University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

1-1-1987

A microcomputer-based simulation model of a flexible manufacturing system with application to scheduling rules.

Thiruvengadam Ravi
University of Windsor

A MICROCOMPUTER-BASED SIMULATION MODEL OF A FLEXIBLE MANUFACTURING SYSTEM WITH APPLICATION TO SCHEDULING RULES

by

Thiruvengadam Ravi

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

Windsor, Ontario, Canada

1987

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

Thiruvengadam Ravi

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

Thiruvengadam Ravi

To the one and only ????

ABSTRACT

This research deals with the development of a microcomputer-based simulation model of a Flexible Manufacturing System (FMS) using the general purpose simulation language SLAM (Simulation Language for Alternative Modeling). The FMS under consideration is composed of general purpose machining centers laid out in separate lanes, a loading and an unloading station, a central parts storage to store the raw materials and work-in-process parts, and automated material handling systems.

The simulation model which is adaptive to variations in the layout, serves as a decision tool to select a scheduling rule from amongst the five rules, namely, Random selection (RANDOM), Fewest Operations Remaining (FOPR), Most Operations Remaining (MOPR), Shortest Processing Time (SPT) and Longest Processing Time (LPT). The model includes realistic aspects such as, alternate routing for certain operations, treatment of fixtures as resources, and part types with priorities. The simulation model is user-interactive and does not require a prior working knowledge of SLAM.

Standard statistical techniques are used to select a scheduling rule from the five rules under consideration. An experimental design setup is used to aid the study of the effects of the different resources on the system performance. These procedures are illustrated through examples.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to Dr. R. S. Lashkari and Dr. S. P. Dutta for their guidance and support during the entire course of this research. I would also like to thank Dr. R. J. Caron and Dr. P. Brill for reviewing my thesis and providing useful suggestions. A special note of thanks goes to Jacquie Mummery and Tom Williams for their invaluable help from time to time.

I wish to thank my friends, all the graduate students in the Department of Industrial Engineering, who have provided invaluable help through useful discussions.

Finally, I am very grateful to my parents, Dr. Thiruvengadam and Dr. Sulochana, for always being there when I needed them, in spite of the distance.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

APPENDICES

CHAPTER I

INTRODUCTION

The fact that approximately 75% of the dollar volume of all metal-worked products are manufactured in lot sizes of less than fifty pieces each (Cook, 1975), spurred intensive research to develop a manufacturing system capable of producing mid-sized batches of a wide variety of parts efficiently. One important outcome of this research was the development of Flexible Manufacturing Systems (FMS) - systems capable of manufacturing mid-sized batches of a wide variety of parts with the efficiency of automated mass production systems and the flexibility of job shop systems. Referring to Fig.1, it can be seen that flexible manufacturing systems fit in between these two categories of manufacturing systems.

Rapid technological advances in the areas of numerically controlled machine tools (NC), automated material handling systems (MHS) and computer

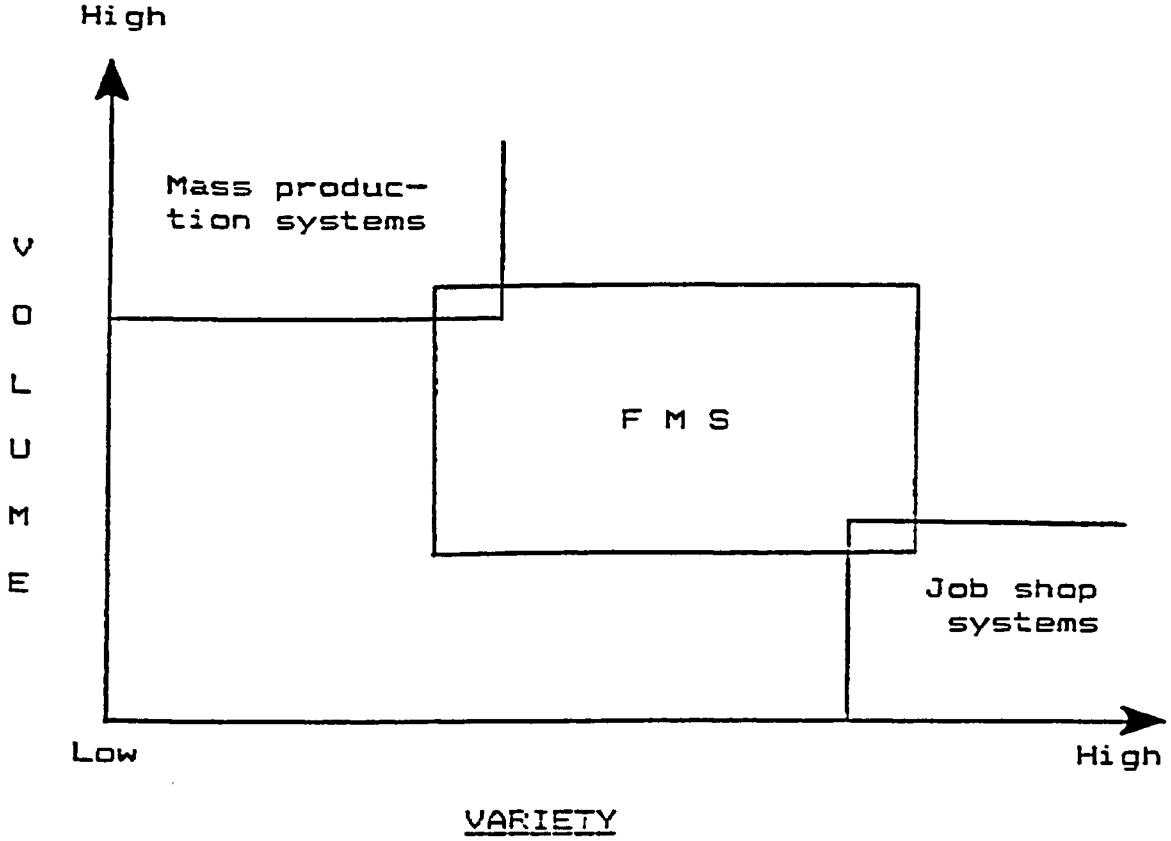


Fig.1 Comparison of various manufacturing systems

[Source: Groover & Zimmers (1984)]

communications have made flexible manufacturing systems possible. The high efficiency and productivity desired from such systems cannot be achieved, solely, by installing these highly sophisticated and expensive hardware. Due considerations should be given to the operational aspects, i.e., the software aspects, of these systems. This is essentially a complex managerial decision problem and at present is a major area of focus of researchers in the field of applied Operations Research.

