PL$="pa"+STR$(X1(V))+","+STR$(Y(V))+";":
         GOSUB 13000
18200 NEXT V:
      PL$="pu;":
      GOTO 13000

20000 REM=== MAIN PROGRAM
20020 REM
20030 REM PROFIT RESPONSE
      GOSUB 20500:
      GOSUB 25000:
20040 REM PROFIT RESPONSE
      GOSUB 20500:
      GOSUB 30000:
20060 REM PROFIT SENSITIVITY
      GOSUB 20500:
      GOSUB 35000:
20080 REM DEMAND VOLUME
```

```
       GOSUB 20500:
       GOSUB 40000:
20100  GOSUB 15000:
       END

20500  REM=== WAIT
20520  REM
20540  PRINT #PL,CZ$;:
       INPUT #PE,BZ:
       IF BZ<927 GOTO 20540
20560  PRINT "[3bell]Press any key when ready
20580  POKE 158,0:
       WAIT 158,15,0:
       PRINT "[cur.up][srl.dwn][cur.up]
20600  RETURN

25000  REM=== PROFIT REGION
25020  REM
25040  XI=351:
       YI=.021:
       TX=50:
       TY=.001:
       XS=100/XI:
       YS=150/YI
25060  X0=60:
       XL$="cutting speed v,sfpm"
25080  Y0=75:
       YL$="feed f, ipr"
25100  GOSUB 16000

26000  REM=== MINIMUM COST CURVE
26020  REM
26040  PRINT "Plotting minimum cost curve"
26060  NP=0
26080  FOR FD=20 TO 1 STEP -1:
           F=FD*0.001
26100      V=K/((F^M)*((1/N-1)*(TC+CT/CO))^N)
26120      IF V>350 THEN 26160
26140      X(NP)=FNXA(V):
           Y(NP)=FNYA(F):
           NP=NP+1
26160  NEXT FD
26180  GOSUB 18000
26200  PL$="lt2,.5;pa"+STR$(X(2))+","+STR$(Y(2))+";pd;"
26220  PL$=PL$+"pa"+STR$(FNXA(200))+","+STR$(Y(1))+";"
26240  PL$=PL$+"lb minimum cost[stop]pu;lt;"
26260  GOSUB 13000

27000  REM=== MAXIMUM PRODUCTION RATE VMAX
27020  REM
27040  PRINT "Plotting maximum production rate curve"
27060  NP=0
27080  FOR FD=20 TO 1 STEP -1:
           F=FD*0.001
```

```
27100      V=K/((F^M)*((1/N-1)*TC)^N)
27120      IF V>350 THEN 27160
27140      X(NP)=FNXA(V):
           Y(NP)=FNYA(F):
           NP=NP+1
27160   NEXT FD
27180   GOSUB 18000
27200   PL$="1t2,.5;pa"+STR$(X(3))+","+STR$(Y(3))+";pd;"
27220   PL$=PL$+"pa"+STR$(FNXA(200))+","+STR$(Y(2))+";"
27240   PL$=PL$+"1b maximum production[stop]pu;1t;"
27260   GOSUB 13000

28000   REM=== ZERO PROFIT BOUNDARIES
28020   REM
28040   PRINT "Plotting zero profit boundaries"
28060   NP=0
28080   FOR FD=20 TO 1 STEP -1:
           F=FD*0.001
28100      FOR V=1 TO 350 STEP 2
28120         C1=(CO*PI*D*L)/(12*V*F)
28140         C2=(PI*D*L*V^(1/N-1)*F^(M/N-1))/(12*K^(1/N)) *
                 (CO*TC+CT)
28160         C3=CO*TH
28180         CU=C1+C2+C3
28200         T1=(PI*D*L)/(12*V*F)
28220         T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
28240         TU=T1+T2+TH
28260         P=(IU-CU)/TU
28280         IF P>0 THEN 28320
28300      NEXT V:
           GOTO 28340
28320      X(NP)=FNXA(V):
           Y(NP)=FNYA(F):
           NP=NP+1
28340   NEXT FD
28360   GOSUB 18000
28361   PL$="1t2,.5;pa"+STR$(X(4))+","+STR$(Y(4))+";pd;"
28362   PL$=PL$+"pa"+STR$(FNXA(200))+","+STR$(Y(3))+";"
28363   PL$=PL$+"1b zero profit[stop]pu;1t;"
28364   GOSUB 13000

28380   REM=== FROM RIGHT TO LEFT
28400   NP=0
28420   FOR FD=20 TO 1 STEP -1:
           F=FD*0.001
28440      FOR V=350 TO 1 STEP -2
28460         C1=(CO*PI*D*L)/(12*V*F)
28480         C2=(PI*D*L*V^(1/N-1)*F^(M/N-1))/(12*K^(1/N))
                 *(CO*TC+CT)
28500         C3=CO*TH
28520         CU=C1+C2+C3
28540         T1=(PI*D*L)/(12*V*F)
28560         T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
```

```
28580          TU=T1+T2+TH
28600          P=(IU-CU)/TU
28620          IF P>0 THEN 28660
28640       NEXT V:
            GOTO 28680
28660       X(NP)=FNXA(V):
            Y(NP)=FNYA(F):
            NP=NP+1
28680    NEXT FD
28700    GOSUB 18000
28720    PL$="1t2,.5;pa"+STR$(X(5))+","+STR$(Y(5))+";pd;"
28740    PL$=PL$+"pa"+STR$(FNXA(200))+","+STR$(Y(3))+";pu;1t;"
28780    GOSUB 13000

29000    REM=== MAXIMUM PROFIT VP
29020    REM
29040    PRINT "Plotting maximum profit curve"
29060    NP=0
29080    FOR FD=20 TO 1 STEP -1:
            F=FD*0.001:
            PM=-1E38
29100       FOR V=1 TO 350 STEP 2
29120          C1=(CO*PI*D*L)/(12*V*F)
29140          C2=(PI*D*L*V^(1/N-1)*F^(M/N-1))/(12*K^(1/N))
                  *(CO*TC+CT)
29160          C3=CO*TH
29180          CU=C1+C2+C3
29200          T1=(PI*D*L)/(12*V*F)
29220          T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
29240          TU=T1+T2+TH
29260          P=(IU-CU)/TU
29280          IF P>PM THEN
                  PM=P:
                  VM=V
29300       NEXT V
29320       X(NP)=FNXA(VM):
            Y(NP)=FNYA(F):
            NP=NP+1
29340    NEXT FD
29360    GOSUB 18000
29361    PL$="1t2,.5;pa"+STR$(X(6))+","+STR$(Y(6))+";pd;"
29362    PL$=PL$+"pa"+STR$(FNXA(200))+","+STR$(Y(4))+";"
29363    PL$=PL$+"1b maximum profit[stop]pu;1t;"
29364    GOSUB 13000
29380    PL$="s1.2618;pa"+STR$(FNX(50))+","+STR$(FNY(30))+";"
29400    PL$=PL$+"1bfIGURE 4.1 : pROFIT[ ]REGION[stop]s10;pu;"
29420    GOSUB 13000
29440    RETURN

30000    REM=== PROFIT RESPONSE
30020    REM
30040    PRINT "Plotting profit response"
30060    XI=151:
         YI=.105:
```

```
      TX=25:
      TY=.01:
      XS=100/XI:
      YS=150/YI
30080 X0=60:
      XL$="cutting speed v,sfpm"
30100 Y0=75:
      YL$="profit #/min"
30120 GOSUB 16000
30140 FOR FD=20 TO 8 STEP -1:
         F=FD*0.001:
         NP=0:
         PM=-1E38
30160    FOR V=1 TO 150 STEP 2
30180       C1=(CO*PI*D*L)/(12*V*F)
30200       C2=(PI*D*L*V^(1/N-1)*F^(M/N-1))/(12*K^(1/N))
               *(CO*TC+CT)
30220       C3=CO*TH
30240       CU=C1+C2+C3
30260       T1=(PI*D*L)/(12*V*F)
30280       T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
30300       TU=T1+T2+TH
30320       P=(IU-CU)/TU:
            IF P<0 THEN 30380
30340       IF P>PM THEN
               PM=P:
               VM=V
30360       X(NP)=FNXA(V):
            Y(NP)=FNYA(P):
            NP=NP+1
30380    NEXT V
30400    GOSUB 18000
30420    IF FD<>INT(FD/5)*5 THEN 30540
30440    PL$="lt2,.5;pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM))+
            ";pd;"
30460    PL$=PL$+"pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM+TY))+";"
30480    PL$=PL$+"pa"+STR$(FNXA(VM+5))+","+STR$(FNYA(PM+TY))+
            ";pu;lt;"
30500    PL$=PL$+"lb F ="+STR$(F)+"[stop]pu;"
30520    GOSUB 13000
30540 NEXT FD
30560 PL$="sl.2618;pa"+STR$(FNX(50))+","+STR$(FNY(30))+";"
30580 PL$=PL$+"lbfIGURE 4.2 : pROFIT[ ]RESPONSE[stop]sl0;pu;"
30600 GOSUB 13000
30620 RETURN

35000 REM=== PROFIT SENSITIVITY
35020 REM
35040 PRINT "Plotting profit sensitivity"
35060 XI=101:
      YI=.105:
      TX=25:
      TY=.01:
      XS=100/XI:
```

```
       YS=150/YI
35080  X0=60:
       XL$="cutting speed v,sfpm"
35100  Y0=75:
       YL$="profit #/min"
35120  GOSUB 16000
35140  NP=0:
       F=0.018:
       PM=-1E38
35160  FOR V=1 TO 100
35180     C1=(CO*PI*D*L)/(12*V*F)
35200     C2=(PI*D*L*V^(1/N-1)*F^(M/N-1))/(12*K^(1/N))
             *(CO*TC+CT)
35220     C3=CO*TH
35240     CU=C1+C2+C3
35260     T1=(PI*D*L)/(12*V*F)
35280     T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
35300     TU=T1+T2+TH
35320     P=(IU-CU)/TU:
          IF P<0 THEN 35380
35340     IF P>PM THEN
             PM=P:
             VM=V
35360     X(NP)=FNXA(V):
          Y(NP)=FNYA(P):
          NP=NP+1
35380  NEXT V
35400  GOSUB 18000
35420  PL$="pr"+STR$(FNX(3))+","+STR$(FNY(3))+
           ";lbF = 0.018[stop]pu;"
35440  GOSUB 13000
35460  PL$="lt2,1;pa"+STR$(FNXA(0))+","+STR$(FNYA(PM))+";pd;"
35480  PL$=PL$+"pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM))+";"
35500  GOSUB 13000
35520  PL$="pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM))+";"
35540  PL$=PL$+"pa"+STR$(FNXA(VM))+","+STR$(FNYA(0))+";pu;lt;"
35560  GOSUB 13000
35580  PL$="pr"+STR$(FNX(3))+","+STR$(FNY(3))+";lbvP[stop]pu;"
35600  GOSUB 13000
35620  PL$="sl.2618;pa"+STR$(FNX(50))+","+STR$(FNY(30))+";"
35640  PL$=PL$+"lbfIGURE 4.3 : pROFIT SENSITIVITY[stop]sl0;pu;"
35660  GOSUB 13000
35680  RETURN

40000  REM=== DEMAND VOLUME
40020  REM
40040  PRINT "Plotting demand volume"
40060  XI=201:
       YI=.21:
       TX=25:
       TY=.02:
       XS=100/XI:
       YS=150/YI
40080  X0=60:
```

```
       XL$="cutting speed v,sfpm"
40100  Y0=75:
       YL$="demand volume w, pcs/min"
40120  GOSUB 16000
40140  FOR FD=20 TO 8 STEP -1:
          F=FD*0.001:
          NP=0:
          PM=-1E38
40160     FOR V=1 TO 200 STEP 2
40180        T1=(PI*D*L)/(12*V*F)
40200        T2=(PI*D*L*V^(1/N-1)*F^(M/N-1)*TC)/(12*K^(1/N))
40220        TU=T1+T2+TH
40240        W=1/TU:
             IF W<0 THEN 40300
40260        IF W>PM THEN
                PM=W:
                VM=V
40280        X(NP)=FNXA(V):
             Y(NP)=FNYA(W):
             NP=NP+1
40300     NEXT V
40320     GOSUB 18000
40340     IF FD<>INT(FD/5)*5 THEN 40460
40360     PL$="lt2,.5;pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM))+
              ";pd;"
40380     PL$=PL$+"pa"+STR$(FNXA(VM))+","+STR$(FNYA(PM+TY))+";"
40400     PL$=PL$+"pa"+STR$(FNXA(VM+5))+","+STR$(FNYA(PM+TY))+
              ";pu;lt;"
40420     PL$=PL$+"lb F ="+STR$(F)+"[stop]pu;"
40440     GOSUB 13000
40460  NEXT FD
40480  PL$="sl.2618;pa"+STR$(FNX(50))+","+STR$(FNY(30))+";"
40500  PL$=PL$+"lbfIGURE 4.4 : dEMAND[ ]VOLUME[stop]sl0;pu;"
40520  GOSUB 13000
40540  RETURN
63999  REM SCRATCH"profit",D0:DSAVE"profit",D0:VERIFY"*",8
```

Appendix V : Cam Design Program

```
DUMP ON
OVERLAY PROGRAM ( FACTORY )
OVERLAY (  1, 1 ) INITIALIZE COMMON BLOCKS
OVERLAY (  1, 2 ) INTRODUCTION
OVERLAY (  1, 3 ) DIAMETER SELECTION
OVERLAY (  1, 4 ) FEED RATE SELECTION
OVERLAY (  1, 5 ) MACHINE SELECTION
OVERLAY (  1, 6 ) MACHINE LISTING
OVERLAY (  1, 7 ) SPINDLE SPEED SELECTION
OVERLAY (  1, 8 ) DESIGN CAM
OVERLAY (  1, 9 ) ESTIMATE COMPLETION TIME
OVERLAY (  1,10 ) PRIORITY
OVERLAY (  1,11 ) COMPUTE MATERIAL INFORMATION
OVERLAY (  1,12 ) COMPUTE TOOL INFORMATION
OVERLAY (  2, 1 ) VERIFY COMPONENT INFORMATION
OVERLAY (  2, 2 ) VERIFY SEQUENCE OF OPERATIONS
OVERLAY (  2, 3 ) VERIFY OPERATING DIAMETERS
OVERLAY (  2, 4 ) ACROSS FEED RATE TABLE
OVERLAY (  2, 5 ) DOWN FEED RATE TABLE
OVERLAY (  2, 6 ) VERIFY THROWS
OVERLAY (  2, 7 ) OPTIMIZATION
OVERLAY (  2, 8 ) SPECIAL SPEED ASSIGNMENT
OVERLAY (  2, 9 ) SPEED SELECTION BY HERBERTS METHOD
OVERLAY (  2,10 ) COMPUTE REVS OF WORKSPINDLE
OVERLAY (  2,11 ) COMPUTE CYCLETIME
OVERLAY (  2,12 ) COMPUTE HUNDREDTHS
OVERLAY (  2,13 ) PRINT CAM CHART
OVERLAY (  2,14 ) SORT ASCEND OR DESCEND
OVERLAY (  2,15 ) COMPUTE QUANTITY
OVERLAY (  3, 1 ) COMPUTE PROFIT
OVERLAY (  3, 2 ) HIGHEST CUTTING SPEED
OVERLAY (  3, 3 ) IDEAL CUTTING SPEED
OVERLAY (  3, 4 ) SELECT APPROPRIATE SPINDLE SPEED
OVERLAY (  3, 5 ) CORRECTED TO QUARTERS
OVERLAY (  3, 6 ) COMPUTE TOOL LIFE
OVERLAY (  3, 7 ) COMPUTE CUTTING TIME
OVERLAY (  4, 1 ) ZEROIZE 1D REAL MATRIX
OVERLAY (  4, 2 ) ZEROIZE 2D REAL MATRIX
OVERLAY (  4, 3 ) CLEAR 1D REAL MATRIX
OVERLAY (  4, 3 ) ERROR IN DATA
OVERLAY (  4, 4 ) CLEAR 2D REAL MATRIX
OVERLAY (  4, 5 ) ACROSS CUTTING SPEED TABLE
OVERLAY (  4, 6 ) DOWN CUTTING SPEED TABLE
OVERLAY (  4, 7 ) TYPE OF OPERATION
DEPTH OF OVERLAY 5
INPUT  1 = CR0
OUTPUT 2 = LP0
INPUT  3 = CR1
OUTPUT 4 = LP1
COMPRESS INTEGER AND LOGICAL
COMPACT
TRACE 0
PRIORITY 1
END
```