This research deals with the development of a microcomputer-based simulation model of an FMS which can be used to address the operating problems. A brief introduction to the types of flexible manufacturing systems and the components that constitute an FMS is given in Chapter 2. The configuration of the FMS that has been considered for the development of the simulation model is also described in Chapter 2. In Chapter 3, the various problems posed by these systems and the modeling techniques that have been used in the past to address these problems are discussed. The objectives of this research are also presented in this chapter. The mode of operation of the FMS and the development of the simulation model on a microcomputer using SLAM (Simulation Language for Alternative Modeling) are discussed in Chapter 4. In Chapter 5, the application of the model to analyse problems is illustrated through the use of three examples. The statistical technique for selecting a scheduling rule (policy) and the experimental design setup used to aid the study of the effects of important resources on the system performance are presented in the same chapter. Chapter 6 summarises the work done and provides some directions for further work that could be done to enhance the features of the model.

CHAPTER II

FLEXIBLE MANUFACTURING SYSTEMS

2.1 Definition and types of FMS

A flexible manufacturing system is described as "an automated batch manufacturing system that consists of a set of numerically controlled machine tools which are connected by automated material handling systems, all under the control of a central computer" (Groover & Zimmers, 1984).

Flexible manufacturing systems are broadly classified into two types (Browne & Rathmill, 1983), namely:

(i) Dedicated systems that are capable of processing a limited variety of parts in medium-sized batches;

(ii) Random systems that are capable of processing a larger variety of parts in small-sized batches.

2.2 Components of an FMS

An FMS consists of several components that interact with one another to achieve the goal of producing medium-sized or small-sized batches of a limited or larger variety of parts. The major components that constitute an FMS are:

(i) Numerically controlled machines

(ii) Automated material handling systems

(iii) Fixtures and pallets

(iv) Control computer(s).

Numerically controlled machines form the nucleus around which the whole system is built. These machines are capable of machining both rotational and non-rotational (prismatic) parts. More than 70% of the existing systems machine prismatic parts, while approximately 27% machine rotational parts and a few of them machine both types of parts (Bilalis & Manalis, 1985). These machines are equipped with servomotors that control the movement of the machine spindle and work table. Manual or programmed methods are used to generate electrical pulses from electronic controllers and these pulses in turn actuate the servomotors. Tool magazines that are capable of holding 100 to 150 cutting tools, e.g., drills, boring tools, reamers, milling cutters, taps, etc., are provided on these machines. An automatic tool interchanger interfaces the machine spindle and the tool magazine. This device can interchange two cutting tools within a few seconds. Hence, as long as the required tool is in the tool magazine, the setup time required for interchanging the tools in between successive, but different processes is negligible. The introduction of a new part to be machined calls for a different set of cutting tools which can be set up quickly, thereby resulting in reduced setup times. This is not the case in conventional manufacturing systems.

Automated material handling systems are broadly classified as primary systems, e.g., tow carts, automatic guided vehicles (AGV's) and conveyors; and secondary systems, e.g., shuttles. The primary systems transport the parts between the various stations, and the secondary systems transport the parts between the primary systems and the stations. The secondary systems also serve as storage areas for the parts at the machines, before and after machining, and are known as the input buffer and the output buffer, respectively. The purpose of the input buffer is to keep the machine busy, so that as soon as a part has been processed, the next part to be processed on the machine would be waiting in the input buffer. The part that has completed an operation on the machine is transferred to the output buffer where it waits for the unloading AGV. This essentially prevents the machined part from waiting on the machine table, which would otherwise result in the machine remaining idle.

Fixtures are used to hold parts in a particular orientation to allow proper machining. A particular part type may require one or more fixture types. Furthermore, a part may be reoriented in the same fixture type for a different operation to be performed. In general, the part is clamped on to a fixture which is then bolted to a pallet. The whole unit is then mounted on a loading AGV which transports it to the appropriate machine. All the pallets used in a particular system are identical (Hartley, 1984). Thus, irrespective of the part type or the fixture type, any pallet can be loaded on to any AGV or machine table.

The coordination of all the activities of an FMS is under the control of a central computer or a group of computers. In situations where a group of computers are utilized, the computers are segregated into a hierarchy with proper communication links established between the different levels in the hierarchy. Some of the important functions performed by the computers include:

- production control
- traffic control
- loading part programs
- tool control
- system performance monitoring and reporting.

Most systems are provided with a central parts storage area to store raw material and work-in-process parts. The capacity of this storage place is quite large and vertical space is made use of to compensate for restrictions in the floor space. Cutting tools are stored in a tool crib (central tool storage) and are sent to the appropriate machines as and when required. Loading stations, where the parts are clamped on to the fixtures and pallets, and unloading stations, where the parts are dismantled from the fixtures and pallets, form an integral part of any FMS.

2.3 Configuration of the proposed FMS

From the basic description of an FMS given in Section 2.2, it is possible to have several manufacturing facilities referred to as 'FMS'. Therefore, it is important to define the configuration of the FMS assumed for the development of the simulation model.

Referring to Fig.2, it can be seen that the FMS consists of 'L' lanes (k=1,2,.....,L), each lane containing a certain number of machines. Thus, there are N_1 machines in Lane 1, N_2 machines in Lane 2, N_k machines in Lane k, and so on. The total number of machines, M, in the entire system is given by,

$$M = N_1 + N_2 + \dots\dots + N_k + \dots\dots + N_L$$

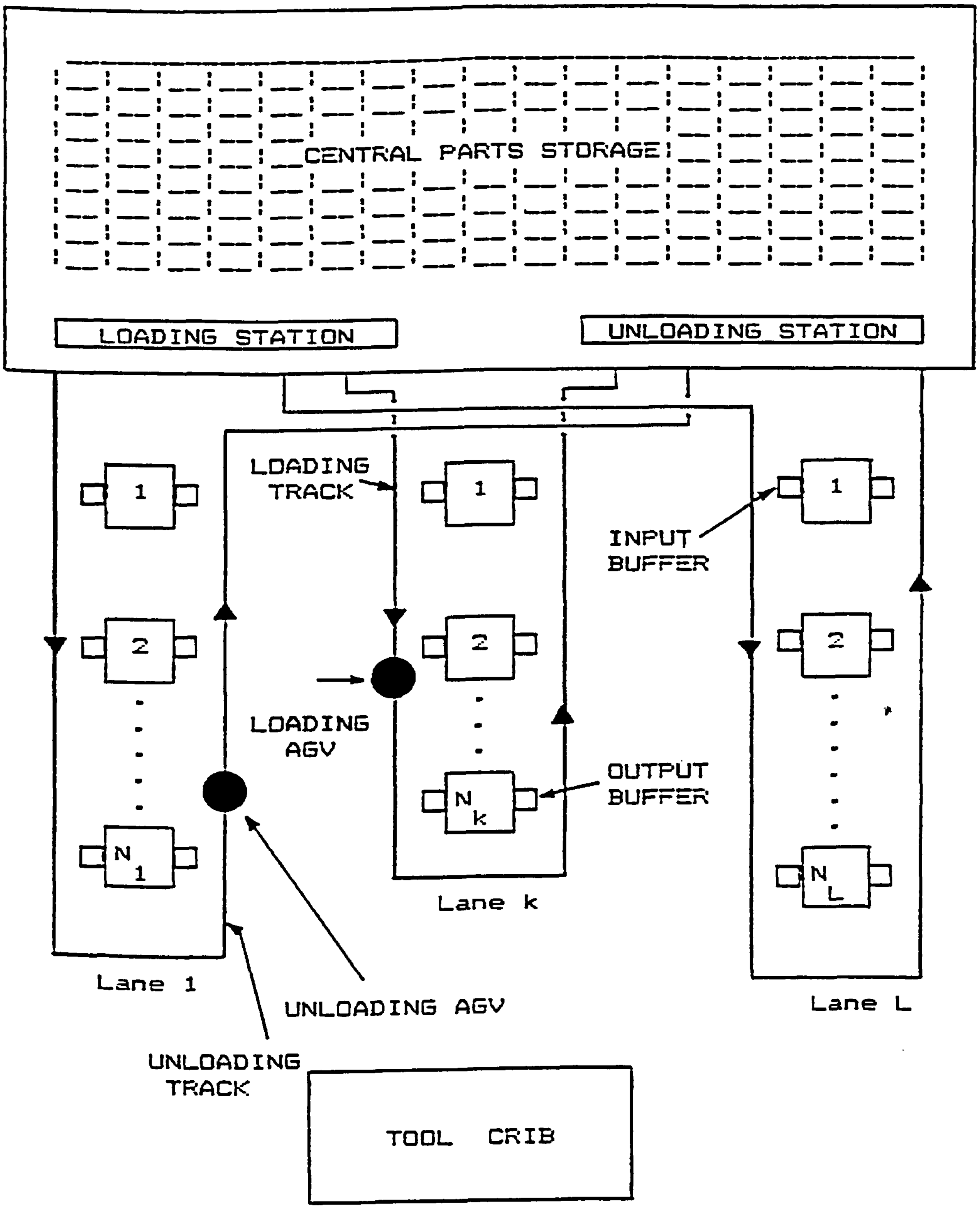


Fig.2 Layout of the FMS with L lanes (L>1)

These M machines are general purpose Horizontal Machining Centers (HMC) or Vertical Machining Centers (VMC). In spite of their versatility, it is not always possible for these machines to perform all the operations, partly due to the technological constraints, for example, the size of the machine table, the accuracy of the machine, etc., and partly due to the type of operation itself, for example, a vertical machining center is more suitable for drilling or reaming operations than the horizontal machining center.

A central parts storage that stores raw materials as well as the work-in-process parts forms a part of the system. To compensate for restrictions in floor space a vertical storage racking system is assumed. Such a centralised storage system is referred to as an Automatic Storage and Retrieval System (AS/RS) and provides greater flexibility in terms of scheduling the parts into the system based on the different rules (Ranky, 1986).

Each machine in the system is provided with an input buffer and an output buffer, each having the capacity to hold one pallet. The parts that are selected to be processed are transported from the central parts storage to the loading station by means of a suitable material handling system, usually a crane or a fork-lift truck. At the loading station they are clamped on to the appropriate fixture and then bolted to the pallet. The whole unit is then mounted on a loading AGV and transported to the respective machine. The parts that have completed an operation enter the unloading station where the entire unit consisting of the part, fixture, pallet and the unloading AGV is dismantled. A crane or a fork-lift truck transports the part from the unloading station to the central parts storage. The loading and unloading stations are manned. Hence, the assembling and dismantling operations are done by human operators.

Transportation of the parts from the loading station to the machines is done by one or more loading AGV's, while transportation of the parts from the machines to the unloading station is done by one or more unloading AGV's. Each lane is provided with a loading and unloading track on which the AGV's travel. The movement of the AGV's is controlled by the computer on a real time basis. The loading and the unloading track are laid out independently (Fig.2) and only one loading or unloading AGV is allowed to move on their respective tracks. This is done to prevent collisions and traffic congestions. A terminal at the loading station and another one at the unloading station provide information regarding the status of the loading or the unloading tracks. The loading and the unloading AGV's are not dispatched until the track on which they are to travel is clear.

The cutting tools are stored in the tool crib. A suitable material handling system, such as an overhead tool conveyor that does not interfere with the AGV's movements, transports the tools from the tool crib to the machines whenever a request for cutting tools is made. It is assumed that the requested cutting tools are always available at the tool crib.

The algorithm used to select the parts from the central parts storage and the algorithm that describes the flow of a part in the system are discussed respectively, in Section 4.1.1 and Section 4.2.2. The microcomputer-based simulation model was developed for the multi-lane type of FMS (Fig.2).

CHAPTER III

LITERATURE REVIEW

3.1 FMS problems

Flexible manufacturing systems are highly complex and dynamic in nature and hence pose a number of problems related to their design and operation. Stecke (1984) has identified problems which are frequently encountered once a decision to install an FMS has been taken. These problems are classified into four categories, namely:

(i) FMS design problems

(ii) FMS planning problems

(iii) FMS scheduling problems

(iv) FMS control problems

The first, second and fourth type of problems are often referred to as 'static' problems - the word static meaning one time allocation of the various resources; while the third type of problem is referred to as a 'dynamic' problem.

(i) FMS design problems

The FMS design problems are mainly concerned with the selection of the proper hardware. They include:

-selection of machines and equipment;

-determination of the various part types to be manufactured;

-determination of the layout of the machines, equipment, etc.

(ii) FMS planning problems

These involve certain decisons that are taken before the parts are loaded into the FMS for processing. They include:

-selection of a set of part types for immediate processing;

-the proportion at which the selected part types would be simultaneously processed in the FMS (part mix determination);

-the allocation of specific operations and tools to specific machines.

(iii) FMS scheduling problems

Scheduling is the allocation of jobs to be processed on specfic machines and is classified as:

(a) Off-line scheduling - Using this approach a scheduling algorithm is applied at the beginning of the scheduling period, to result in a complete schedule for that period.

(b) On-line scheduling (dynamic scheduling) - Using this approach the decision concerning the part to be processed next on a particular machine is made in real time at the termination of the operation currently being performed on that machine.

The FMS on-line scheduling problems are encountered when the system is operating and they include:

-determination of the next part to be processed on a machine;

-determination of alternate routing for a part.

(iv) FMS control problems

These are problems that arise with respect to keeping a constant check on the performance and quality standards of the machines. They include:

-design of preventive maintenance schedules for the machines and equipment;

-development of automatic inspection systems for parts and tools.

3.2 FMS models

A comprehensive review of the various techniques used to model flexible manufacturing systems have been provided by Wilhelm and Sarin (1983). These techniques are:

(i) Queueing network (analytical modeling)

(ii) Mathematical programming

(iii) Simulation.