```
      MASTER  RAJ

      LOGICAL MORE COMPONENTS / .FALSE. /
      PAUSE 'AS *CR1,!'
30    CALL INITIALIZE COMMON BLOCKS
      CALL INTRODUCTION
      CALL DIAMETER SELECTION
      CALL FEED RATE SELECTION
      CALL MACHINE SELECTION (MAJOR)
      CALL MACHINE LISTING (MAJOR)
      CALL SPINDLE SPEED SELECTION (MAJOR)
      CALL DESIGN CAM (MAJOR)
      CALL ESTIMATE COMPLETION TIME (MAJOR)
      CALL PRIORITY (MAJOR)
      CALL COMPUTE MATERIAL INFORMATION
      CALL COMPUTE TOOL INFORMATION (MAJOR)
      READ(3,31,END=99) MORE COMPONENTS
      IF (MORE COMPONENTS) GO TO 30

      STOP 'FACTORY INFORMATION AVAILABLE TO MANAGER'
99    CALL ERROR IN DATA ('OVERCOME LARGE MASTER')
31    FORMAT(L1)

      END

      SUBROUTINE INITIALIZE COMMON BLOCKS

      COMMON / BLOCK1 / REAL1(2),INTEGER1(2)
      COMMON / BLOCK2 / INTEGER2(25), REAL2(25)
      COMMON / BLOCK3 / REAL3(49),INTEGER3
      COMMON / BLOCK4 / REAL4(78)
      COMMON / BLOCK5 / REAL5(110),REAL6(8,5)
      COMMON / BLOCK6 / REAL7(14)
      INTEGER1(1),INTEGER1(2),INTEGER3 = 0
      DO 30 I = 1,25
30    INTEGER2(I) = 0
      CALL ZEROIZE 1D REAL MATRIX (REAL1,2)
      CALL ZEROIZE 1D REAL MATRIX (REAL2,25)
      CALL ZEROIZE 1D REAL MATRIX (REAL3,49)
      CALL ZEROIZE 1D REAL MATRIX (REAL4,78)
      CALL ZEROIZE 1D REAL MATRIX (REAL5,110)
      CALL CLEAR 1D REAL MATRIX (REAL5,75)
      PAUSE 'AS *CR0,:EPP3071.TOOL LIFE'
      READ(1,31,END=99) ((REAL6(I,J),I = 1,8), J = 1,5)
      CALL ZEROIZE 1D REAL MATRIX (REAL7,7)
      DO 32 I = 8,14
32    REAL7(I) = - 5.6 E 76
      RETURN
99    CALL ERROR IN DATA ('INITIALIZE COMMON BLOCKS')
31    FORMAT(20X,8G0.0)

      END

      SUBROUTINE ZEROIZE 1D REAL MATRIX (MATRIX,ROWS)
```

```
      REAL    MATRIX(ROWS)
      INTEGER ROW,ROWS
      DO 30 ROW = 1,ROWS
30    MATRIX(ROW) = 0.0
      RETURN

      END

      SUBROUTINE ZEROIZE 2D REAL MATRIX (MATRIX ,ROWS,COLUMNS)

      REAL    MATRIX(ROWS,COLUMNS)
      INTEGER ROW,ROWS,COLUMN,COLUMNS
      DO 30 ROW    = 1,ROWS
      DO 30 COLUMN = 1,COLUMNS
30    MATRIX(ROW,COLUMN) = 0.0
      RETURN

      END

      SUBROUTINE CLEAR 1D REAL MATRIX (MATRIX,ROWS)

      REAL    MATRIX(ROWS),BLANK / '        ' /
      INTEGER ROW,ROWS
      DO 30 ROW = 1,ROWS
30    MATRIX(ROW) = BLANK
      RETURN

      END

      SUBROUTINE CLEAR 2D REAL MATRIX (MATRIX,ROWS,COLUMNS)

      REAL    MATRIX(ROWS,COLUMNS),BLANK / '        ' /
      INTEGER ROW,ROWS,COLUMN,COLUMNS
      DO 30 COLUMN = 1,COLUMNS
      DO 30 ROW    = 1,ROWS
30    MATRIX(ROW,COLUMN) = BLANK
      RETURN

      END

      SUBROUTINE ERROR IN DATA (SUBROUTINE NAME)

      REAL    SUBROUTINE NAME(4)
      WRITE(2,31) (SUBROUTINE NAME(I),I = 1,4)
      CALL CLEAR 1D REAL MATRIX (SUBROUTINE NAME,4)
      PAUSE 'EN 0'

      STOP 'ERROR IN DATA'
31    FORMAT(' ',//,T5,'ERROR IN READ STATEMENT OF SUBROUTINE '
     ^            ,2X,4A8,//,'PLEASE EDIT NECESSARY FILES AND '
     ^                   ,' TYPE  " %;MESS; " ')

      END
```

```
      SUBROUTINE INTRODUCTION

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      COMMON / BLOCK4 / METAL NAMES(3),OPERATION NAMES(25,3)
      REAL    METAL NAMES
      INTEGER SEQUENT
      PAUSE 'AS *CRO,:EPP3071.ITEM-NAMES'

      READ(1,35,END=99) ((OPERATION NAMES(I,J),J = 1,3) ,
     ^                                          I = 1,25)
      CALL VERIFY COMPONENT INFORMATION
      DO 30 METALS = 1,MATERIAL
   30 READ(1,35,END=99) (METAL NAMES(I),I = 1,3)
      CALL VERIFY SEQUENCE OF OPERATIONS
      WRITE(2,36)TOTAL LENGTH,DIAMETER,(METAL NAMES(J),
     ^J=1,3),(I,(OPERATION NAMES(SEQUENT(I),J),J=1,3),
     ^I=1,LIMIT)
      RETURN
   99 CALL ERROR IN DATA ('INTRODUCTION')
   35 FORMAT(20X,3A8)
   36 FORMAT(' ',3(/),T20,'GENERAL INFORMATION',3(/),
     ^T5,'THE ACTUAL LENGTH OF THE COMPONENT IS ...: ',F6.3,
     ^/,T5,'THE ACTUAL DIAMETER OF THE COMPONENT IS .: ',F6.3,
     ^/,T5,'THE SPECIFIED MATERIAL IS ...............: ',3A8,
     ^/,T5,'THE SEQUENCE OF OPERATIONS ARE  : ',
     ^//,(T5,I2,').....',3A8))

      END

      SUBROUTINE VERIFY COMPONENT INFORMATION

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      READ(3,34,END=99) TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      IF (TOTAL LENGTH .LE. 0.0 .OR.
     ^     DIAMETER .LE. 0.0) STOP 'DIMENSIONLESS COMPONENT'
      IF (MATERIAL .LE. 0 .OR. MATERIAL .GT. 10)
     ^     STOP 'MATERIAL CODE OUTSIDE SCOPE'
      IF (LIMIT .LE. 0) STOP 'OPERATIONLESS COMPONENT'
      IF (LIMIT .GT. 25) STOP 'TOO MANY SEQUENCES'
      RETURN
   99 CALL ERROR IN DATA ('VERIFY COMPONENT INFORMATION')
   34 FORMAT(10G0.0)

      END

      SUBROUTINE VERIFY SEQUENCE OF OPERATIONS

      COMMON / BLOCK1 /  TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 /  SEQUENT(25),DIAMETERS(25)
      INTEGER SEQUENT,SEQUENCE
      READ(3,34,END=99) (SEQUENT(I),I = 1,LIMIT)
      DO 30 SEQUENCE = 1,LIMIT
      IF (SEQUENT(SEQUENCE) .LE. 0 .OR.
```

```
^        SEQUENT(SEQUENCE) .GT. 25)
^        STOP 'SPECIAL OPERATION NOT CONSIDERED'
      IF (SEQUENT(SEQUENCE) .EQ. 24 .AND.
^                  SEQUENCE  .NE. 25 .AND.
^        SEQUENT(SEQUENCE + 1) .NE. 23)
^        STOP 'AFTER DWELL ALWAYS INDEX TURRET'
30    CONTINUE
      IF (SEQUENT(1) .NE. 1) STOP 'FEED STOCK FIRST'
      IF (SEQUENT(LIMIT) .NE. 25) STOP 'CLEAR STOCK LAST'
      RETURN
99    CALL ERROR IN DATA ('VERIFY SEQUENCE OF OPERATIONS')
34    FORMAT(10G0.0)

      END

      SUBROUTINE DIAMETER SELECTION

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / STANDARD SIZES(10)
      INTEGER DIAMETERS
      PAUSE 'AS *CRO,:EPP3071.STANDARD-DIA'
      CALL VERIFY OPERATING DIAMETERS
      DO 31 DIAMETERS = 1,73
      READ(1,34,END=99)(STANDARD SIZES(I),I=1,10)
      IF(STANDARD SIZES(MATERIAL).EQ.0.0)GO TO 33
      IF(DIAMETER.LE.STANDARD SIZES(MATERIAL))GO TO 32
31    CONTINUE
32    DIAMETER=STANDARD SIZES(MATERIAL)
      CALL ZEROIZE 1D REAL MATRIX (STANDARD SIZES,10)
      RETURN
33    WRITE(2,38)
      CALL ZEROIZE 1D REAL MATRIX (STANDARD SIZES,10)
      RETURN
99    CALL ERROR IN DATA ('DIAMETER SELECTION')
34    FORMAT(10G0.0)
38    FORMAT('1',T40,'*************************************'
     ^,//,        T40,'****                          *********'
     ^,//,        T40,'****  NO MATERIAL AVAILABLE   *********'
     ^,//,        T40,'****                          *********'
     ^,//,        T40,'*************************************')

      END

      SUBROUTINE VERIFY OPERATING DIAMETERS

      COMMON / BLOCK1 /  TOTAL LENGTH,DIAMETER ,MATERIAL,LIMIT
      COMMON / BLOCK2 /  SEQUENT(25),DIAMETERS(25)
      INTEGER SEQUENT,SEQUENCE
      LOGICAL NO OPERATION / .TRUE. /
      READ(3,36,END=99) (DIAMETERS(I),I = 1,LIMIT)
      DO 30 SEQUENCE = 1,LIMIT
      IF (DIAMETERS(SEQUENCE) .NE. 0.0) NO OPERATION = .FALSE.
      IF (DIAMETERS(SEQUENCE) .LT. 0.0 .OR.
^        DIAMETERS(SEQUENCE) .GT. DIAMETER)
```

```
^       STOP 'OPERATING DIAMETER OUTSIDE RANGE'
30  CONTINUE
    IF (NO OPERATION) STOP 'NO OPERATING DIAMETERS'
    RETURN
99  CALL ERROR IN DATA ('VERIFY OPERATING DIAMETERS')
36  FORMAT(10G0.0)

    END

    SUBROUTINE FEED RATE SELECTION

    COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
    COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
    COMMON / BLOCK3 / FEED RATES(8,6)
    COMMON / BLOCK5 / CAM CHART(25,6)
    INTEGER ACROSS FEED RATE TABLE
    INTEGER DOWN FEED RATE TABLE
    INTEGER SEQUENT,SEQUENCE,ROW,COLUMN
    PAUSE 'AS *CRO,:EPP3071.FEED RATES'
    READ(1,31,END=99)((FEED RATES(I,J),J=1,6),I=1,8)
    COLUMN = ACROSS FEED RATE TABLE (MATERIAL)
    DO 30 SEQUENCE = 1,LIMIT
    IF(DIAMETERS(SEQUENCE) .EQ. 0.0) GO TO 30
    ROW = DOWN FEED RATE TABLE (SEQUENCE)
    CAM CHART(SEQUENCE,2) = FEED RATES(ROW,COLUMN) / 10000.0
    IF(SEQUENT(SEQUENCE)  .EQ. 12 .OR.
   ^SEQUENT(SEQUENCE) .EQ. 13) CAM CHART(SEQUENCE,2)= 999.999
30  CONTINUE
    CALL ZEROIZE 2D REAL MATRIX (FEED RATES,8,6)
    RETURN
99  CALL ERROR IN DATA ('FEED RATE SELECTION')
31  FORMAT(20X,6G0.0)

    END

    INTEGER FUNCTION ACROSS FEED RATE TABLE (MATERIAL)

    INTEGER COLUMN
    COLUMN = MATERIAL - 4
    IF (MATERIAL .LE. 2) COLUMN = 1
    IF (MATERIAL .GT. 2 .AND. MATERIAL .LT. 6) COLUMN = 2
    IF (MATERIAL .EQ. 6 . OR. MATERIAL .EQ. 7) COLUMN = 3
    ACROSS FEED RATE TABLE = COLUMN
    RETURN

    END
```

```
      INTEGER FUNCTION DOWN FEED RATE TABLE (SEQUENCE)

      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      DATA    SEQUENCES / 3 * 1,2,3,4,4 * 5,6,2 * 1 ,
     ^                    8,3 * 3,1,7,2 * 8,3,3 * 1 /
      INTEGER SEQUENCE,SEQUENT,SEQUENCES(25)
      DOWN FEED RATE TABLE = SEQUENCES(SEQUENT(SEQUENCE))
      RETURN

      END

      SUBROUTINE MACHINE SELECTION (MAJOR)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / MACHINE DATA(7,3),FORCE(7,4),MINOR
      REAL    MAXMACLNGTH,MACHINE DATA
      INTEGER SPECIAL MACHINE
      PAUSE 'AS *CRO,:EPP3071.MACHINE-INFO'
      READ(1,34,END=99)((MACHINE DATA(I,J),J=1,3),I=1,7)
      READ(3,36,END=99) SPECIAL MACHINE,MAXMACLNGTH
      MAXIMUM MACHINE = SPECIAL MACHINE
      IF(SPECIAL MACHINE .EQ. 0) SPECIAL MACHINE = 1
      IF(SPECIAL MACHINE .EQ. 1) MAXIMUM MACHINE = 7
      DO 31 I = SPECIAL MACHINE,MAXIMUM MACHINE
      IF((DIAMETER   .LE. MACHINE DATA(I,1)).AND.
     ^   (MAXMACLNGTH .LE. MACHINE DATA(I,2)).AND.
     ^   (TOTAL LENGTH  .LE. MACHINE DATA(I,3)))GO TO 32
   31 CONTINUE
   33 WRITE(2,35)

      STOP 'JOB ABANDONED : MACHINES'

   32 MINOR = I
      MAJOR = 7
      IF (SPECIAL MACHINE .NE. 1) MAJOR = MINOR
      RETURN
   99 CALL ERROR IN DATA ('MACHINE SELECTION')
   34 FORMAT(20X,3F0.0)
   35 FORMAT('1',T40,'*************************************'
     ^,//,        T40,'*****                       *********'
     ^,//,        T40,'*****  NO  CAPABLE  MACHINES *********'
     ^,//,        T40,'*****                       *********'
     ^,//,        T40,'*************************************')
   36 FORMAT(2G0.0)

      END

      SUBROUTINE MACHINE LISTING (MAJOR)

      COMMON / BLOCK3 / MACHINE NAMES(7,3),FORCE(7,4),MINOR
      REAL    MACHINE NAMES
      PAUSE 'AS *CRO,:EPP3071.MACHINE NAME'
      READ(1,30,END=99) ((MACHINE NAMES(I,J),J = 1,3) ,
     ^                                        I = 1,7)
```