Queueing network models have been based on the research initiated by Jackson (1957), in which network models were decomposed and analysed as a set of independent service stations. Based on the queueing network models of Baskett, et al. (1975) and the computer algorithms of Buzen (1973), Solberg (1977) developed a computer program (CAN-Q) to model an FMS. Buzacott and Shantikumar (1980) extended this methodology to study the part selection problem and evaluate alternative plans for work-in-process inventory. These network models, which provide the researchers with average performance measures of the system, have been applied mainly to solve the planning problems of an FMS. Realistic modeling of an FMS using this approach is severely restricted because the existing models fail to incorporate the various realistic features of an FMS such as, limited buffer capacities at the workstations, complex server disciplines, etc.

Mathematical programming techniques, e.g., integer programming, have been applied in certain static decision models. A set of 0-1 integer programming formulations were developed to evaluate the loading aspects of an FMS (Stecke, 1981). Kusiak (1983) proposed a few integer programming formulations of the loading problems. An optimization approach to determine the optimal part routing policy for an FMS has been proposed by Kimemia & Gershwin (1985). Shanker and Tzen (1985) developed a mathematical model to study the loading and dispatching problems of a randomly generated FMS. Solutions from this mathematical model were used as inputs to a simulation model.

Though mathematical programming techniques are attractive in terms of their application to solve static problems of small and medium size, their application to large sized problems is restricted. This is due to the large formulations that are difficult to solve with the existing IP algorithms and packages. Moreover, certain constraints that are characteristic of an FMS (for example, tooling constraints) are difficult to consider due to the complexity involved in the formulations.

Simulation models have had the widest scope of application, partly due to the paucity of other types of models, and partly due to their ability to incorporate the realistic features of an FMS. Simulation is more suited for investigating the operational aspects of these complex manufacturing systems (Law, 1986). Stecke and Solberg (1981) explored alternative loading and scheduling strategies for the system installed at the Caterpillar Tractor Company. Due to the complexity of the various system aspects, analytical treatment was not viable and simulation was resorted to. Of the sixteen priority rules that were investigated, the weighted shortest processing time was found to produce the best results for this system in terms of the number of completed parts. Though the results from this study cannot be extrapolated to other systems that have different configurations and process different part types, it demonstrated the usefulness of simulation in evaluating different scheduling strategies. ElMaraghy (1982) developed a special simulator called FMSSIM (Flexible Manufacturing Simulator) to analyse flexible manufacturing systems during their design phase. The simulation model was made flexible enough to permit variations in the quantities of the various resources used by the system. The model served as a decision tool in deciding the quantities of the different resources for various systems processing different part types.

Parallel to the mathematical modeling of the FMS installed at the SCAMP Systems Ltd, Rathmill, et al. (1983) developed a simulation model to provide a detailed investigation of the material handling, conveying and overall workflow aspects of the system. Acree (1983) studied various part, tool, and cart scheduling rules for a system that was installed by the military. One of the important findings of this study was the superiority of individual tool allocation to the machines as and when they are needed, as opposed to total tool allocation. This study was system specific and since it was done for the military, very little information has been made available to the public.

Carrie, et al. (1984) used a simulation model to analyse the sensitivity of system performance to variations in part mix, operation times and facility reliability for the system installed at Anderson Strathclyde, Scotland. Wortman and Miner (1984) developed MAP/1, a simulator specifically designed for modeling discrete manufacturing systems. Using this simulator they evaluated an FMS design in which the variables analysed were the number of dedicated and flexible machines and the number of fixtures and carts. Another application of the MAP/1 simulator was in deciding the number of fixtures for a machine tool manufacturer. Chang, et al. (1986) used the general purpose simulation language, SLAM (Pritsker, 1984), to design the material handling system of an FMS. This study demonstrated the flexibility of SLAM as compared to that of Q-GERT (Pritsker, 1977) or GPSS (Schriber, 1974), in terms of the modeling and output analysis.

Since an FMS consists machines which are flexible enough to perform a wide variety of operations, certain operations on certain part types could be processed by different machines. This gives rise to the concept of alternate routings. However, this inherent feature of an FMS, has not been treated by the simulation models appearing in the literature.

The efficient operation of an FMS is dependent, to a certain extent, on the availability of the various resources like pallets, carts, human operators, etc. Though a major portion of the work carried out so far has dealt with the issue of deciding the number of pallets or carts and the allocation of these resources to different parts, there has been no report of any work that has considered the fixtures as a constraining resource. With a considerable increase in the number of random systems, which machine a larger variety of parts as compared to dedicated systems, the use of standardised pallets to hold the parts in proper orientation for the machining to be performed is extremely difficult. This has led to the development of fixtures to interface the part and the pallet. These fixtures are expensive due to the precision machining involved and are available in limited quantities. Even with a considerable increase in the use of modular fixtures, which are assembled from standard components (Drake, 1984), the availability of these standard components result in a limited quantity of the fixtures. Moreover, a part that has been selected to enter the system would be prevented from doing so unless the required fixture type is available, in spite of the availability of the other shared resources like pallets and the AGV's. Hence, fixtures are important resources and should be considered during the development of the model.

In dedicated flexible manufacturing systems the variety of parts being processed is small and the batch sizes of the various part types are large enough to maintain a constant number of parts of each type as long as the set of part types remain unchanged. Much research has been done in the past to formulate models that determine the proper part mix - the proportion of the various part types in the system - to be maintained in the system at all times. The part mix determination problem controls the scheduling of parts into the system, since a part that leaves the system is always replaced by another one of the same type, to maintain the same proportion. However, as mentioned earlier, with a considerable increase in the number of random systems that are capable of processing a larger variety of parts in smaller batch sizes (say, 5 or less), the determination of a suitable part mix is highly inappropriate. Such systems are similar to the typical job shop systems in which batch sizes of one are not uncommon and the scheduling of parts into the system has to be based on some criterion. The complex nature of these systems rule out the possibility of the development of off-line scheduling algorithms to input parts into the system. Moreover, inclusion of the realistic features, e.g., alternate routings, etc.

complicate these off-line algorithms. On-line scheduling, or dynamic scheduling, which essentially depends on the state of the system, is more suited to such systems. The complexities involved in the development of the queueing network and mathematical models to address the problem of on-line scheduling along with the consideration of the realistic features such as, alternate routing and the treatment of fixtures as resources, justify the use of a simulation approach.

3.3 Objectives of the research

This research deals with the development of a simulation model of the FMS described in Sec 2.3 which can be used as a decision tool to select a scheduling rule from the five rules under consideration.

Five simple priority scheduling rules considered in this thesis, based on the part characteristics, are:

(i) Random selection rule (RANDOM)
(ii) Fewest operations remaining (FOPR) rule
(iii) Most operations remaining (MOPR) rule
(iv) Shortest processing time (SPT) rule
(v) Longest processing time (LPT) rule

The simulation model which would incorporate features such as, alternate routings, treatment of fixtures as resources and part types with priorities, is to be developed on the microcomputer using the general purpose simulation language, SLAM. The simulation model is to be user-interactive so that it does not require a prior working knowledge of SLAM. This front-end interface between the user and the computer is to provide flexibility in using the model.