```
      NUMBER = 0
      WRITE(2,31)
      DO 29 MACHINE = MINOR,MAJOR
      NUMBER = NUMBER + 1
   29 WRITE(2,32) NUMBER, (MACHINE NAMES(MACHINE,K),K = 1,3)
      CALL ZEROIZE 2D REAL MATRIX (MACHINE NAMES,7,3)
      RETURN
   99 CALL ERROR IN DATA ('MACHINE LISTING')
   30 FORMAT(20X,3A8)
   31 FORMAT(' ',T5,' LIST OF POSSIBLE MACHINES',2(/))
   32 FORMAT(T26,' MACHINE NO. ',I2,'  ......',2X,3A8)

      END

      SUBROUTINE SPINDLE SPEED SELECTION (MAJOR)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / SPINDLE RPMS(7),FORCE(7,6),MINOR
      COMMON / BLOCK5 / CAM CHART(25,6)
      EQUIVALENCE (SPINDLE RPMS(1),MACHINE NAMES(1))
      REAL    MACHINE NAMES(7,3)
      LOGICAL OPTIMIZE / .FALSE. /
      READ(3,33,END=99) OPTIMIZE,SPECIAL SPEED
      CALL VERIFY THROWS
      IF (.NOT. OPTIMIZE) GO TO 30
      WRITE(2,36)
   29 CALL OPTIMIZATION (MAJOR)
      GO TO 32
   30 IF (SPECIAL SPEED .NE. 0.0) GO TO 31
      WRITE(2,37)
      CALL SPEED SELECTION BY HERBERTS METHOD (MAJOR,& 29)
      GO TO 32
   31 WRITE(2,35)
      CALL SPECIAL SPEED ASSIGNMENT (MAJOR,SPECIAL SPEED)
   32 CALL ZEROIZE 2D REAL MATRIX (FORCE,7,6)
      CALL CLEAR 2D REAL MATRIX (MACHINE NAMES,7,3)
      PAUSE 'AS *CRO,:EPP3071.MACHINE NAME'
      READ(1,38,END=99) ((MACHINE NAMES(I,J),J = 1,3) ,
     ^                                          I = 1,7)
      RETURN
   99 CALL ERROR IN DATA ('SPINDLE SPEED SELECTION')
   33 FORMAT(L1,G0.0)
   35 FORMAT(' ',/,T5,'METHOD OF SPEED SELECTION : ASSIGNMENT')
   36 FORMAT(' ',/,T5,'METHOD OF SPEED SELECTION : OPTIMUM')
   37 FORMAT(' ',//,T5,'METHOD OF SPEED SELECTION : HERBERTS')
   38 FORMAT(20X,3A8)

      END

      SUBROUTINE VERIFY THROWS

      COMMON / BLOCK1 /  TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK5 /  CAM CHART(25,6)
      INTEGER SEQUENCE
```

```
      LOGICAL NO OPERATION / .TRUE. /
      READ(3,31,END=99) (CAM CHART(I,1),I = 1,LIMIT)
      DO 30 SEQUENCE = 1,LIMIT
      IF (CAM CHART(SEQUENCE,1) .NE. 0.0) NO OPERATION=.FALSE.
      IF (CAM CHART(SEQUENCE,1) .LT. 0.0)
     ^      STOP 'NEGATIVE THROW NOT ALLOWED'
      IF (CAM CHART(SEQUENCE,1) .GT. TOTAL LENGTH)
     ^      STOP 'WRONG THROW SPECIFIED'
30    CONTINUE
      IF (NO OPERATION) STOP 'NO OPERATING THROWS'

      RETURN
99    CALL ERROR IN DATA ('VERIFY THROWS')
31    FORMAT(10G0.0)

      END

      SUBROUTINE OPTIMIZATION (MAJOR)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / SPINDLE RPMS(7),FORCE(7,6),MINOR
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      INTEGER SPEED
      PAUSE 'AS *CRO,:EPP3071.SPINDLESPEED'
      DO 33 SPEED   = 1,98
      READ(1,34,END=99)(SPINDLE RPMS(I),I=1,7)
      DO 33 MACHINE = MINOR,MAJOR
      VELOCITY = SPINDLE RPMS(MACHINE)
      IF(VELOCITY .EQ. 0.0) GO TO 33
      CALL COMPUTE PROFIT (VELOCITY,TOTAL PROFIT)
      IF(OPTIMIZED DATA(MACHINE,2).GT.TOTAL PROFIT)GO TO 33
      OPTIMIZED DATA(MACHINE,1) =  VELOCITY
      OPTIMIZED DATA(MACHINE,2) = TOTAL PROFIT
33    TOTAL PROFIT = 0.0
      RETURN
99    CALL ERROR IN DATA ('OPTIMIZATION')
34    FORMAT(10G0.0)

      END

      SUBROUTINE SPECIAL SPEED ASSIGNMENT (MAJOR,SPECIAL SPEED)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / SPINDLE RPMS(7),FORCE(7,6),MINOR
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      CALL SELECT APPROPRIATE SPINDLESPEED(MAJOR,SPECIAL SPEED)
      DO 30 MACHINE = MINOR,MAJOR
      VELOCITY = OPTIMIZED DATA(MACHINE,1)
      IF (VELOCITY.NE.SPECIALSPEED)WRITE(2,31)MACHINE,VELOCITY
      CALL COMPUTE PROFIT (VELOCITY,TOTAL PROFIT)
30    OPTIMIZED DATA(MACHINE,2) = TOTAL PROFIT
      RETURN
31    FORMAT(T5,'SPEED NOT AVAILABLE ON MACHINE :',I2,5X ,
     ^       'NEXT CLOSEST SPEED IS  =  ',F6.1)
```

```
      END

      SUBROUTINE COMPUTE PROFIT (S,T P)

      COMMON / BLOCK1 /  TOTAL LENGTH , DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 /  SEQ(25),CD(25)
      COMMON / BLOCK5 /  CAM(25,4),RN(10),K1(8),M1(8) ,
     ^                         N1(8),C1(8),T1(8)
      REAL P/ 3.1415926 /,IU/ 0.5 /,CO/ 0.04 /,TC/ 1.0 /
      REAL TH/1.0/,L,K,M,N ,K1,M1,N1
      INTEGER SEQ,SQ,OP
      PROFIT = 0.0
      DO 30 SQ = 1,LIMIT
      CALL TYPE OF OPERATION (SQ,OP,& 30)
      D = CD(SQ)
      V = (P * D * S) / 12.0
      F = CAM(SQ,2)
      K = K1(OP)
      L = CAM(SQ,1)
      M = M1(OP)
      N = N1(OP)
      C = C1(OP)
      PDL = P * D * L
      CT1 = (PIE * D * L * V ** (1/N - 1) * F ** (M/N - 1)) /
     ^     (12 * K ** (1/N)) * (CO * TC + CT)
      C2 = ((CO * PDL) / (12 * V * F))
      C3 = CO * TH
      CU = CT1 + C2 + C3
      TM1 = PDL / (12 * V * F)
      T2 = (PIE * D * L * V ** (1/N - 1) * F ** (M / N - 1.0)
     ^     * TC) / (12 * K ** (1/N))
      TU = TM1 + T2 + TH
      PROFIT  = (IU - CU) / TU
      T P = T P + PROFIT
   30 CONTINUE
      RETURN

      END

      SUBROUTINE TYPE OF OPERATION (SEQUENCE,OPERATION,*)

      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      COMMON / BLOCK5 / FORCE(142),TEMP(8)
      INTEGER SEQUENT,SEQUENCE,OPERATION
      IF (DIAMETERS(SEQUENCE) .EQ. 0.0 .OR.
     ^     DIAMETERS(SEQUENCE) .EQ. 999.999) RETURN 1
      DO 30 OPERATION = 1,8
      IF (SEQUENT(SEQUENCE) .EQ. IFIX(TEMP(OPERATION))) RETURN
   30 CONTINUE

      STOP 'UNEXPECTED OPERATION ENCOUNTERED'

      END
```

```
      SUBROUTINE SPEED SELECTION BY HERBERTS METHOD (MAJOR,*)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK3 / SPINDLE RPMS(7),FORCE(7,6),MINOR
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      REAL    PIE / 3.14159263 /
      CALL HIGHEST CUTTING SPEED (HIGHEST)
      CALL IDEAL CUTTING SPEED (HIGHEST,& 31)
      VELOCITY = HIGHEST / (PIE * DIAMETER)
      CALL SELECT APPROPRIATE SPINDLE SPEED (MAJOR,VELOCITY)
      DO 30 MACHINE = MINOR,MAJOR
      VELOCITY = OPTIMIZED DATA (MACHINE,1)
      CALL COMPUTE PROFIT (VELOCITY,TOTAL PROFIT)
   30 OPTIMIZED DATA(MACHINE,2) = TOTAL PROFIT
      RETURN
   31 RETURN 1

      END

      SUBROUTINE HIGHEST CUTTING SPEED (HIGHEST)


      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      COMMON / BLOCK3 / SPINDLE RPMS(7),CUTTING SPEEDS(7,6),
     ^                  MINOR
      INTEGER ACROSS CUTTING SPEED TABLE
      INTEGER DOWN CUTTING SPEED TABLE
      INTEGER SEQUENT,SEQUENCE,ROW,COLUMN
      PAUSE 'AS *CRO,:EPP3071.CUTTINGSPEED '
      READ(1,31,END=99)((CUTTING SPEEDS(I,J),J=1,6),I=1,4)
      HIGHEST = 0.0
      COLUMN = ACROSS CUTTING SPEED TABLE (MATERIAL)
      DO 30 SEQUENCE = 1,LIMIT
      IF ((DIAMETERS(SEQUENCE) .EQ. 0.0) .OR.
     ^    (SEQUENT(SEQUENCE).EQ.11.OR.(SEQUENT(SEQUENCE).GE.14
     ^     .AND. SEQUENT(SEQUENCE) .LE. 16) .OR.
     ^     (SEQUENT(SEQUENCE) .GE. 19
     ^     .AND. SEQUENT(SEQUENCE) .LE.22))) GO TO 30
      ROW = DOWN CUTTING SPEED TABLE (SEQUENCE)
      CUTTING VELOCITY = CUTTING SPEEDS(ROW,COLUMN)
      IF (HIGHEST .LT. CUTTING VELOCITY)
     ^    HIGHEST = CUTTING VELOCITY
   30 CONTINUE
      RETURN
   99 CALL ERROR IN DATA ('HIGHEST CUTTING SPEED')
   31 FORMAT(6G0.0)

      END

      SUBROUTINE IDEAL CUTTING SPEED (HIGHEST,*)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
```

```
      COMMON / BLOCK3 / SPINDLE RPMS(7),CUTTING SPEEDS(7,6),
     ^                  MINOR
      INTEGER ACROSS CUTTING SPEED TABLE
      INTEGER DOWN CUTTING SPEED TABLE
      INTEGER SEQUENT,SEQUENCE,ROW,COLUMN,TIMES
      COLUMN = ACROSS CUTTING SPEED TABLE (MATERIAL)
      DO 32 TIMES = 1,100
      DO 30 SEQUENCE = 1,LIMIT
      IF ((DIAMETERS(SEQUENCE) .EQ. 0.0) .OR.
     ^    (SEQUENT(SEQUENCE).EQ.11.OR.(SEQUENT(SEQUENCE).GE.14
     ^     .AND. SEQUENT(SEQUENCE) .LE. 16) .OR.
     ^     (SEQUENT(SEQUENCE) .GE. 19
     ^     .AND. SEQUENT(SEQUENCE) .LE.22))) GO TO 30
      ROW = DOWN CUTTING SPEED TABLE (SEQUENCE)
      IF (HIGHEST .GT. CUTTING SPEEDS(ROW,COLUMN)) GO TO 31
   30 CONTINUE
      RETURN
   31 IF (HIGHEST .LE. 5.0) GO TO 33
   32 HIGHEST = HIGHEST - 5.0
   33 WRITE(2,34)
      RETURN 1
   34 FORMAT(' ',T5,/,'SPEED SELECTION BY HERBERTS METHOD NOT',
     ^         ' POSSIBLE SO',/,T5,'SPEED SELECTION WILL NOW',
     ^         ' BE DONE BY OPTIMIZATION TECHNIQUE')

      END

      SUBROUTINE SELECT APPROPRIATE SPINDLE SPEED (MAJOR ,
     ^                                  SPINDLE SPEED)

      COMMON / BLOCK3 / SPINDLE RPMS(7),CUTTING SPEEDS(7,6),
     ^                  MINOR
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      INTEGER SPEEDS
      PAUSE 'AS *CRO,:EPP3071.SPINDLESPEED '
      DO 32 SPEEDS = 1,98
      READ(1,34,END=99) (SPINDLE RPMS(I),I = 1,7)
      DO 30 MACHINE = MINOR,MAJOR
      IF ((OPTIMIZED DATA(MACHINE,1) .NE. 0.0) .OR.
     ^    (SPINDLE RPMS(MACHINE) .EQ. 0.0)) GO TO 30
      IF (SPINDLE RPMS(MACHINE) .GE. SPINDLE SPEED)
     ^    OPTIMIZED DATA(MACHINE,1) = SPINDLE RPMS(MACHINE)
   30 CONTINUE
      DO 31 MACHINE = MINOR,MAJOR
      IF (OPTIMIZED DATA(MACHINE,1) .EQ. 0.0) GO TO 32
   31 CONTINUE
      RETURN
   32 CONTINUE
      DO 33 MACHINE = MINOR,MAJOR
      IF (OPTIMIZED DATA(MACHINE,1) .EQ. 0.0)
     ^    OPTIMIZED DATA(MACHINE,1) = SPINDLE RPMS(MACHINE)
   33 CONTINUE
      RETURN
   99 CALL ERROR IN DATA ('SELECT APPROPRIATE SPINDLE SPEED')
```

```
34  FORMAT(7G0.0)

    END

    INTEGER FUNCTION ACROSS CUTTING SPEED TABLE (MATERIAL)

    INTEGER COLUMN
    COLUMN = MATERIAL - 4
    IF (MATERIAL .LE. 2) COLUMN = 1
    IF (MATERIAL .GT. 2 .AND. MATERIAL .LT. 6) COLUMN = 2
    IF (MATERIAL .EQ. 6 . OR. MATERIAL .EQ. 7) COLUMN = 3
    ACROSS CUTTING SPEED TABLE = COLUMN
    RETURN

    END

    INTEGER FUNCTION DOWN CUTTING SPEED TABLE (SEQUENCE)

    COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
    INTEGER SEQUENT,SEQUENCE,ROW
    ROW = 0
    IF ((SEQUENT(SEQUENCE).GE.2.AND.SEQUENT(SEQUENCE) .LE. 5)
   ^    .OR.SEQUENT(SEQUENCE).EQ.17.OR.SEQUENT(SEQUENCE) .EQ.
   ^     18) ROW = 1
    IF (SEQUENT(SEQUENCE) .GE. 6 .AND. SEQUENT(SEQUENCE) .LE.
   ^     10) ROW = 2
    IF (SEQUENT(SEQUENCE) .EQ. 12) ROW = 3
    IF (SEQUENT(SEQUENCE) .EQ. 13) ROW = 4
    DOWN CUTTING SPEED TABLE = ROW
    RETURN

    END

    SUBROUTINE DESIGN CAM (MAJOR)

    COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
    COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
    COMMON / BLOCK3 / MACHINE NAMES(7,3),FORCE(7,4),MINOR
    COMMON / BLOCK5 / CAM CHART(25,6)
    COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
    EQUIVALENCE (CAM CHART(51),ZEROIZE 1(1))
    EQUIVALENCE (CAM CHART(101),ZEROIZE 2(1))
    REAL    MACHINE NAMES ,ZEROIZE 1(25),ZEROIZE 2(50)
    INTEGER SEQUENT
    PAUSE 'AS *CRO,:EPP3071.MACHINE-INFO'
    READ(1,36,END=99) ((FORCE(I,J),J = 1,1),I = 1,7)
    READ(3,35,END=99) (CAM CHART(I,4),I = 1,LIMIT)
    DO 30 MACHINE = MINOR,MAJOR
    CALL ZEROIZE 1D REAL MATRIX (ZEROIZE 1,25)
    CALL ZEROIZE 1D REAL MATRIX (ZEROIZE 2,50)
    CALL COMPUTE REVS OF WORKSPINDLE (MACHINE)
    CALL COMPUTE CYCLETIME (CYCLETIME,REVS PER PIECE,MACHINE)
    CALL COMPUTE HUNDREDTHS (REVS PER PIECE)
30  CALL PRINT CAM CHART (CYCLETIME,MACHINE)
```

```
    WRITE(2,37)
    CALL ZEROIZE 2D REAL MATRIX (FORCE,7,4)
    RETURN
99  CALL ERROR IN DATA ('DESIGN CAM')
35  FORMAT(10G0.0)
36  FORMAT(46X,G0.0)
37  FORMAT(' ',T5,'THE CAM CHARTS ARE IN RESULT FILES')

    END

    SUBROUTINE COMPUTE REVS OF WORKSPINDLE (MACHINE)

    COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
    COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
    COMMON / BLOCK3 / MACHINE DATA(7,7),MINOR
    COMMON / BLOCK5 / CAM CHART(25,6)
    COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
    REAL    MACHINE DATA
    INTEGER SEQUENT
    DO 27 J=1,LIMIT
    IF (CAM CHART(J,2) .EQ. 999.999) GO TO 27
    IF (CAM CHART(J,1) .NE.  0.0) GO TO 26
    IF (SEQUENT(J) .NE. 1 .AND. SEQUENT(J) .NE. 23 .AND.
   ^    SEQUENT(J) .NE. 24 .AND. SEQUENT(J) .NE. 25) GO TO 27
    GO TO 28
26  CAM CHART(J,3) = ANINT(CAM CHART(J,1) / CAM CHART(J,2))
28  IF (SEQUENT(J) .EQ. 1 .OR. SEQUENT(J) .EQ. 23)
   ^CAM CHART(J,3) = ANINT(((OPTIMIZED DATA(MACHINE,1)/60.0)*
   ^                (MACHINE DATA(MACHINE,4))) + 5.0)
    IF (SEQUENT(J) .EQ. 24 .OR. SEQUENT(J) .EQ. 25)
   ^CAM CHART(J,6) = 1.0
27  CONTINUE
    RETURN

    END

    SUBROUTINE COMPUTE CYCLETIME (CYCLETIME,REVS PER PIECE ,
   ^                              MACHINE)

    COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
    COMMON / BLOCK3 / FORCE(7,7),MINOR
    COMMON / BLOCK5 / CAM CHART(25,6)
    COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
    INTEGER SEQUENCE
    TOTAL,REMAIN = 0.0
    DO 30 SEQUENCE = 1,LIMIT
    TOTAL = TOTAL + CAM CHART(SEQUENCE,3)
30  REMAIN = REMAIN + CAM CHART(SEQUENCE,6)
    REVS PER PIECE = (TOTAL / (100.0 - REMAIN)) * 100.0
    CYCLETIME =      ((60.0 * REVS PER PIECE) /
   ^                    OPTIMIZED DATA(MACHINE,1))
    CYCLETIME = CORRECTED TO QUARTERS (CYCLETIME)
    REVS PER PIECE = (CYCLETIME * OPTIMIZED DATA(MACHINE,1)
   ^                   / 60.0) / 100.0
```

```
      RETURN

      END

      REAL FUNCTION CORRECTED TO QUARTERS (VALUE)

      DIFFERENCE = VALUE - IFIX(VALUE)
      DO 30 I = 1,4
      IF (DIFFERENCE .GT. FLOAT(I) * 0.25) GO TO 30
      CORRECTED TO QUARTERS = IFIX(VALUE) + FLOAT(I) * 0.25
      IF (ABS(FLOAT(I) * 0.25 - DIFFERENCE) .GT.
     ^     ABS(FLOAT(I - 1) * 0.25 - DIFFERENCE))
     ^CORRECTED TO QUARTERS = IFIX(VALUE) + FLOAT(I - 1) * 0.25
      RETURN
 30   CONTINUE

      END

      SUBROUTINE COMPUTE HUNDREDTHS (REVS PER PIECE)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK5 / CAM CHART(25,6)
      INTEGER SEQUENCE
      DO 30 SEQUENCE = 1,LIMIT
      IF ((CAM CHART(SEQUENCE,3) .NE. 0.0) .AND.
     ^     (CAM CHART(SEQUENCE,6) .EQ. 0.0))
     ^       CAM CHART(SEQUENCE,6) = CAM CHART(SEQUENCE,3)
     ^                               / REVS PER PIECE
 30   CONTINUE
      RETURN

      END

      SUBROUTINE PRINT CAM CHART (CYCLETIME,MACHINE)

      COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      COMMON / BLOCK3 / MACHINE NAMES(7,3),FORCE(7,4) ,MINOR
      COMMON / BLOCK4 / METAL NAMES(3),OPERATION NAMES(25,3)
      COMMON / BLOCK5 / CAM CHART(25,6)
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      REAL    MACHINE NAMES,METAL NAMES
      INTEGER SEQUENT
      WRITE(4,33)(MACHINE NAMES(MACHINE,N),N=1,3),
     ^           (METAL NAMES(N),N=1,3),DIAMETER,
     ^           OPTIMIZED DATA(MACHINE,1),
     ^            OPTIMIZED DATA(MACHINE,1),CYCLETIME
      WRITE(4,34)((OPERATION NAMES(SEQUENT(I),K),K=1,3),
     ^           (CAM CHART(I,J),J=1,6),I=1,LIMIT)
      OPTIMIZED DATA(MACHINE,4) = CYCLETIME
      RETURN
 33   FORMAT('1',3(/),' CAM-DESIGN-CHART',/,
     ^' MACHINE NAME   : ',3A8,/,
     ^' MATERIAL NAME  : ',3A8,/,
```

```
   ^' MATERIAL SIZE  : ',F6.4,2X,'DIA.',/,
   ^' SPINDLE SPEED  : ',/,
   ^' FAST  : ',F9.4,2X,'RPM.',/,
   ^' SLOW  : ',F9.4,2X,'RPM.',/,
   ^' CYCLE TIME     : ',F9.4,2X,'SECS.',/)
34  FORMAT((' ', 3A8, 6(F8.4,1X)))

    END

    SUBROUTINE ESTIMATE COMPLETION TIME (MAJOR)

    COMMON / BLOCK3 / FORCE1(7,5),BATCH SIZE,SCRAP RATE ,
   ^                  NON PRODUCTION RATE,FORCE2(11),MINOR
    COMMON / BLOCK6 / PRIORITY DATA(7,4)
    REAL    TIME PER HOUR / 3600.0 /,NON PRODUCTION RATE
    READ(3,31,END=99) BATCH SIZE,SCRAP RATE,
   ^                  NON PRODUCTION RATE
    REDUCTION = (100.0-SCRAP RATE-NON PRODUCTION RATE)/ 100.0
    DO 30 MACHINE = MINOR,MAJOR
    HOURLY PRODUCTION = (TIME PER HOUR * REDUCTION) /
   ^                    PRIORITY DATA(MACHINE,4)
    PRIORITY DATA(MACHINE,3) =  BATCH SIZE/ HOURLY PRODUCTION
30  PRIORITY DATA(MACHINE,3) = CORRECTED TO QUARTERS
   ^                           (PRIORITY DATA(MACHINE,3))
    RETURN
99  CALL ERROR IN DATA ('ESTIMATE COMPLETION TIME')
31  FORMAT(3G0.0)

    END

    SUBROUTINE PRIORITY (MAJOR)

    COMMON/BLOCK3/MACHINE NAMES(7,3),MATRIX(7,2),FORS(7,2),
   ^              MINOR
    COMMON / BLOCK6 / PRIORITY DATA(7,4)
    DATA TEXT / '          ',' PROFIT ','DUE DATE','  TIME  ' /
    REAL    MACHINE NAMES,MATRIX,TEXT(4)
    LOGICAL ASCEND / .FALSE. /
    NUMBER = MAJOR - MINOR + 1
    DO 31 I = 2,4
    INITIAL = 0
    DO 30 MACHINE = MINOR,MAJOR
    INITIAL = INITIAL + 1
    MATRIX(INITIAL,1) = FLOAT(MACHINE)
30  MATRIX(INITIAL,2) = PRIORITY DATA(MACHINE,I)
    CALL SORT ASCEND OR DESCEND (ASCEND,MATRIX,NUMBER,2,2)
    WRITE(2,32) TEXT(I),(((MACHINE NAMES(IFIX(MATRIX(K,1)) ,
   ^                     JJ),JJ = 1,3) ,MATRIX(K,2)) ,
   ^                               K = 1,NUMBER)
31  ASCEND = .NOT. ASCEND
    RETURN
```

```
32  FORMAT(' ',2(/),T5,'PRIORITY CRITERIA : ',1X,A8,2(/),
   ^      (   T10,3A8,2X,'..................',F10.4))

    END

    SUBROUTINE SORT ASCEND OR DESCEND (ASCEND,ARRAY ,
   ^                                      ROWS,COLUMNS,CHOICE)

    REAL    ARRAY(ROWS,COLUMNS)
    INTEGER ROW,ROWS,COLUMN,COLUMNS,CHOICE,GAP
    INTEGER BEFORE,AFTER
    LOGICAL ASCEND,DESCEND
    IF (ROWS .LE. 1 .OR. COLUMNS .LE. 1) RETURN
    DESCEND = .NOT. ASCEND
    GAP = ROWS
30  GAP = GAP / 2
    IF (GAP .LE. 0) RETURN
    DO 33 ROW = 1,ROWS - GAP
    BEFORE = ROW + GAP
    AFTER  = ROW
31  IF ((ASCEND .AND. ARRAY(BEFORE,CHOICE) .GE.
   ^       ARRAY(AFTER,CHOICE)).OR.(DESCEND .AND.
   ^       ARRAY(BEFORE,CHOICE).LE.ARRAY(AFTER,CHOICE)))GOTO 33
    DO 32 COLUMN = 1,COLUMNS
    TEMP = ARRAY(BEFORE,COLUMN)
    ARRAY(BEFORE,COLUMN) = ARRAY(AFTER,COLUMN)
32  ARRAY(AFTER,COLUMN) =TEMP
    BEFORE = AFTER
    AFTER  = AFTER - GAP
    IF (AFTER .GT. 0) GO TO 31
33  CONTINUE
    GO TO 30

    END

    SUBROUTINE COMPUTE MATERIAL INFORMATION

    COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
    COMMON / BLOCK3 / FORCE(7,5),BATCH SIZE,SCRAP RATE ,
   ^                  NON PRODUCTION RATE,FORCE2(11),MINOR
    COMMON / BLOCK4 / METAL NAMES(3),OPERATION NAMES(25,3)
    REAL PIE / 3.14159263 /,MATERIAL COST,NON PRODUCTION RATE
    REAL    METAL NAMES
    PAUSE 'AS *CRO,:EPP3071.MATERIALCOST'
    DO 30 METALS = 1,MATERIAL
30  READ(1,31,END=99) MATERIAL COST
    VOLUME OF MATERIAL REQUIRED = PIE * (DIAMETER/ 24.0) ** 2
   ^                        *(TOTAL LENGHT/12.0)*BATCH SIZE
   ^                                 * (100.0 - SCRAP RATE -
   ^                                   NON PRODUCTION RATE)
    COST OF MATERIAL REQUIRED = VOLUME OF MATERIAL REQUIRED /
   ^                            MATERIAL COST
    WRITE(2,32) (METAL NAMES(I),I=1,3),DIAMETER,BATCH SIZE ,
   ^            VOLUME OF MATERIAL REQUIRED ,
```

```
^                 COST OF MATERIAL REQUIRED
   RETURN
99 CALL ERROR IN DATA ('COMPUTE MATERIAL INFORMATION')
31 FORMAT(20X,2G0.0)
32 FORMAT(' ',2(/),T5,'MATERIAL INFORMATION',2(/),
^                   T5,'MATERIAL',T25,'..  :' ,3A8,/,
^                   T5,'DIAMETER',T25,'..  :',F5.2,/,
^                   T5,'BATCH SIZE',T25,'..  :',F6.0,/,
^                   T5,'VOLUME REQD',T25,'  :',F8.2, 'CU.FT'
^                ,/,T5,'COST',T25,'..  :',F8.2)

   END

   SUBROUTINE COMPUTE TOOL INFORMATION (MAJOR)

   COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
   COMMON / BLOCK3 / QUANTITY(10),FORCE1(28),NUMBER
   COMMON / BLOCK5 / FORCE2(100),TOOL LIST(10),REQUIRE(8,5)
   EQUIVALENCE (FORCE2(1),TOOL NAMES(1))
   EQUIVALENCE (FORCE2(76),TOOL COST(1))
   REAL    TOOL NAMES(25,3),TOOL COST(25)
   INTEGER TOOL
   PAUSE 'AS *CRO,:EPP3071.TOOL LIFE'
   READ(1,33,END=99) ((REQUIRE(I,J),I=1,8),J=1,5)
   CALL ZEROIZE 1D REAL MATRIX (QUANTITY,10)
   CALL ZEROIZE 1D REAL MATRIX (TOOL LIST,10)
   CALL COMPUTE QUANTITY (MAJOR)
   PAUSE 'AS *CRO,:EPP3071.TOOL DATA'
   READ(1,31,END=99) ((TOOL NAMES(I,J),J = 1 ,3),I = 1,25)
   PAUSE 'AS *CRO,:EPP3071.TOOL COSTS'
   READ(1,34,END=99) (TOOL COST(I),I = 1,25)
   DO 30 TOOL = 1,NUMBER - 1
30 TOOL COST(IFIX(TOOL LIST(TOOL))) = QUANTITY(TOOL) *
  ^TOOL COST(IFIX(TOOL LIST(TOOL)))
   WRITE(2,32) (((TOOL NAMES(IFIX(TOOL LIST(I)),J),J = 1,3),
^              QUANTITY(I),TOOL COST(IFIX(TOOL LIST(I)))) ,
^                   I = 1,NUMBER - 1)
   RETURN
99 CALL ERROR IN DATA ('COMPUTE TOOL INFORMATION')
31 FORMAT(20X,3A8)
32 FORMAT(' ',2(/),T5,'TOOL INFORMATION',2(/),
^        (T5,3A8,'......',T35,'TOOL',2X,
^         ', QUANTITY',2X,E20.7,2X,'COST',2X,F8.2))
33 FORMAT(20X,8G0.0)
34 FORMAT(32X,G0.0)

   END

   SUBROUTINE COMPUTE QUANTITY (MAJOR)

   COMMON / BLOCK1 / TOTAL LENGTH,DIAMETER,MATERIAL,LIMIT
   COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
   COMMON / BLOCK3 / QUANTITY(10),FORCE1(25),BATCH SIZE ,
^                    SCRAP RATE,NON PRODUCTION RATE,KOUNTER
```

```
      COMMON / BLOCK5 / FORCE2(100),TOOL LIST(10)
      REAL    NON PRODUCTION RATE
      INTEGER SEQUENCE,SEQUENT,OPERATION
      KOUNTER = 1
      DO 33 SEQUENCE = 1,LIMIT
      IF (DIAMETERS(SEQUENCE) .EQ. 0.0) GO TO 33
      TOOL LIST(KOUNTER) = FLOAT(SEQUENT(SEQUENCE))
      IF (KOUNTER .EQ. 1) GO TO 31
      DO 30 OPERATION = 1,KOUNTER - 1
      IF (TOOL LIST(OPERATION) .EQ. TOOL LIST(KOUNTER))GO TO 32
30    CONTINUE
31    KOUNTER = KOUNTER + 1
      TOOL LIFE = 5.6 E 76
      CUTTING TIME = 0.0
32    PREVIOUS 1 = TOOL LIFE
      PREVIOUS 2 = CUTTING TIME
      CALL TYPE OF OPERATION (SEQUENCE,OPERATION,& 33)
      TOOL LIFE = COMPUTE TOOL LIFE (MAJOR,SEQUENCE,OPERATION ,
     ^             RPM)
      IF (TOOL LIFE .GT. PREVIOUS 1) TOOL LIFE = PREVIOUS 1
      CUTTINGTIME = COMPUTECUTTINGTIME(SEQUENCE,RPM)+PREVIOUS 2
      QUANTITY(KOUNTER - 1) = ((CUTTING TIME) * (BATCH SIZE *
     ^                        ((100.0 + SCRAP RATE) / 100.0)))
     ^                        / TOOL LIFE + 1
33    CONTINUE
      RETURN