CHAPTER IV

SIMULATION MODEL OF THE FMS

The development of the simulation model of the FMS under consideration is discussed in this chapter. The operation of the FMS, namely, the part-machine selection algorithm and the flow of the part in the system is explained in Section 4.1.1 and Section 4.1.2, respectively. The development of the different computer programs which together form the simulation model of the FMS is explained in Section 4.2. In Section 4.3 the techniques that were used to verify the model are outlined.

4.1 Mode of operation of the FMS

The conversion of a part type from the raw material state (generally, castings or forgings) to a fully machined product involves a series of processing steps. These processing steps are generally combined into one or more operations. Each operation requires one or more specific cutting tools. The procedure of combining these processing steps into operations is usually done by an experienced process planner and a certain degree of objectivity is involved.

As an example, the development of the operations sheet for the sample prismatic part shown in Fig.3 is given below.

Processing steps	Cutting tools
1. ROUGH MILL BOTTOM FACE	FACE MILLING CUTTER (M1)
2. FINISH MILL BOTTOM FACE	FACE MILLING CUTTER (M2)
3. ROUGH MILL TOP FACE	FACE MILLING CUTTER (M1)
4. FINISH MILL TOP FACE	FACE MILLING CUTTER (M2)
5. MILL DOVETAIL SLOT	SPECIAL CARBIDE TIPPED MILLING CUTTER (M3)
6. DRILL CENTER HOLE (10mm)	DRILL - 10mm DIA (D1)
7. ENLARGE CENTER HOLE to 20mm	DRILL - 25mm DIA (D2)
8. REAM CENTER HOLE (20mm)	REAM - 20mm DIA (R1)
9. ROUGH BORE CENTER HOLE	BORING TOOL (B1)
10. FINISH BORE CENTER HOLE	BORING TOOL (B2)

These processing steps can be combined into two different sets of operations as shown below.

Operation set # 1	Operation set # 2
Operation # 1 : Steps 1,2	Operation # 1 : Steps 1,2
Operation # 2 : Steps 3,4,5	Operation # 2 : Steps 3,4
Operation # 3 : Steps 6,7,8	Operation # 3 : Steps 5
Operation # 4 : Steps 9,10	Operation # 4 : Steps 6,7,8
	Operation # 5 : Steps 9,10

In this model it is assumed that each operation on a part type is done in a single setup on a machine. After the completion of an operation, a part returns to be refixtured or reoriented in the same fixture for the next

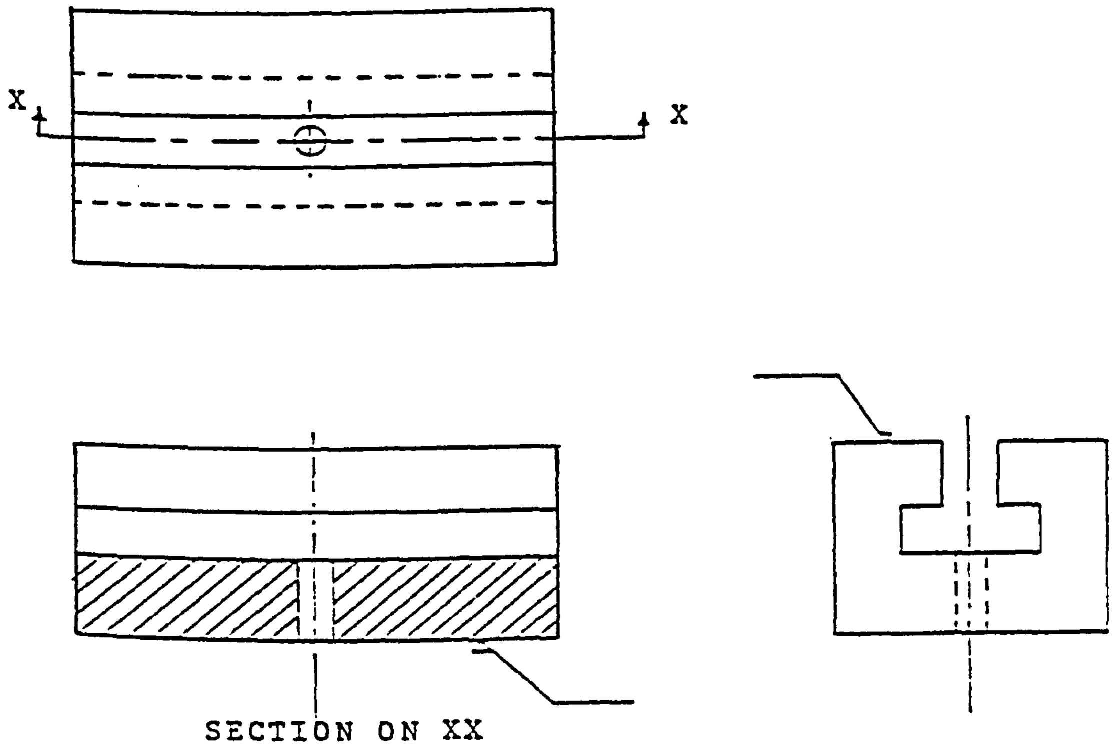


Fig.3 Sample prismatic part

operation. Hence, the number of visits made by a part into the system is equal to the number of operations to be performed on it.

The mode of operation of the FMS under consideration is explained by dividing it into two stages. The first stage deals with the selection of a part from the parts in the central parts storage based on the scheduling rule that is followed. The second stage deals with the actual flow of the selected part in the system.

4.1.1 Part-machine selection algorithm

Parts are selected from the central parts storage based on the scheduling rule being followed. Though this appears simple initially; due to the unavailabiltity of the right fixture and machine(s) for the operation of the selected part, and the presence of parts that have priority over other parts, it is not always possible to adhere to the rule being followed. In order that the system is always loaded with parts, a compromise has to be made when the scheduled part cannot enter the system. The algorithm for selecting a part from the central parts storage under such conditions is explained below. This part-machine selection algorithm consists of a set of blocks, A through P. The flow chart of this algorithm is shown in Fig.4.

ALGORITHM:

(A) When the input buffer of a machine(s) becomes available, the central parts storage is requested to send a part for processing. Go to block (B).

(B) If all the parts in the central parts storage have been considered, then go to block (P), otherwise go to block (C).

(C) If there are parts in the central parts storage with priority then go to block (D), otherwise go to block (N).

(D) A set of parts that have priority over other parts is formed. Go to block (E).

(E) A part is selected from this set based on the scheduling rule being followed. This part is called as the candidate part. Go to block (F).

(F) The operation to be performed on this part type is noted and the availability of the fixture type for this operation is checked. If the fixture type is available, then go to block (G), otherwise go to block (K).

(G) If the candidate part could be processed on the machine(s) whose input buffers are empty, then go to block (H), otherwise go to block (K).