      END

      REAL FUNCTION COMPUTE TOOL LIFE (MAJOR,SEQUENCE,
     ^      OPERATION,RPM)

      COMMON / BLOCK2 / SEQUENT(25),DIAMETERS(25)
      COMMON / BLOCK3 / FORCE1(7,7),MINOR
      COMMON / BLOCK5 / CAM CHART(25,2),FORCE2(60),K1(8),M1(8),
     ^                  N1(8),C1(8),T1(8)
      COMMON / BLOCK6 / OPTIMIZED DATA(7,4)
      REAL    PIE / 3.14159263 /,M,N,K1,M1,N1
      INTEGER SEQUENCE,SEQUENT,OPERATION
      VELOCITY = OPTIMIZED DATA(1,1)
      DO 30 MACHINE = MINOR,MAJOR
      IF (OPTIMIZED DATA(MACHINE,1) .GT. VELOCITY)
     ^      VELOCITY = OPTIMIZED DATA(MACHINE,1)
30    CONTINUE
      RPM = VELOCITY
      CONSTANT = K1(OPERATION)
      FEED = CAM CHART(SEQUENCE,2)
      VELOCITY = PIE * DIAMETERS(SEQUENCE) * VELOCITY / 12.0
      M        = M1(OPERATION)
      N        = N1(OPERATION)
      COMPUTE TOOL LIFE = (CONSTANT / (VELOCITY * FEED ** M))
     ^                      ** (1 / N)
      RETURN
```

```
END

REAL FUNCTION COMPUTE CUTTING TIME (SEQUENCE,RPM)

COMMON / BLOCK5 / CAM CHART(25,2)
INTEGER SEQUENCE
THROW = CAM CHART(SEQUENCE,1)
FEED  = CAM CHART(SEQUENCE,2)
IF (THROW .EQ.0.0.OR.FEED .EQ.0.0) STOP 'NO CUTTING TIME'
COMPUTE CUTTING TIME = THROW / (FEED * RPM)
RETURN

END


FINISH
```

Results of the Cam Design Package

GENERAL INFORMATION

THE ACTUAL LENGTH OF THE COMPONENT IS ...: 2.000
THE ACTUAL DIAMETER OF THE COMPONENT IS .: 0.375
THE SPECIFIED MATERIAL IS ....: BRASS-RO
THE SEQUENCE OF OPERATIONS ARE :

1)......FEED-STOCK
2)......INDEX-TURRET
3)......CENTRE-DRILLING
4)......DWELL
5)......INDEX-TURRET
6)......TURNING
7)......DWELL
8)......INDEX-TURRET
9)......DRILLING [3/16-5/16]
10)......CLEAR-STOCK
11)......FORMING
12)......DWELL
13)......INDEX-TURRET
14)......CONING
15)......DWELL
16)......INDEX-TURRET
17)......REAMING
18)......INDEX-TURRET
19)......PARTING-OFF
20)......PARTING-OFF
21)......CLEAR-STOCK

LIST OF POSSIBLE MACHINES

MACHINE NO. 1	......	B.S.A	38
MACHINE NO. 2	......	B.S.A	48
MACHINE NO. 3	......	B.S.A	58
MACHINE NO. 4	......	B.S.A	68
MACHINE NO. 5	......	B.S.A	98
MACHINE NO. 6	......	B.S.A	138
MACHINE NO. 7	......	B.S.A	168

METHOD OF SPEED SELECTION : OPTIMIZATION

PRIORITY CRITERIA : PROFIT

B.S.A 48	....	3.6586
B.S.A 38	....	3.6513
B.S.A 68	....	3.6468
B.S.A 58	....	3.6434
B.S.A 98	....	3.6278
B.S.A 138	....	3.6139
B.S.A 168	....	3.5840

PRIORITY CRITERIA : DUE DATE

B.S.A	48	.... 11.5000
B.S.A	38	.... 15.0000
B.S.A	68	.... 19.7500
B.S.A	58	.... 20.7500
B.S.A	98	.... 34.7500
B.S.A	138	.... 39.2500
B.S.A	168	.... 49.7500

PRIORITY CRITERIA : TIME

B.S.A	168	.... 21.5000
B.S.A	138	.... 17.0000
B.S.A	98	.... 15.0000
B.S.A	58	.... 9.0000
B.S.A	68	.... 8.5000
B.S.A	38	.... 6.5000
B.S.A	48	.... 5.0000

MATERIAL INFORMATION

MATERIAL	.. :BRASS-RO
DIAMETER	.. : 0.38
BATCH SIZE	.. : 5000.
VOLUME REQUIRED	.. : 26.041 CU.FT
COST	.. : 260.410 (COST UNITS)

TOOL INFORMATION

TOOL	QUANTITY	COST(COST UNITS)
CENTRE-DRILLING	2.39	23.92
TURNING	3.65	36.59
DRILLING [3/16-5/16]	6.04	6.05
FORMING	10.78	107.87
CONING	16.11	161.14
REAMING	5.06	50.67
PARTING-OFF	2.37	23.71

CAM-DESIGN-CHART

MACHINE NAME : B.S.A 48
MATERIAL NAME : BRASS-RO
MATERIAL SIZE : 0.3750 DIA.
SPINDLE SPEED -
FAST : 4995.0000 RPM.
SLOW : 4995.0000 RPM.
CYCLE TIME : 6.2500 SECS.

FEED-STOCK	0.000	0.000	26.000	0.000	0.000	4.997
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
CENTRE-DRILLING	0.120	0.007	17.000	0.000	0.000	3.267
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
TURNING	0.570	0.013	46.000	0.000	0.000	8.841
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
DRILLING [3-5/16]	0.280	0.005	56.000	0.000	0.000	10.763
CLEAR-STOCK	0.000	0.000	0.000	0.000	0.000	1.000
FORMING	0.100	0.001	71.000	1.000	0.000	13.646
DWELL	0.000	0.000	0.000	1.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
CONING	0.135	0.003	45.000	0.000	0.000	8.649
DWELL	0.000	0.000	0.000	0.000	0.000	1.000
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
REAMING	0.510	0.013	39.000	0.000	0.000	7.495
INDEX-TURRET	0.000	0.000	26.000	0.000	0.000	4.997
PARTING-OFF	0.060	0.003	20.000	0.000	0.000	3.844
PARTING-OFF	0.020	0.003	7.000	0.000	0.000	1.345
CLEAR-STOCK	0.000	0.000	0.000	0.000	0.000	1.000

Appendix VI : Scheduling Programs

```
DUMP ON
PROGRAM ( SCHEDULE )
INPUT  1 = CR1
OUTPUT 2 = LP2
USE    3 = DA3 / DIRECT / 1024
OUTPUT 5 = DA5 / DIRECT ( HISTORY-ODER(1) ) / 1024
OUTPUT 6 = DA6 / DIRECT ( HISTORY-MCIN(1) ) / 1024
COMPRESS INTEGER AND LOGICAL
COMPACT
TRACE 0
END


MASTER  RAJ


COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
^                  ITEM (6),JOB NUMBER,JOBS REMAINING
COMMON / BLOCK 3 / PRESENT TIME
COMMON / BLOCK 4 / I REF(51)
COMMON / BLOCK 5 / KOUNTER 3,KOUNTER 5,KOUNTER 6
REAL    LAMBDA / 7.0 /,I REF
INTEGER DAY,WEEKS,YEARS
INTEGER TIME PER DAY / 28800 /
INTEGER END OF FILE / 998877 /
JOB NUMBER,JOBS REMAINING = 0
READ(1,31) WEEKS,MONTHS,YEARS
CALL RLEASE (1)
CALL INITIAL RUN
CALL G05ECF (5.0, I REF, 50, IFAIL)
DO 30 DAY  = 1,YEARS*20*12 + MONTHS*20 + WEEKS*5
PRESENT TIME = PRESENT TIME + FLOAT(TIME PER DAY)
30 CALL SUBSEQUENT RUNS
KOUNTER 3 = JOB NUMBER + 1
IF (JOBS REMAINING .EQ. 0) KOUNTER 3 = 1
WRITE (3 ' KOUNTER 3) END OF FILE
WRITE (5 ' KOUNTER 5) END OF FILE
WRITE (6 ' KOUNTER 6) END OF FILE

STOP 'SIMULATED RUN OVER RAJ'

31 FORMAT(3G0.0)

END

SUBROUTINE INITIAL RUN

COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
^                  ITEM(6)
COMMON / BLOCK 2 / STATE OF MACHINES(21,2),JOBS  ON
^                  MACHINES(21)
COMMON / BLOCK 5 / KOUNTER 3,KOUNTER 5,KOUNTER 6
DEFINE FILE 3 (3750,21,U,KOUNTER 3)
DEFINE FILE 5 (3750, 7,U,KOUNTER 5)
```

```
      DEFINE FILE 6 (3750, 7,U,KOUNTER 6)
      REAL    CHANGE OVER TIME / 1800.0 /
      INTEGER STARTED / 987654 /
      KOUNTER 3,KOUNTER 5,KOUNTER 6 = 1
      DO 30 MACHINE = 1,21
 29   CALL INTEGER PART OF ORDER
      CALL REAL PART OF ORDER
      CALL SELECT POSSIBLE MACHINES (& 29)
      IF (REAL ORDER ITEM(5) .LT. CHANGE OVER TIME)
     ^    REAL ORDER ITEM(5)=REAL ORDER ITEM(5)+CHANGEOVER TIME
      INTEGER ORDER ITEM(6) = MACHINE
      INTEGER ORDER ITEM(7) = STARTED
      STATE OF MACHINES(MACHINE,1) = STARTING TIME(MACHINE)
      STATE OF MACHINES(MACHINE,2) = COMPLETION TIME(MACHINE)
      JOBS  ON MACHINES(MACHINE)   = MACHINE
      KOUNTER 3 = MACHINE
 30   CALL WRITE ORDER
      CALL PRINT MACHINE HISTORY
      RETURN

      END

C     THESE TWO ROUTINES GENERATES RANDOM ORDERS ACCORDINGLY.
C     THE ORDERS ARE DIVIDED INTO : -
C     INTEGER                         REAL


C     1 = COMPONENT TYPE              1 = DIAMETER
C     2 = MATERIAL CODE               2 = LENGTH
C     3 = BATCH SIZE                  3 = CYCLE TIME
C     4 = PRIORITY                    4 = LONGEST PROCESS TIME
C     5 = CAPABLE MACHINES            5 = TOTAL PROCESS TIME
C     6 = ACTUAL MACHINE NUMBER       6 = DUE DATE
C     7 = STARTED POINTER
C     8 = ENDED POINTER
C     9 = JOB NUMBER

      SUBROUTINE INTEGER PART OF ORDER

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6),JOB NUMBER,JOBS REMAINING
      INTEGER ORDER ITEM(1) = INT(RANDOM VALUE(1.0,10.0,1))
      INTEGER ORDER ITEM(2) = INT(RANDOM VALUE(1.0,10.0,1))
      INTEGER ORDER ITEM(3) = INT(RANDOM VALUE(0.0, 0.0,2))
      INTEGER ORDER ITEM(4) = INT(RANDOM VALUE(0.0,10.0,1))
      INTEGER ORDER ITEM(9) = JOB NUMBER + 1
      JOB NUMBER = JOB NUMBER + 1
      JOBS REMAINING = JOBS REMAINING + 1
      RETURN

      END

      SUBROUTINE REAL PART OF ORDER
      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
```

```
^                         ITEM(6),JOB NUMBER,JOBS REMAINING
   COMMON / BLOCK 3 / PRESENT TIME
   REAL     LOWER,LOWER FIDDLE FACTOR / 15000.0 /
   REAL     UPPER,UPPER FIDDLE FACTOR / 500000.0 /
   LOWER = 1.0
   UPPER = 1.5
   REAL ORDER ITEM(1) = RANDOM VALUE(0.0,2.0,1)
   REAL ORDER ITEM(2) = RANDOM VALUE(0.0,7.125,1)
   REAL ORDER ITEM(3) = RANDOM VALUE(0.0,10.0,4)
   REAL ORDER ITEM(4) = RANDOM VALUE(1.0,4.0,1) *
^                        REAL ORDER ITEM(3) / 8.0
30 REAL ORDER ITEM(5) = RANDOM VALUE(LOWER,UPPER,1) *
^                  FLOAT(INTEGER ORDER ITEM(3))        *
^                        REAL ORDER ITEM(3)
   IF (REAL ORDER ITEM(5) .GT. LOWER FIDDLE FACTOR) GO TO 31
   LOWER = LOWER + 1.0
   UPPER = UPPER + 1.0
   GO TO 30
31 IF (REAL ORDER ITEM(5) .LT. UPPER FIDDLE FACTOR) GO TO 32
   LOWER = LOWER - 0.1
   UPPER = UPPER - 0.1
   GO TO 30
32 REAL ORDER ITEM(6) = RANDOM VALUE(1.0,1.5,1) *
^                        REAL ORDER ITEM(5) + PRESENT TIME
   RETURN

   END

   REAL FUNCTION RANDOM VALUE (LOWER,UPPER,TYPE)

   COMMON / BLOCK 4 / I REF(51)
   REAL     LOWER,MEAN 2 /4000.0/,STANDARD DEVIATION /1000.0/
   REAL     MEAN 4 / 10.0 /,I REF
   INTEGER TYPE,G05EYF
   GO TO (30,31,32,33),TYPE
30 RANDOM VALUE = G05DAF(LOWER,UPPER)
   RETURN
31 RANDOM VALUE = G05DDF(MEAN 2,STANDARD DEVIATION)
   RETURN
32 RANDOM VALUE = FLOAT(G05EYF(I REF,50))
   RETURN
33 RANDOM VALUE = (UPPER / G05DBF(MEAN 4))
   RETURN

   END

   SUBROUTINE SELECT POSSIBLE MACHINES (*)

   COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
^                     ITEM(6),JOB NUMBER,JOBS REMAINING
   DATA MAXIMUM DIAMETERS / 0.375,0.500,0.625,0.750 ,
^                           1.000,1.625,2.000 /
   DATA MAXIMUM LENGTHS   / 2 * 4.000,2 * 5.375,3 * 7.125 /
   REAL     MAXIMUM DIAMETERS(7),MAXIMUM LENGTHS(7)
```

```
      INTEGER ORDER ITEM(5),INTEGER ORDER ITEM(6) = 0
      INTEGER ORDER ITEM(7),INTEGER ORDER ITEM(8) = 0
      DO 30 MACHINE = 1,7
      IF ((REAL ORDER ITEM(1) .LE. MAXIMUM DIAMETERS(MACHINE))
     ^                                 .AND.
     ^      (REAL ORDER ITEM(2) .LE. MAXIMUM LENGTHS(MACHINE)))
     ^         GO TO 31
30    CONTINUE
      JOBS REMAINING = JOBS REMAINING - 1
      JOB NUMBER = JOB NUMBER - 1
      RETURN 1
31    INTEGER ORDER ITEM(5) = MACHINE
      RETURN

      END

      SUBROUTINE WRITE ORDER

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 5 / KOUNTER 3
      FIND (3 ' KOUNTER 3)
      WRITE (3 ' KOUNTER 3) (INTEGER ORDER ITEM(I),I = 1,9) ,
     ^                        (REAL ORDER ITEM(I),I = 1,6)
      RETURN

      END

      REAL FUNCTION DUE DATE RATING (ONE)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 3 / PRESENT TIME
      DUE DATE RATING = (200.0 / (REAL ORDER ITEM(6) -
     ^                 PRESENT TIME - REAL ORDER ITEM(5)))
      IF (DUE DATE RATING .GE. 175.0) DUE DATE RATING = 1000.0
      RETURN

      END

      REAL FUNCTION OPERATIONS REMAINING (ONE)

      INTEGER NUMBER OF OPERATIONS / 1 /
      INTEGER AVERAGE NUMBER OF OPERATIONS / 1 /
      OPERATIONS REMAINING = ( NUMBER OF OPERATIONS /
     ^          AVERAGE NUMBER OF OPERATIONS ) * 100.0
      IF ( OPERATIONS REMAINING .GT. 100.0 )
     ^    OPERATIONS REMAINING = 1000.0
      RETURN

      END
```

```
      REAL FUNCTION PROCESS TIME REMAINING (ONE)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 3 / PRESENT TIME
      REMAINING PROCESS TIME = REAL ORDER ITEM(5)
      PROCESS TIME REMAINING = (REMAINING PROCESS TIME /
     ^                         (REAL ORDER ITEM(6) -
     ^                           PRESENT TIME)) * 200.0
      IF (PROCESS TIME REMAINING .GE. 175.0)
     ^     PROCESS TIME REMAINING = 1000.0
      RETURN

      END

      REAL FUNCTION CRITICAL ORDER RATING (ONE)


      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      DATA    VALUE / 200.0,150.0,100.0 /
      REAL    VALUE(3)
      INTEGER CRITERIA
      CRITICAL ORDER RATING = 0.0
      DO 30 CRITERIA = 1,3
      IF (INTEGER ORDER ITEM(4) .EQ. CRITERIA)
     ^     CRITICAL ORDER RATING = VALUE(CRITERIA)
   30 CONTINUE
      IF (CRITICAL ORDER RATING .GT. 100.0)
     ^     CRITICAL ORDER RATING = 1000.0
      RETURN

      END

      REAL FUNCTION SET UP COST RATING (ONE)

      INTEGER NUMBER OF CHANGES / 1 /
      SET UP COST RATING = (100.0 / NUMBER OF CHANGES)
      IF (SET UP COST RATING .GT. 100.0)
     ^     SET UP COST RATING = 1000.0
      RETURN

      END

      REAL FUNCTION LONGEST OPERATION RATING (ONE)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      LONGEST OPERATION RATING = (100.0 / REAL ORDER ITEM(4))
      IF (LONGEST OPERATION RATING .GE. 75.0)
     ^     LONGEST OPERATION RATING = 1000.0
      RETURN

      END
```

```
      SUBROUTINE OBTAIN PRIORITY RATING

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      REAL    LONGEST OPERATION RATING
      PRIORITY RATING = DUE DATE RATING (ONE)
     ^                + OPERATIONS REMAINING (ONE)
     ^                + PROCESS TIME REMAINING (ONE)
     ^                + CRITICAL ORDER RATING (ONE)
     ^                + SET UP COST RATING (ONE)
     ^                + LONGEST OPERATION RATING (ONE)
      IF (PRIORITY RATING .GT. 6000.0) STOP 'ERROR IN RATING'
      INTEGER ORDER ITEM(4) = INT(PRIORITY RATING)
      RETURN

      END

      SUBROUTINE SUBSEQUENT RUNS

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6),JOB NUMBER,JOBS REMAINING
      COMMON / BLOCK 5 / KOUNTER 3
      INTEGER ORDERS,TODAYS ORDERS
   30 TODAYS ORDERS = RANDOM VALUE (0.0,0.0,3)
      IF (TODAYS ORDERS .LT.  0 .OR.
     ^     TODAYS ORDERS .GT. 15) GO TO 30
      IF (JOBS REMAINING .EQ. 0 .AND.
     ^     TODAYS ORDERS .EQ. 0) GO TO 34
      IF (TODAYS ORDERS .EQ. 0) GO TO 33
      DO 32 ORDERS = 1,TODAYS ORDERS
   31 CALL INTEGER PART OF ORDER
      CALL REAL PART OF ORDER
      CALL SELECT POSSIBLE MACHINES (& 31)
      CALL OBTAIN PRIORITY RATING
      KOUNTER 3 = JOB NUMBER
   32 CALL WRITE ORDER
   33 CALL INTEROGATE STATE OF MACHINES
   34 CALL PRINT MACHINE HISTORY
      RETURN

      END

      SUBROUTINE READ ORDER

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 5 / KOUNTER 3
      INTEGER POINTER, PARITY ERROR KOUNTER
      POINTER, PARITY ERROR KOUNTER = KOUNTER 3
   30 FIND (3 ' KOUNTER 3)
      READ (3 ' KOUNTER 3,ERR = 31)
     ^                     (INTEGER ORDER ITEM(I),I = 1,9) ,
     ^                     (REAL ORDER ITEM(I),I = 1,6)
      RETURN
```

```
31  IF (PARITY ERROR KOUNTER .EQ. (POINTER + 10))
   ^     STOP 'RUN TERMINATED DUE TO PARITY ERROR RAJ'
    KOUNTER 3 = POINTER
    PARITY ERROR KOUNTER = PARITY ERROR KOUNTER + 1
    GO TO 30

    END

    SUBROUTINE INTEROGATE STATE OF MACHINES

    COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
   ^                   ITEM(6),JOB NUMBER,JOBS REMAINING
    COMMON / BLOCK 2 / STATE OF MACHINES(21,2),
   ^                   JOBS  ON MACHINES(21)
    COMMON / BLOCK 3 / PRESENT TIME
    COMMON / BLOCK 5 / KOUNTER 3
    LOGICAL MACHINE NOT FREE,NO JOB ASSIGNED
    DO 31 MACHINE = 1,21
    IF (JOBS REMAINING .EQ. 0) RETURN
    MACHINENOTFREE=PRESENTTIME.LE.STATEOF MACHINES(MACHINE,2)
    IF (MACHINE NOT FREE) GO TO 31
    KOUNTER 3 = JOBS  ON MACHINES(MACHINE)
    IF (KOUNTER 3 .NE. 0) CALL ELIMINATE ORDER (MACHINE)
30  CALL HIGHEST PRIORITY ORDER (MACHINE,NO JOB ASSIGNED)
    CALL ALLOCATE ORDER TO MACHINE (MACHINE,NO JOB ASSIGNED)
31  CONTINUE
    RETURN

    END

    SUBROUTINE ELIMINATE ORDER (MACHINE)

    COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
   ^                   ITEM(6)
    COMMON / BLOCK 5 / KOUNTER 3
    INTEGER ENDED / 987654 /
    CALL READ ORDER
    INTEGER ORDER ITEM(6) = MACHINE
    INTEGER ORDER ITEM(8) = ENDED
    CALL PRINT ORDER HISTORY
    KOUNTER 3 = KOUNTER 3 - 1
    CALL WRITE ORDER
    RETURN

    END

    SUBROUTINE PRINT ORDER HISTORY

    COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
   ^                   ITEM(6)
    COMMON / BLOCK 2 / STATE OF MACHINES(21,2),
   ^                   JOBS  ON MACHINES(21)
    COMMON / BLOCK 5 / KOUNTER 3,KOUNTER 5
    FIND (5 ' KOUNTER 5)
```

```
      MACHINE = INTEGER ORDER ITEM(6)
      WRITE (5 ' KOUNTER 5) INTEGER ORDER ITEM(6) ,
     ^      INTEGER ORDER ITEM(9),INTEGER ORDER ITEM(4) ,
     ^      (STATE OF MACHINES(MACHINE,I),I = 1,2)
      RETURN

      END

      SUBROUTINE HIGHEST PRIORITY ORDER (MACHINE,NO JOB
     ^                                    ASSIGNED)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6),JOB NUMBER,JOBS REMAINING
      COMMON / BLOCK 5 / KOUNTER 3
      INTEGER STARTED / 987654 /,ENDED / 987654 /
      INTEGER ORDERS,PRIORITY
      LOGICAL NO JOB ASSIGNED,MACHINE NOT SUITABLE
      PRIORITY = - STARTED
      DO 31 ORDERS = 1,JOB NUMBER
      KOUNTER 3 = ORDERS
      CALL READ ORDER
      MACHINE NOT SUITABLE = ((INTEGER ORDER ITEM(5)*3)-2) .GT.
     ^                           MACHINE
      IF (INTEGER ORDER ITEM(4) .LE. PRIORITY .OR.
     ^     MACHINE NOT SUITABLE                    ) GO TO 30
      IF (INTEGER ORDER ITEM(7) .EQ. STARTED  .OR.
     ^      INTEGER ORDER ITEM(8) .EQ. ENDED        ) GO TO 31
      K 3 = KOUNTER 3 - 1
      PRIORITY = INTEGER ORDER ITEM(4)
   30 INTEGER ORDER ITEM(4) = INTEGER ORDER ITEM(4) +
     ^                  INT(DUE DATE RATING (UPDATE))
      KOUNTER 3 = KOUNTER 3 - 1
      CALL WRITE ORDER
   31 CONTINUE
      KOUNTER 3 = K 3
      NO JOB ASSIGNED = PRIORITY .EQ. (- STARTED)
      RETURN

      END

      SUBROUTINE ALLOCATE ORDER TO MACHINE (MACHINE,NO JOB
     ^                                      ASSIGNED)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6),JOB NUMBER,JOBS REMAINING
      COMMON / BLOCK 2 / STATE OF MACHINES(21,2),
     ^                   JOBS  ON MACHINES(21)
      COMMON / BLOCK 5 / KOUNTER 3
      INTEGER STARTED / 987654 /
      LOGICAL NO JOB ASSIGNED
      IF (NO JOB ASSIGNED) JOBS  ON MACHINES(MACHINE) = 0
      IF (NO JOB ASSIGNED) RETURN
      CALL READ ORDER
      STATE OF MACHINES(MACHINE,1) = STARTING TIME(MACHINE)
```

```
      STATE OF MACHINES(MACHINE,2) = COMPLETION TIME(MACHINE)
      JOBS  ON MACHINES(MACHINE)   = INTEGER ORDER ITEM(9)
      INTEGER ORDER ITEM(6) = MACHINE
      INTEGER ORDER ITEM(7) = STARTED
      KOUNTER 3 = KOUNTER 3 - 1
      CALL WRITE ORDER
      JOBS REMAINING = JOBS REMAINING - 1
      RETURN

      END

      REAL FUNCTION STARTING TIME (MACHINE)

      COMMON / BLOCK 2 / STATE OF MACHINES(21,2)
      REAL    CHANGE OVER TIME / 1800.0 /
      STARTING TIME=STATEOFMACHINES(MACHINE,2)+CHANGEOVER TIME
      RETURN

      END

      REAL FUNCTION COMPLETION TIME (MACHINE)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 2 / STATE OF MACHINES(21,2)
      COMMON / BLOCK 3 / PRESENT TIME
      TIMELAPSE = ABS( PRESENT TIME-STATEOF MACHINES(MACHINE,2))
      COMPLETION TIME = PRESENTTIME-TIME LAPSE+REALORDERITEM(5)
      RETURN

      END

      SUBROUTINE PRINT MACHINE HISTORY

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                   ITEM(6)
      COMMON / BLOCK 2 / STATE OF MACHINES(21,2),
     ^                   JOBS  ON MACHINES(21)
      COMMON / BLOCK 5 / KOUNTER 3,KOUNTER 5,KOUNTER 6
      DO 30 MACHINE = 1,21
      KOUNTER 3 = JOBS  ON MACHINES(MACHINE)
      INTEGER ORDER ITEM(6) = MACHINE
      INTEGER ORDER ITEM(4),INTEGER ORDER ITEM(9) = 0
      IF (KOUNTER 3 .NE. 0) CALL READ ORDER
      FIND (6 ' KOUNTER 6)
      WRITE (6 ' KOUNTER 6) INTEGER ORDER ITEM(6) ,
     ^      INTEGER ORDER ITEM(9),INTEGER ORDER ITEM(4) ,
     ^      (STATE OF MACHINES(MACHINE,I),I = 1,2)
   30 CONTINUE
      RETURN

      END

      FINISH
```

```
      DUMP ON
      PROGRAM ( ORDERS )
      INPUT  3 = DA3 / DIRECT ( ORDERS(1) ) /1024
      OUTPUT 7 = LP7
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 0
      PRIORITY 1
      END


      MASTER  RAJ

      DEFINE FILE 3 (3750,21,U,KOUNTER 3)
      REAL    REAL ORDER ITEM(6)
      INTEGER INTEGER ORDER ITEM(9),ORDER,END OF FILE /998877/
30    READ (3 ' KOUNTER 3) (INTEGER ORDER ITEM(I),I = 1,9) ,
     ^                        (REAL ORDER ITEM(I),I = 1,6)
      IF (INTEGER ORDER ITEM(1) .EQ. END OF FILE)
     ^     STOP 'ALL ORDERS PRINTED RAJ'
      WRITE(7,31) (INTEGER ORDER ITEM(I),I = 1,9) ,
     ^            (REAL ORDER ITEM(I),I = 1,6)
      GO TO 30
31    FORMAT(T1,2(I2,1X),I5,1X,I5,1X,I1,1X ,I2,1X ,
     ^            3(I6,1X),4(F5.2,1X),F12.2,1X,E20.9 /)

      END

      FINISH
```

```
DUMP ON
PROGRAM ( OUTPUT )
INPUT  3 = DA3 / DIRECT ( ORDERS(1) ) / 1024
INPUT  5 = DA5 / DIRECT ( HISTORY-ODER(1) ) / 1024
INPUT  6 = DA6 / DIRECT ( HISTORY(1) ) / 1024
OUTPUT 7 = LP7
COMPRESS INTEGER AND LOGICAL
COMPACT
TRACE 0
PRIORITY 1
END


MASTER  RAJ

COMMON / BLOCK 2 / KOUNTER 3,KOUNTER 5,KOUNTER 6
DEFINE FILE 3 (3750,21,U,KOUNTER 3)
DEFINE FILE 5 (3750, 7,U,KOUNTER 5)
DEFINE FILE 6 (3750, 7,U,KOUNTER 6)
KOUNTER 3,KOUNTER 5,KOUNTER 6 = 1
CALL PROCESSED ORDERS
CALL HISTORY OF MACHINES
CALL UNPROCESSED ORDERS

STOP 'END OF OUTPUT RAJ'

END

SUBROUTINE PROCESSED ORDERS


COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
^                       ITEM(6)
COMMON / BLOCK 2 / KOUNTER 3,KOUNTER 5
LOGICAL CHECK 5 FOR RETURN
WRITE(7,31)
30 READ (5 ' KOUNTER 5) (INTEGER ORDER ITEM(I),I = 1,3) ,
^                          (REAL ORDER ITEM(I),I = 1,2)
IF (INTEGER ORDER ITEM(1) .EQ. 998877) RETURN
IF (CHECK 5 FOR RETURN(5)) RETURN
WRITE(7,32) (INTEGER ORDER ITEM(I),I = 1,3) ,
^            (REAL ORDER ITEM(I),I = 1,2)
GO TO 30
31 FORMAT(//,T55,'  PROCESSED ORDERS  ',/ ,
^            T5 ,'   MACHINE NUMBER   ',2X ,
^                '       JOB NUMBER     ',2X ,
^                '        PRIORITY      ',2X ,
^                '       START TIME     ',2X ,
^                '        FINISH TIME   ',//)
32 FORMAT(T14,I2,18X ,I6,16X,I6,9X ,2(E20.6,2X))

END

SUBROUTINE HISTORY OF MACHINES
```

```
      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                    ITEM(6)
      COMMON / BLOCK 2 / KOUNTER 3,KOUNTER 5,KOUNTER 6
      LOGICAL CHECK 5 FOR RETURN
      WRITE(7,31)
   30 READ (6 ' KOUNTER 6) (INTEGER ORDER ITEM(I),I = 1,3) ,
     ^                        (REAL ORDER ITEM(I),I = 1,2)
      IF (INTEGER ORDER ITEM(1) .EQ. 998877) RETURN
      IF (CHECK 5 FOR RETURN(5)) RETURN
      WRITE(7,32) (INTEGER ORDER ITEM(I),I = 1,3) ,
     ^            (REAL ORDER ITEM(I),I = 1,2)
      GO TO 30
   31 FORMAT(//,T55,'   MACHINE HISTORY  ',/ ,
     ^            T5,'   MACHINE NUMBER   ',2X ,
     ^               '       JOB NUMBER      ',2X ,
     ^               '         PRIORITY       ',2X ,
     ^               '         START TIME     ',2X ,
     ^               '         FINISH TIME   ',//)
   32 FORMAT(T14,I2,18X,I6,16X,I6,9X,2(E20.6,2X))