(H) If candidate part could be processed on more than one machine, then the machine whose total distance from the loading and unloading stations is the smallest is selected to process the part. Go to block (I).

(I) This candidate part is then sent to the selected machine for processing. Go to block (J).

(J) If there are any more input buffers that are empty, then go to block (A), otherwise go to block (P).

(K) This candidate part is removed from further consideration. Go to block (L).

(L) If there are any parts in the central parts storage with priority, then go to block (M), otherwise go to block (N).

(M) A new set of parts, excluding the candidate part, is formed from the set of parts that have priority over other parts. Go to block (E).

(N) If there are parts in the central parts storage without priority, then go to block (O), otherwise go to block (P).

(O) A set of parts without priority is formed. Go to block (E).

(P) The selection algorithm is stopped.

4.1.2 Part flow in the system

Based on the algorithm in Section 4.1.1, a part is

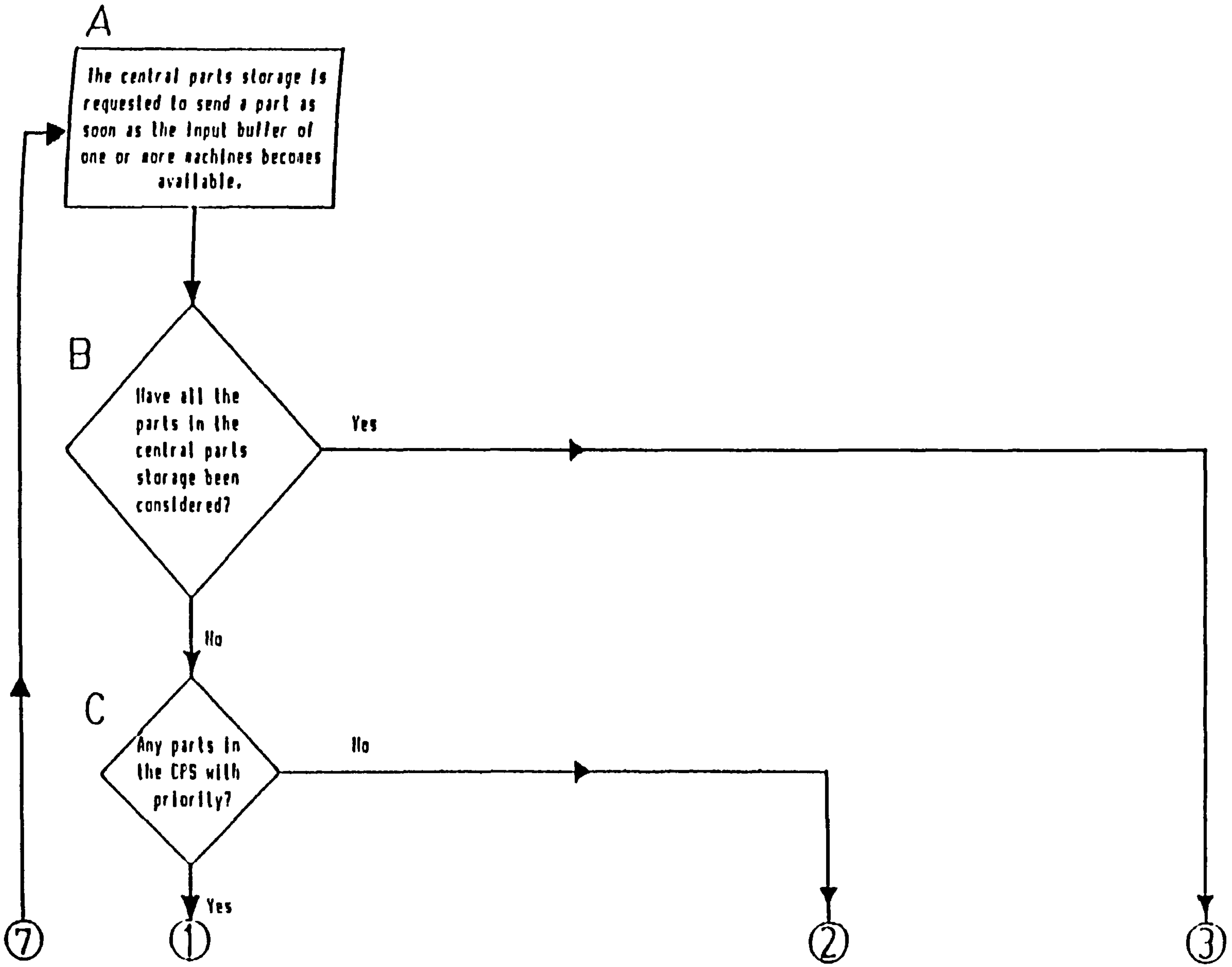

A
The central parts storage is requested to send a part as soon as the input buffer of one or more machines becomes available.
B
Have all the parts in the central parts storage been considered?
Yes
No
C
Any parts in the CPS with priority?
No
Yes
⑦
①
②
③