      END

      SUBROUTINE UNPROCESSED ORDERS

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
     ^                    ITEM(6)
      COMMON / BLOCK 2 / KOUNTER 3
      LOGICAL CHECK 4 FOR RETURN
      READ (3 ' KOUNTER 3) INTEGER ORDER ITEM(1)
      IF (INTEGER ORDER ITEM(1) .EQ. 998877) WRITE(7,31)
      IF (INTEGER ORDER ITEM(1) .EQ. 998877) RETURN
      KOUNTER 3 = 1
      WRITE(7,32)
   30 READ (3 ' KOUNTER 3) (INTEGER ORDER ITEM(I),I = 1,9) ,
     ^                        (REAL ORDER ITEM(I),I = 1,6)
      IF (INTEGER ORDER ITEM(8) .EQ. 987654) GO TO 30
      IF (INTEGER ORDER ITEM(1) .EQ. 998877) RETURN
      IF (CHECK 4 FOR RETURN(4)) RETURN
      WRITE(7,33) INTEGER ORDER ITEM(9),INTEGER ORDER ITEM(4) ,
     ^            INTEGER ORDER ITEM(5),REAL ORDER ITEM(6)
      GO TO 30
   31 FORMAT(//,T5,' ALL ORDERS PROCESSED ')
   32 FORMAT(//,T35,' UNPROCESSED ORDERS ',/ ,
     ^            T5,'     JOB NUMBER     ',2X ,
     ^               '         PRIORITY       ',2X ,
     ^               '         MACHINES       ',2X ,
     ^               '         DUE DATE      ',//)
   33 FORMAT(T12,I6,16X,I6,18X,I2,11X,E20.6)

      END

      LOGICAL FUNCTION CHECK 5 FOR RETURN (ITEMS)

      COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
```

```
^                      ITEM(6)
 CHECK 5 FOR RETURN = INTEGER ORDER ITEM(1) .EQ. 0 .AND.
^                       INTEGER ORDER ITEM(2) .EQ. 0 .AND.
^                       INTEGER ORDER ITEM(3) .EQ. 0 .AND.
^                       REAL ORDER ITEM(1) .EQ. 0.0 .AND.
^                       REAL ORDER ITEM(2) .EQ. 0.0
 RETURN

 END

 LOGICAL FUNCTION CHECK 4 FOR RETURN (ITEMS)

 COMMON / BLOCK 1 / INTEGER ORDER ITEM(9),REAL ORDER
^                      ITEM(6)
 CHECK 4 FOR RETURN = INTEGER ORDER ITEM(9) .EQ. 0 .AND.
^                       INTEGER ORDER ITEM(4) .EQ. 0 .AND.
^                       INTEGER ORDER ITEM(5) .EQ. 0 .AND.
^                       REAL ORDER ITEM(6) .EQ. 0.0
 RETURN

 END

 FINISH
```

```
DUMP ON
PROGRAM ( GANTT )
INPUT  7 = CR7
OUTPUT 8 = LP8
COMPRESS INTEGER AND LOGICAL
COMPACT
TRACE 0
PRIORITY 1
END


MASTER  RAJ

COMMON / BLOCK 1 / TICK,SIZE,WIDTH,GAP
COMMON / BLOCK 2 / WEEKS,MONTHS,YEARS
COMMON / BLOCK 3 / TIME PER DAY,Y LIMIT,X SCALE FACTOR
COMMON / BLOCK 4 / JOB NUMBER,START TIME,FINISH TIME
DATA    YESTERDAYS JOB NUMBER / 21 * - 999 /
INTEGER DAY,DAYS / 5 /,WEEKS,YEARS
INTEGER YESTERDAYS JOB NUMBER(21)
CALL INITIALIZATION
DAYS = DAYS * WEEKS * MONTHS * YEARS + 1
DO 30 DAY = 1,DAYS
Y LIMIT = ((GAP + WIDTH) * 21.0) - WIDTH / 2.0
DO 30 MACHINE = 1,21
READ(7,32,END=31) JOB NUMBER,START TIME,FINISH TIME
IF (JOB NUMBER .EQ. YESTERDAYS JOB NUMBER(MACHINE) .OR.
^      JOB NUMBER .EQ. 0) GO TO 30
YESTERDAYS JOB NUMBER(MACHINE) = JOB NUMBER
IF (FINISH TIME .GT. HIGHEST VALUE) HIGHEST VALUE =
^     FINISH TIME
CALL DRAW RECTANGLE
CALL FILL IN JOB NUMBER
30  Y LIMIT = Y LIMIT - (WIDTH + GAP)
31  LAST DAY = IFIX(HIGHEST VALUE / (TIME PER DAY *
^                   X SCALE FACTOR)) + 1
CALL PLOT X AXIS (LAST DAY)
CALL PLOT Y AXIS
CALL DEVEND

STOP 'END OF GANTT CHART PLOTTING RAJ'

32  FORMAT(3G0.0)

END

SUBROUTINE INITIALIZATION

COMMON / BLOCK 1 / TICK,SIZE,WIDTH,GAP
COMMON / BLOCK 2 / WEEKS,MONTHS,YEARS
COMMON / BLOCK 3 / TIME PER DAY,Y LIMIT,X SCALE FACTOR
INTEGER WEEKS,YEARS
READ(7,30) TICK,SIZE,WIDTH,GAP,X SCALE FACTOR
READ(7,30) WEEKS,MONTHS,YEARS
```

```
      TIME PER DAY = 28800 / X SCALE FACTOR
      CALL OPEN GINO GP
      CALL CHASWI (1)
      CALL CHASIZ (SIZE,SIZE)
      CALL SHIFT2 (25.0 + (20.0 * SIZE),25.0 + (20.0 * SIZE))
      RETURN
30    FORMAT(5G0.0)

      END

      SUBROUTINE PLOT X AXIS (LAST DAY)

      COMMON / BLOCK 1 / TICK,SIZE
      COMMON / BLOCK 2 / WEEKS,MONTHS,YEARS
      COMMON / BLOCK 3 / TIME PER DAY
      INTEGER DAY,DAYS / 5 /,WEEKS,YEARS
      X LIMIT = FLOAT(DAYS * WEEKS * MONTHS * YEARS + 2) *
     ^                TIME PER DAY
      IF (X LIMIT .LT. FLOAT(LAST DAY) * TIME PER DAY)
     ^     X LIMIT = FLOAT(LAST DAY) * TIME PER DAY
      DAYS = DAYS * WEEKS * MONTHS * YEARS + 2
      IF (DAYS .LT. LAST DAY) DAYS = LAST DAY
      CALL MOVTO2 (X LIMIT,0.0)
      CALL LINTO2 (0.0    ,0.0)
      DO 30 DAY = 1,DAYS
      X POSITION = FLOAT(DAY) * TIME PER DAY
      CALL MOVTO2 (X POSITION,0.0   )
      CALL LINTO2 (X POSITION,- TICK)
      CALL MOVTO2 (X POSITION - (1.5 * SIZE) ,
     ^            - (TICK + 2.0 * SIZE))
30    CALL CHAINT (DAY,3)
      CALL MOVTO2 ((X LIMIT / 2.0) - (12.0 * SIZE) ,
     ^            - (20.0 * SIZE))
      CALL CHAHOL ('TIME IN DAYS*.')
      RETURN

      END

      SUBROUTINE PLOT Y AXIS

      COMMON / BLOCK 1 / TICK,SIZE,WIDTH,GAP
      COMMON / BLOCK 3 / TIME PER DAY,Y LIMIT
      Y LIMIT = ((GAP + WIDTH) * 21.0) + WIDTH / 2.0
      CALL MOVTO2 (0.0,0.0    )
      CALL LINTO2 (0.0,Y LIMIT)
      Y POSITION = Y LIMIT - (WIDTH / 2.0)
      DO 30 MACHINE = 1,21
      CALL MOVTO2 (0.0   ,Y POSITION)
      CALL LINTO2 (- TICK,Y POSITION)
      CALL MOVTO2 (- (TICK + 3.0 * SIZE) ,
     ^                  Y POSITION - (SIZE / 2.0))
      CALL CHAINT (MACHINE,2)
30    Y POSITION = Y POSITION - (WIDTH + GAP)
      CALL MOVTO2 (- (20.0 * SIZE) ,
```

```
^           (Y LIMIT / 2.0) - (14.0 * SIZE))
CALL ROTAT2 (  90.0)
CALL CHAHOL ('MACHINE NUMBER*.')
CALL ROTAT2 (- 90.0)
RETURN

END

SUBROUTINE DRAW RECTANGLE

COMMON / BLOCK 1 / TICK,SIZE,WIDTH
COMMON / BLOCK 3 / TIME PER DAY,Y LIMIT,X SCALE FACTOR
COMMON / BLOCK 4 / JOB NUMBER,START TIME,FINISH TIME
START TIME = START TIME / X SCALE FACTOR
FINISH TIME = FINISH TIME / X SCALE FACTOR
CALL MOVTO2 (START TIME ,Y LIMIT         )
CALL LINTO2 (FINISH TIME,Y LIMIT         )
CALL LINTO2 (FINISH TIME,Y LIMIT + WIDTH)
CALL LINTO2 (START TIME ,Y LIMIT + WIDTH)
CALL LINTO2 (START TIME ,Y LIMIT         )
RETURN

END

SUBROUTINE FILL IN JOB NUMBER

COMMON / BLOCK 1 / TICK,SIZE,WIDTH
COMMON / BLOCK 3 / TIME PER DAY,Y LIMIT
COMMON / BLOCK 4 / JOB NUMBER,START TIME,FINISH TIME
INTEGER FIELD WIDTH
FIELD WIDTH = IFIX(ALOG10(FLOAT(JOB NUMBER))) + 1
X POSITION = ((FINISH TIME - START TIME) / 2.0)
^           - ((FIELD WIDTH / 2.0) * SIZE) + START TIME
Y POSITION = Y LIMIT + (WIDTH / 2.0) - (SIZE / 2.0)
CALL MOVTO2 (X POSITION,Y POSITION)
CALL CHAINT (JOB NUMBER,FIELD WIDTH)
RETURN