Fig.4 Flow chart of part-machine selection algorithm

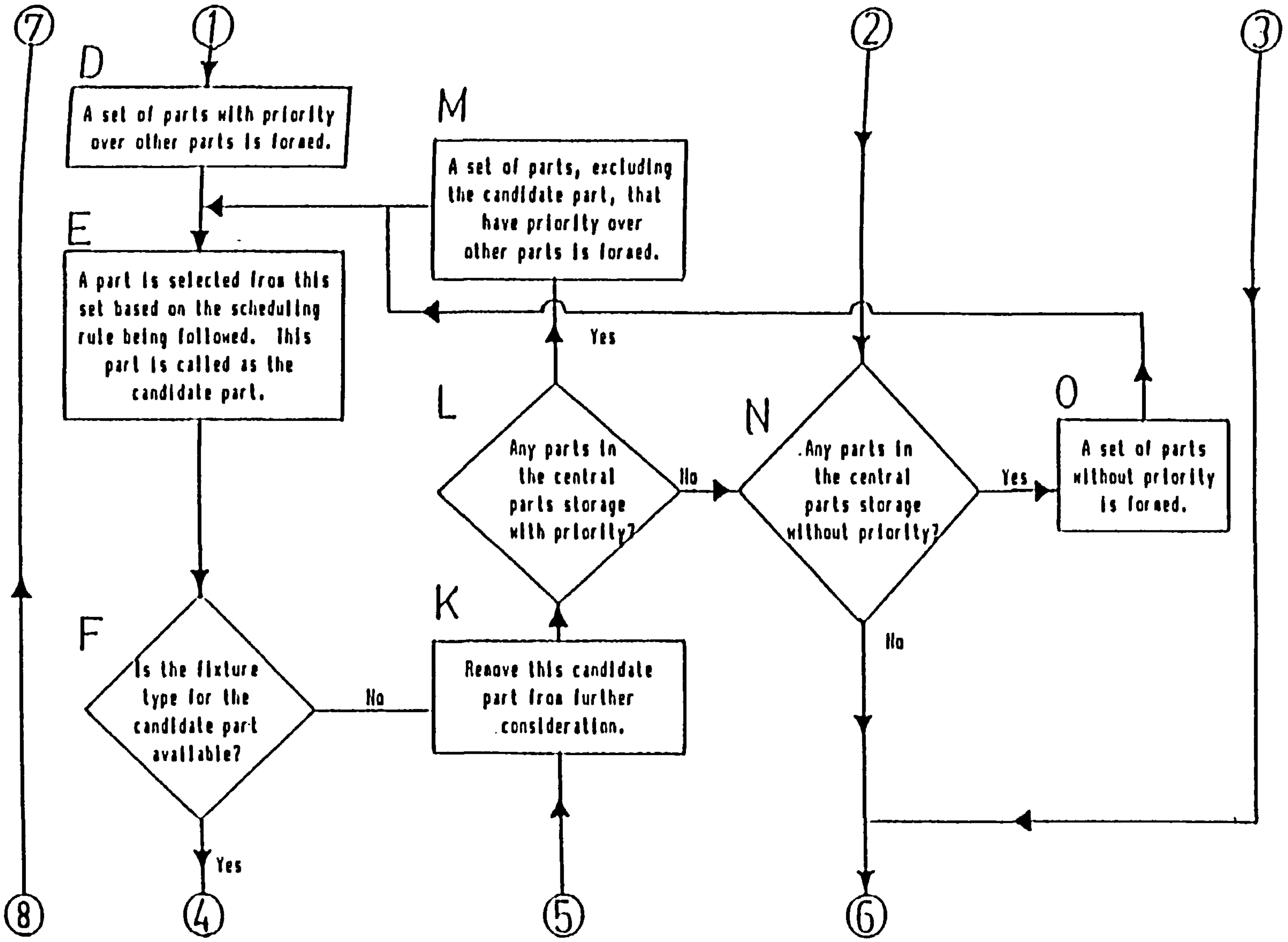

7
1
2
3
D
A set of parts with priority over other parts is formed.
M
A set of parts, excluding the candidate part, that have priority over other parts is formed.
E
A part is selected from this set based on the scheduling rule being followed. This part is called as the candidate part.
Yes
L
Any parts in the central parts storage with priority?
No
N
.Any parts in the central parts storage without priority?
Yes
O
A set of parts without priority is formed.
K
F
Is the fixture type for the candidate part available?
No
Remove this candidate part from further .consideration.
No
Yes
8
4
5
6

Fig.4 (continued)

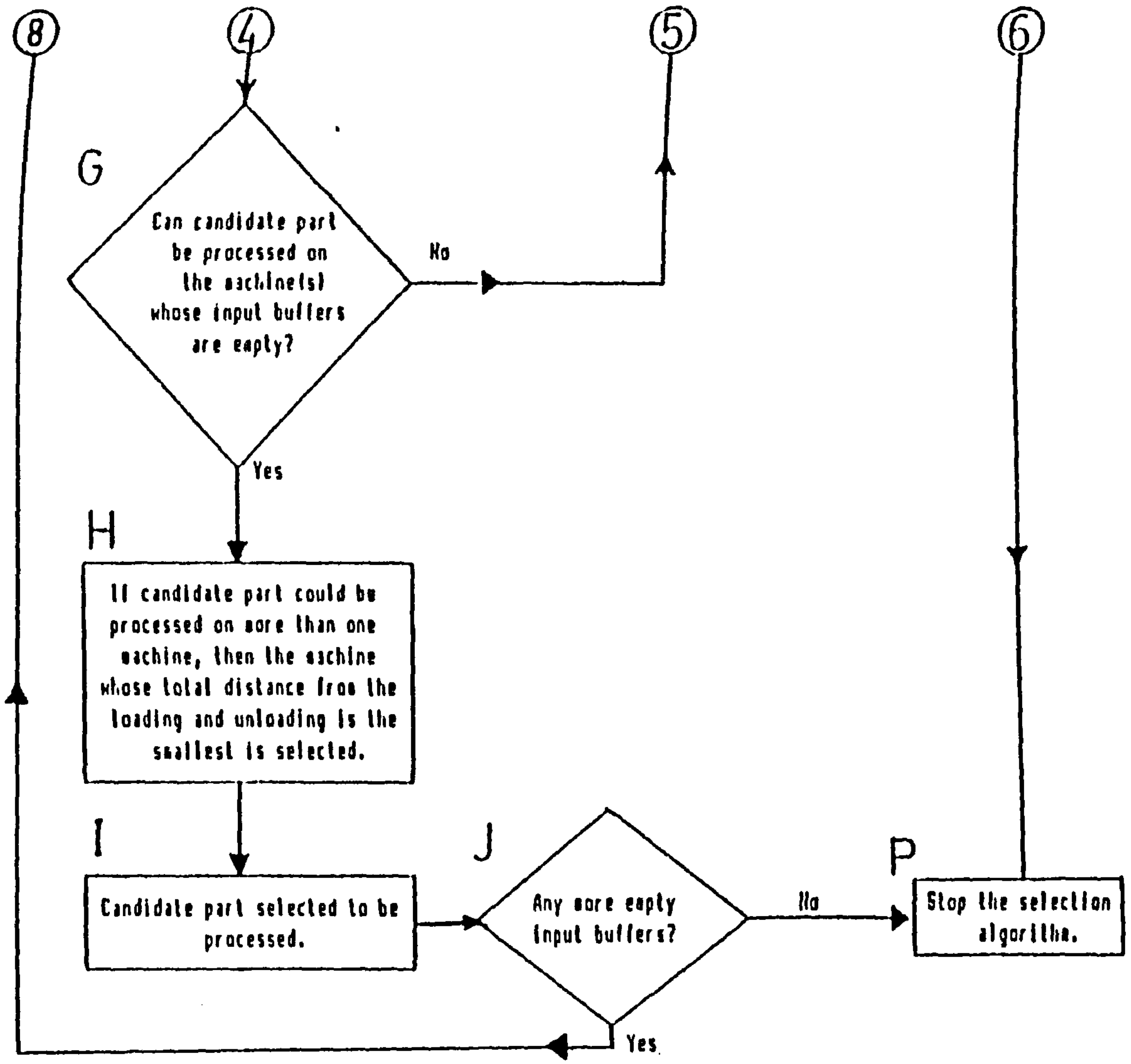

8
4
5
6
G
Can candidate part be processed on the machine(s) whose input buffers are empty?
No
Yes
H
If candidate part could be processed on more than one machine, then the machine whose total distance from the loading and unloading is the smallest is selected.
I
Candidate part selected to be processed.
J
Any more empty input buffers?
No
Yes
P
Stop the selection algorithm.