END


FINISH
```

Results of the Scheduling Package

Randomly Generated Orders

2	2	4266	6	2	1	N	N	1	0.5	1.3	8.4	3.3	52537	67812
3	5	2159	3	1	2	N	N	2	0.3	3.7	2.4	0.5	16206	22753
2	3	3497	0	7	3	N	N	3	1.6	4.1	1.0	0.3	15903	23231
1	2	3145	6	6	4	N	N	4	1.6	1.4	0.7	0.2	15265	18549
5	2	5061	2	6	5	N	N	5	1.5	3.8	0.6	0.1	18912	22896
8	5	4661	5	7	6	N	N	6	1.9	0.9	5.5	2.6	34819	46617
9	1	3052	8	1	7	N	N	7	0.3	2.9	1.8	0.4	18168	26109
5	9	3828	2	7	8	N	N	8	1.7	3.5	7.9	2.5	31862	38874
8	6	2645	0	2	9	N	N	9	0.4	3.0	4.1	2.0	26750	35149
3	5	3646	6	6	10	N	N	10	1.2	2.1	0.5	0.1	15860	20786
3	7	2752	4	6	11	N	N	11	1.3	1.5	4.3	1.9	26790	32821
2	5	2480	6	5	12	N	N	12	0.8	1.3	18.3	2.6	45401	63091
7	4	4550	3	5	13	N	N	13	0.9	7.0	1.5	0.5	15422	21739
4	5	4917	2	6	14	N	N	14	1.2	1.4	1.1	0.4	17096	21349
3	5	3970	8	5	15	N	N	15	1.0	4.8	0.8	0.4	18102	18413
9	3	4195	1	4	16	N	N	16	0.7	1.8	1.3	0.4	17694	24875
8	2	2826	1	5	17	N	N	17	0.3	6.4	1.7	0.2	15666	17163
5	3	2673	9	5	18	N	N	18	1.0	0.2	4.6	1.5	16531	22678
1	6	4499	5	5	19	N	N	19	0.5	6.6	0.4	0.2	16049	19977
5	2	2679	0	5	20	N	N	20	1.0	3.4	4.8	2.1	27421	28018
4	7	4846	8	7	21	N	N	21	1.7	4.9	14.9	5.1	98049	112706
7	3	2933	2200	6	17	N	N	22	1.5	6.5	1.5	0.7	19480	51006
2	8	4969	2369	1	2	N	N	23	0.2	0.7	3.4	1.1	23924	57061
4	3	4326	2214	6	16	N	0	24	1.4	3.1	27.7	7.1	159284	205319
5	6	4324	1339	7	20	N	N	25	1.9	0.8	3.3	0.6	18530	55401
1	8	3680	1235	1	3	N	N	26	0.1	0.9	12.0	2.8	52064	82582
7	8	3406	2200	6	18	N	N	27	1.1	4.4	2.4	0.4	18281	49576
4	7	4615	412	5	13	N	N	28	0.9	5.3	8.7	2.5	55144	92586
1	2	3506	1341	6	19	N	N	29	1.5	0.6	0.6	0.1	15676	50943
2	9	5710	1234	3	7	N	N	30	0.0	4.8	7.4	2.9	43514	78332
8	1	4508	2200	4	10	N	N	31	0.7	2.5	1.2	0.2	16256	45423
8	6	1874	2200	7	19	N	N	32	1.9	4.9	0.5	0.2	15853	74408
6	5	2333	1356	1	1	N	N	33	0.3	3.8	1.6	0.5	15776	77789
9	6	3423	2200	7	20	N	N	34	1.8	0.4	0.5	0.2	16807	75978
6	8	2448	1337	5	14	N	N	35	0.8	2.5	4.4	1.2	22180	89771
3	8	4372	1338	7	21	N	N	36	1.8	0.1	0.5	0.1	15123	79481
9	3	4704	392	5	15	N	N	37	0.2	5.9	11.6	5.5	60664	127084
3	6	3946	2362	1	1	N	N	38	0.1	2.8	0.8	0.3	17265	107634
6	7	4517	380	5	13	N	N	39	0.9	1.9	12.6	4.8	59012	160472
9	7	3700	1457	7	20	N	N	40	1.7	0.6	1.4	0.4	18460	109917
3	9	3348	3200	7	19	N	N	41	1.8	1.7	1.8	0.3	18307	107030
9	6	4065	1334	6	17	N	N	42	1.6	4.2	0.8	0.4	17281	112046
3	6	3218	424	6	18	N	N	43	1.2	4.6	3.6	1.4	28945	123939
6	8	3306	1339	7	20	N	N	44	1.7	0.3	2.2	0.8	15577	137603
8	3	4495	2371	7	19	N	N	45	2.0	4.0	1.1	0.5	17257	135277
3	5	3545	1457	2	4	N	0	46	0.4	2.1	4.4	1.3	19164	168290
1	7	3789	2200	6	17	N	0	47	1.6	2.0	1.5	0.5	18925	164845
6	4	3637	2200	5	13	N	0	48	0.9	1.4	1.1	0.5	17793	163208
4	9	3991	375	2	5	N	0	49	0.4	2.6	9.2	4.1	51876	212440

Results of the Output Program

PROCESSED ORDERS

M/C	J/N	PRY	START	FINISH
2	2	3	1800.00	16205.91
3	3	0	1800.00	15902.78
4	4	6	1800.00	15264.80
5	5	2	1800.00	18911.68
7	7	8	1800.00	18168.39
9	9	0	1800.00	26750.07
10	10	6	1800.00	15859.76
11	11	4	1800.00	26790.22
13	13	3	1800.00	15422.22
14	14	2	1800.00	17096.12
15	15	8	1800.00	18102.01
16	16	1	1800.00	17693.76
17	17	1	1800.00	15666.24
18	18	9	1800.00	16531.30
19	19	5	1800.00	16048.97
20	20	0	1800.00	27421.04
1	1	6	1800.00	52536.94
2	23	2369	18005.91	40129.80
6	6	5	1800.00	34818.56
8	8	2	1800.00	31861.84
10	31	2200	17659.76	32116.18
12	12	6	1800.00	45400.81
17	22	2200	17466.24	35145.89
18	27	2200	18331.30	34812.23
19	29	1341	17848.97	31725.09
20	25	1339	29221.04	45950.78
1	33	1356	54336.94	68313.25
3	26	1235	17702.78	67966.97
7	30	1234	19968.39	61682.30
13	28	412	17222.22	70565.93
14	35	1337	18896.12	39276.16
15	37	392	19902.01	78766.28
19	32	2200	33525.09	47578.54
20	34	2200	47750.78	62757.61
1	38	2362	70113.25	85578.00
17	42	1334	36945.89	52427.19
18	43	424	36612.23	63756.76
19	41	3200	49378.54	65885.85
20	40	1457	64557.61	81217.26
21	21	8	1800.00	98048.73
13	39	380	72365.93	129577.68
19	45	2371	67685.85	83142.87
20	44	1339	83017.26	96794.73
21	36	1338	99848.73	113171.48

MACHINE HISTORY

M/N	J/N	PRY	START	FINISH
1	1	6	1800.00	52536.94
2	2	3	1800.00	16205.91
3	3	0	1800.00	15902.78
4	4	6	1800.00	15264.80
5	5	2	1800.00	18911.68
6	6	5	1800.00	34818.56
7	7	8	1800.00	18168.39
8	8	2	1800.00	31861.84
9	9	0	1800.00	26750.07
10	10	6	1800.00	15859.76
11	11	4	1800.00	26790.22
12	12	6	1800.00	45400.81
13	13	3	1800.00	15422.22
14	14	2	1800.00	17096.12
15	15	8	1800.00	18102.01
16	16	1	1800.00	17693.76
17	17	1	1800.00	15666.24
18	18	9	1800.00	16531.30
19	19	5	1800.00	16048.97
20	20	0	1800.00	27421.04
21	21	8	1800.00	98048.73
1	1	6	1800.00	52536.94
2	23	2369	18005.91	40129.80
3	26	1235	17702.78	67966.97
4	0	0	1800.00	15264.80
5	0	0	1800.00	18911.68
6	6	5	1800.00	34818.56
7	30	1234	19968.39	61682.30
8	8	2	1800.00	31861.84
9	0	0	1800.00	26750.07
10	31	2200	17659.76	32116.18
11	0	0	1800.00	26790.22
12	12	6	1800.00	45400.81
13	28	412	17222.22	70565.93
14	0	0	1800.00	17096.12
15	0	0	1800.00	18102.01
16	24	2214	19493.76	176977.49
17	22	2200	17466.24	35145.89
18	27	2200	18331.30	34812.23
19	29	1341	17848.97	31725.09
20	25	1339	29221.04	45950.78
21	21	8	1800.00	98048.73
1	33	1356	54336.94	68313.25
2	0	0	18005.91	40129.80
3	26	1235	17702.78	67966.97
4	0	0	1800.00	15264.80
5	0	0	1800.00	18911.68
6	0	0	1800.00	34818.56
7	30	1234	19968.39	61682.30
8	0	0	1800.00	31861.84

M/N	J/N	PRY	START	FINISH
9	0	0	1800.00	26750.07
10	0	0	17659.76	32116.18
11	0	0	1800.00	26790.22
12	0	0	1800.00	45400.81
13	28	412	17222.22	70565.93
14	35	1337	18896.12	39276.16
15	37	392	19902.01	78766.28
16	24	2214	19493.76	176977.49
17	0	0	17466.24	35145.89
18	0	0	18331.30	34812.23
19	32	2200	33525.09	47578.54
20	34	2200	47750.78	62757.61
21	21	8	1800.00	98048.73
1	38	2362	70113.25	85578.00
2	0	0	18005.91	40129.80
3	0	0	17702.78	67966.97
4	0	0	1800.00	15264.80
5	0	0	1800.00	18911.68
6	0	0	1800.00	34818.56
7	0	0	19968.39	61682.30
8	0	0	1800.00	31861.84
9	0	0	1800.00	26750.07
10	0	0	17659.76	32116.18
11	0	0	1800.00	26790.22
12	0	0	1800.00	45400.81
13	39	380	72365.93	129577.68
14	0	0	18896.12	39276.16
15	0	0	19902.01	78766.28
16	24	2214	19493.76	176977.49
17	42	1334	36945.89	52427.19
18	43	424	36612.23	63756.76
19	41	3200	49378.54	65885.85
20	40	1457	64557.61	81217.26
21	21	8	1800.00	98048.73
1	0	0	70113.25	85578.00
2	0	0	18005.91	40129.80
3	0	0	17702.78	67966.97
4	0	0	1800.00	15264.80
5	0	0	1800.00	18911.68
6	0	0	1800.00	34818.56
7	0	0	19968.39	61682.30
8	0	0	1800.00	31861.84
9	0	0	1800.00	26750.07
10	0	0	17659.76	32116.18
11	0	0	1800.00	26790.22
12	0	0	1800.00	45400.81
13	39	380	72365.93	129577.68
14	0	0	18896.12	39276.16
15	0	0	19902.01	78766.28
16	24	2214	19493.76	176977.49
17	0	0	36945.89	52427.19
18	0	0	36612.23	63756.76

M/N	J/N	PRY	START	FINISH
19	45	2371	67685.85	83142.87
20	44	1339	83017.26	96794.73
21	36	1338	99848.73	113171.48
1	0	0	70113.25	85578.00
2	0	0	18005.91	40129.80
3	0	0	17702.78	67966.97
4	46	1457	17064.80	34428.82
5	49	375	20711.68	70787.53
6	0	0	1800.00	34818.56
7	0	0	19968.39	61682.30
8	0	0	1800.00	31861.84
9	0	0	1800.00	26750.07
10	0	0	17659.76	32116.18
11	0	0	1800.00	26790.22
12	0	0	1800.00	45400.81
13	48	2200	131377.68	147371.08
14	0	0	18896.12	39276.16
15	0	0	19902.01	78766.28
16	24	2214	19493.76	176977.49
17	47	2200	54227.19	71352.56
18	0	0	36612.23	63756.76
19	0	0	67685.85	83142.87
20	0	0	83017.26	96794.73
21	0	0	99848.73	113171.48

UNPROCESSED ORDERS

J/N	PRY	M/C	DUE
24	2214	6	205319.08
46	1457	2	168289.61
47	2200	6	164844.58
48	2200	5	163207.90
49	375	2	212440.34

References

(1) Frost E.H., Smith, Marten H.R.
Machine Shop Production Optimization - A target for research
The Production Engineer - Nov. 1968

(2) Bellman R.
Applied Dynamic Programming
Princeton Univ Press - 1962

(3) Knuth D.E.
The Art of Computer Programming Vol 3
Addison-Wesley - 1968

(4) Milner D.A.
Manufacturing Systems: An Approach to Machining Process Optimization
IEEE Transactions on Manufacturing Technology, - June 1974

(5) Yong C.K.
Dynamic Programming in Manufacturing Systems
Bsc Project, Univ of Aston in B'ham, Dept of Prod Eng - 1977

(6) Behrend H.
How to Monitor Absence from Work : from Head Count to Computer
Institute of Personel Management - 1978

(7) Jones R.M.
Manpower : Absenteeism
Dept. of Employement, Paper No. 4 - 1971

(8) Taylor P.J.
Notes for Managers : No. 15 - Absenteeism
The Industrial Society - 1973

(9) Clark J.
Time Out ? : A study of Absenteeism among nurses
Royal College of Nursing - 1975

(10) Hallaire J.
Employement of Special Groups : Part Time Employement
Volume 6

(11) Confederation of British Industry
Absenteeism : An Analysis of the Problem
Confederation of British Industry - 1970

(12) Rubenstein R.Y.
Simulation and the Monte Carlo Method
Wiley - 1981

(13) Evans, G.W.
Simulation using Digital Computer
Prentice-Hall - 1967

(14) Hammersley J.M.
Monte Carlo Methods
Methuen - 1964

(15) Law T.D.
Simulation
Univ of Aston in B'ham, Dept Publication

(16) King J.R.
On the Optimum Size of Workforce Engaged in the Servicing of Automatic Machines
Inst of Prod Eng - 1970

(17) Palm D.C.
The Assignment of Workers in Servicing Automatic Machines
Journal of Industrial Engineering - 1958

(18) Ashcroft H.
The Productivity of Several Machines Under the Care of One Operator
Journal of the Royal Statistical Soceity - 1950

(19) Emshoff J.R.
Design and use of computer Simulation models
Macmillan - 1970

(20) Lewis T.G.
Computer Principles of Modelling and Simulation
Houghton Mifflin - 1979

(21) Milner D.A.
A Series of Lecture Notes
Univ of Aston in B'ham. Dept of Prod Eng - 1976

(22) Hewlett-Packard 7220C and 7220T
Graphic Plotters Operating and Programming Manual using HP-GL Instructions
Hewlett-Packard Publications - 1981

(23) GinoF
GinoF : User Manual
CAD Centre - 1976

(24) GinoGraf
GinoGraf : User Manual
CAD Centre - 1976

(25) GinoSurf
GinoSurf : User Manual
CAD Centre - 1978

(26) Gilbert W.W.
Economics of Machining : Machining-Theory and Practice
American Society for Metals - 1950

(27) Taylor F.W.
The Art of Cutting Metal
Transactions of the ASME - 1907

(28) Wu S.M., Ermer D.S.
Maximum Profit as the criterion in the determination of the optimum cutting conditions.
Journal of Engineering for Industry - Nov. 1966

(29) Okushima K., Hitomi K.
A study of economic machining : An analysis of the maximum profit cutting speed.
International Journal of Production Research - 1964

(30) Armarego J.A., Russel
Maximum profit Rate as a Criterion for the selection of Machining Conditions.
Int. Journal of Machine Tool Design and Research - 1966

(31) Wu S M., Tee L.H.
Computerized determination of optimum conditions for a fixed demand.
Advances in Machine Tool Design and Research - 1968

(32) Goranski G.K.
Calculating cutting speeds and feeds with computer.
National Lending Library for Science and Technology - 1967

(33) Shaw M.C.
Metal Cutting Principles
MIT Press - 1957

(34) Frost E.H., Smith, Marten H.R.
Optimising control of a batch-production machine shop
Proc. IEE Vol. 118, No 12, - Dec. 1971

(35) Colding B., Konig W.
Validity of the Taylor Equation in Metal Cutting
Annals of the C I R P, Vol XVIV pp 793-812, - 1971

(36) Crookall J.R
The Performance-Envelope Concept in the Economics of Machining
Int. J Mach Tool, Vol 9 pp 261-278, - 1969

(37) French D., Milner D.A., Weston W.J.
The Application of Numerical Control to Centre Lathe Operations - The use of Computing Programming to obtain the Parameters for Machining. Proceedings of the 12th International Machine Tool Design and Research - Conference
Macmillan - 1970

(38) Solomentsov Y.M.
Calculating Optimal Machining Conditions
Russian Engineering Journal, Vol XLIX, - 1970

(39) Russel J.H.
Computer Applications to Factory Control Problems
The Production Engineer - 1963

(40) Tobias S.A., Koenigsberger F.
Proceedings of the ... International M T D R Conference
Advances in Machine Tool design and Research - 1968

(41) Brewer R.C, Rueda R.R.
A Simplified Approach to the Optimum Selection of Machining Parameters
Engineering Digest - Sept 1963

(42) Machine Tool Industry Research Association
Production Technology : Abstracts and Reports from Eastern Europe, 1967 - 1973

(43) Technical Staff of the Machinability Data Centre
Machining Data Handbook - 1972

(44) Armarego E.J.A.
The Machining of Metals
Prentice-Hall - 1969

(45) Brindley J.D.
Machining Parameters and Computer Control
Phd Thesis, Univ of Aston in B'ham, Dept of Prod Eng - 1977

(46) Kronenberg M.
Machining Science and Application : theory and practice for operation and development
Pergamon - 1966

(47) American Society for Metals
Machining Theory and Practice - 1950

(48) Metal Cutting Tool Institute
Metal Cutting Tool Handbook
Metal Cutting Tool Institute - 1965

(49) Koenigsberger F.
Machine Tool Structures
Pergamon - 1970

(50) GinoZone
GinoZone : User Manual
CAD Centre - 1979

(51) Alfred Herbert Ltd
Tool Layout and Cam Design for B.S.A. Single-Spindle Autos
B.S.A Tools Ltd. - 1974

(52) Town H.C.
Automatic Machine Tools
Ileffe - 1968

(53) Alfred Herbert Ltd.
How to Layout Turret Lathe Tools
Herbert - 1919

(54) Cook S.J.
Single-Spindle Automatics
Machining Publishing Co. - 1962

(55) Durney W.C.
Capstan and Turret Lathes
Pitman - 1945

(56) B.S.A Tools Ltd.
Layout Tools and Designing Cams for B.S.A
B.S.A - 1949

(57) Machinery
Programme Controlled Lathes
Machinery Publishing Co. - 1966

(58) Morgan G.A.W
Swiss-type Automatics : Part 3 : Cam Layout and Tool Design
Machinery Publishing Co. - 1955

(59) Houghton P.S.
Lathes : Vol 2 : Automatic Lathes, Single and Multi Spindle
Pitman - 1963

(60) I C L 1900 Series : CSL (Control and Simulation languages)
International Computers Ltd - Technical Publication - 1966

(61) I C L 1900 Series : Extended Fortran
International Computers Ltd - Technical Publication - 1971

(62) I C L 1900 Series : Fortran 32K/48K Disc Compilers
International Computers Ltd - Technical Publication - 1972

(63) I C L 1900 Series : Fortran, George 3 & 4 Compilers
International Computers Ltd - Technical Publication - 1972

(64) I C L 1900 Series : Operating System, George 3 & 4
International Computers Ltd - Technical Publication - 1969

(65) Eilon S.
Due Dates in Job Shop Scheduling
Int Jrnl Prod Res - Mar 1976

(66) Beckmann M.J.
Dynamic Programming of Economic Decisions
Springer-Verlag - 1968

(67) Conway R., Maxwell W.L.
Theory of Scheduling
Addison-Wesley - 1967

(68) Elmaghraby S.E.
The Design of Production Systems
Chapman and Hall - 1966

(69) Lock D.
Industrial Scheduling Techniques
Gower Press - 1971

(71) National Engineering Laboratory
Automated Small Batch Production : Technical study
National Engineering Laboratory - 1978

(72) Onyeagoro E.A.
Cellular Scheduling for Batch Production - Phd Thesis
Univ of Aston in B'ham. Dept of Prod Eng - 1977

(73) Trusler J.D.C
Production Control by Computer : with particular reference to jobbing and small batch production
Machining Publishing Co. - 1968

(74) Eilon S.
Job Shop Scheduling with Due Dates
Int Jrnl Prod Res - 1976

(75) Woodward S.N.
The use of Computers in the Management of Batch Production
Graduate Centre for Management Studies - 1971

(76) Mind Your Own Business
Cairnmark Ltd. - monthly

(77) Computing
VNU Business Publications - weekly

(78) MiniComputer
Compass Press Ltd. - monthly

(79) DataLink
VNU Business Publications - weekly

(80) Infomatics
VNU Business Publications - monthly

(81) Systems International
IPC Business Press Ltd. - monthly

(82) Office Systems
IPC Business Press Ltd. - monthly

(83) CBM 8032 Users Manual
Commodore Business Machines - 1980

(84) CBM 8050 Users Manual
Commodore Business Machines - 1980

(85) MCS6500 Microcomputer Family Programming Manual
MOS Technology - 1976

(86) The Pet Revealed
Nick Hampshire Publication - 1979