Fig.4 (continued)

selected to be processed on a particular machine. The flow of this selected part in the system is explained below and the flow chart is shown in Fig.5.

ALGORITHM:

(i) The selected part waits in the central parts storage until the loading crane, loader, loading AGV and pallet are available. Once these resources are available, go to step(ii).

(ii) The part is transported to the loading station by the loading crane. At the loading station the part is clamped in the appropriate fixture and then bolted on to a pallet. The entire unit is then mounted on a loading AGV.

This loading AGV is kept waiting at the loading station until the loading track is clear. Once it is clear, go to step (iii).

(iii) The unit is transported to the machine and then transferred to the input buffer of the machine. The unit is kept waiting in the input buffer until the machine table is empty. Once it is empty, go to step (iv).

(iv) The unit is transferred to the machine table. A check is made to see if all the cutting tools that are required for this operation are available in the tool magazine.

A request is made to the tool crib to send the necessary cutting tools that are not available. The processing on the parts is not started until all the cutting tools are available. Once all tools are available, go to step (v).

(v) The processing steps for the operation on the part are carried out. After the completion of the operation, the unit is kept waiting on the machine table until the output buffer of the machine is empty. Once it is available, go to step (vi).

(vi) The unit is transferred to the output buffer of the machine. The unit is kept waiting on the output buffer until the unloading crane, unloader and unloading AGV are available. Once these resources are available, go to step (vii).

(vii) The unloading AGV is kept waiting at the unloading station until the unloading track is clear. Once it is clear, go to step (viii).

(viii) The unloading AGV is then sent to the machine to pick up the unit and transport it to the unloading station. At the unloading station the unit is removed from the unloading AGV and dismantled. The part is then transported to the central parts storage by the unloading crane. Go to step (ix).

(ix) If all the operations on the part have been completed, then the part does not enter for further processing, otherwise the part stays in the central parts storage.

The travel times from the central parts storage to the loading station, from the loading station to the machines, from the machines to the unloading station, and from the unloading station to the central parts storage, are deterministic because of the use of automated material handling systems that result in constant times. The use numerically controlled machines results in processing times that are deterministic. However, the loading times at the loading and unloading stations are stochastic since the loading and unloading operations are performed by human operators. As a result of this, the time spent in waiting by the parts for the different resources and the machines are of a random nature. Hence, the output from the simulation model is stochastic.

4.2 Development of the simulation model

The simulation model of the proposed FMS, whose configuration was described in Section 2.3 and whose mode of operation was described in Section 4.1, was developed on an IBM/PC microcomputer using SLAM. SLAM is a general purpose simulation language that is used to develop computer models of discrete change or continuous change

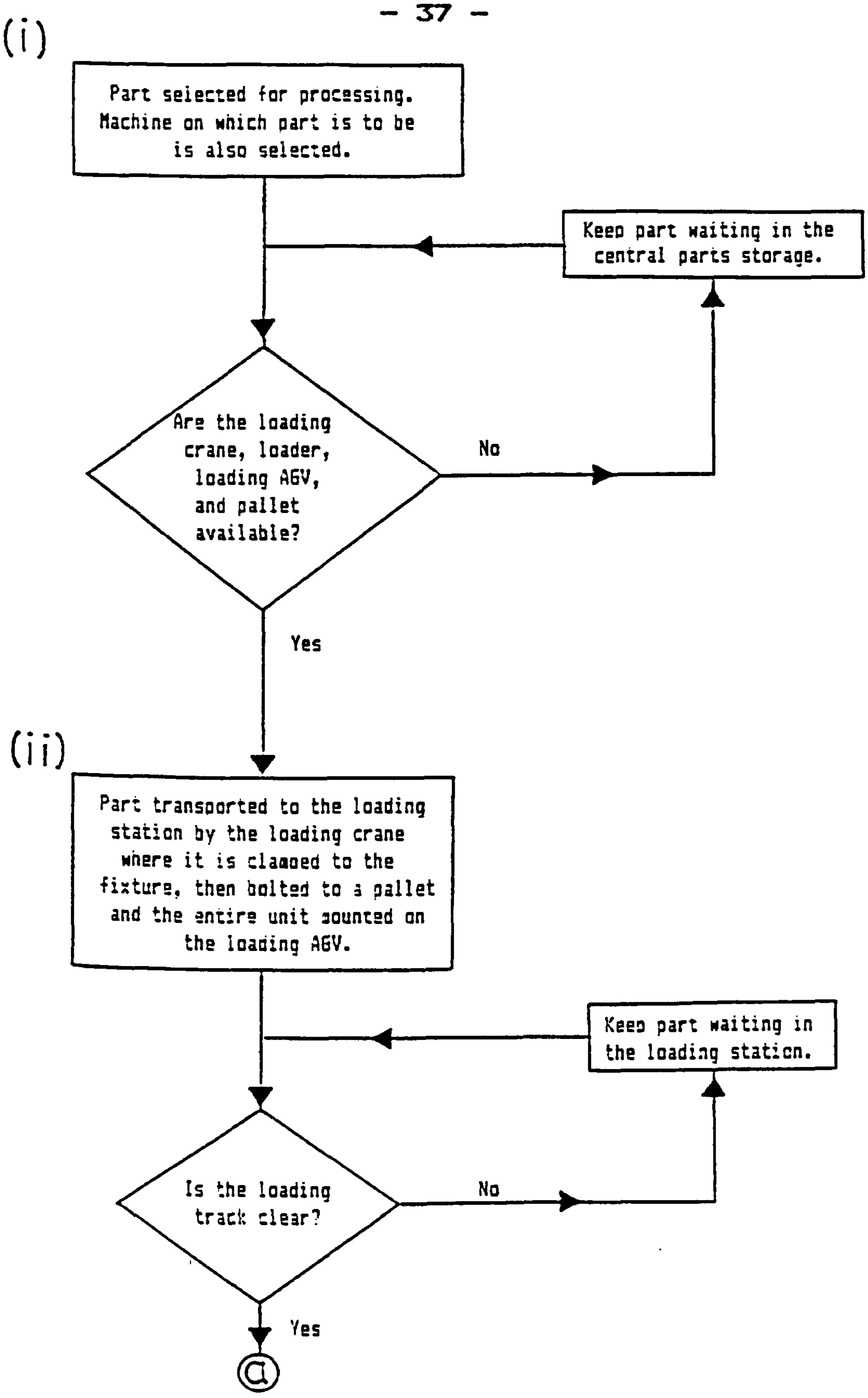

(i)
Part selected for processing. Machine on which part is to be is also selected.
Keep part waiting in the central parts storage.
Are the loading crane, loader, loading AGV, and pallet available?
No
Yes
(ii)
Part transported to the loading station by the loading crane where it is clamped to the fixture, then bolted to a pallet and the entire unit mounted on the loading AGV.
Keep part waiting in the loading station.
Is the loading track clear?
No
Yes
a

Fig.5 Flow chart of part flow in the system

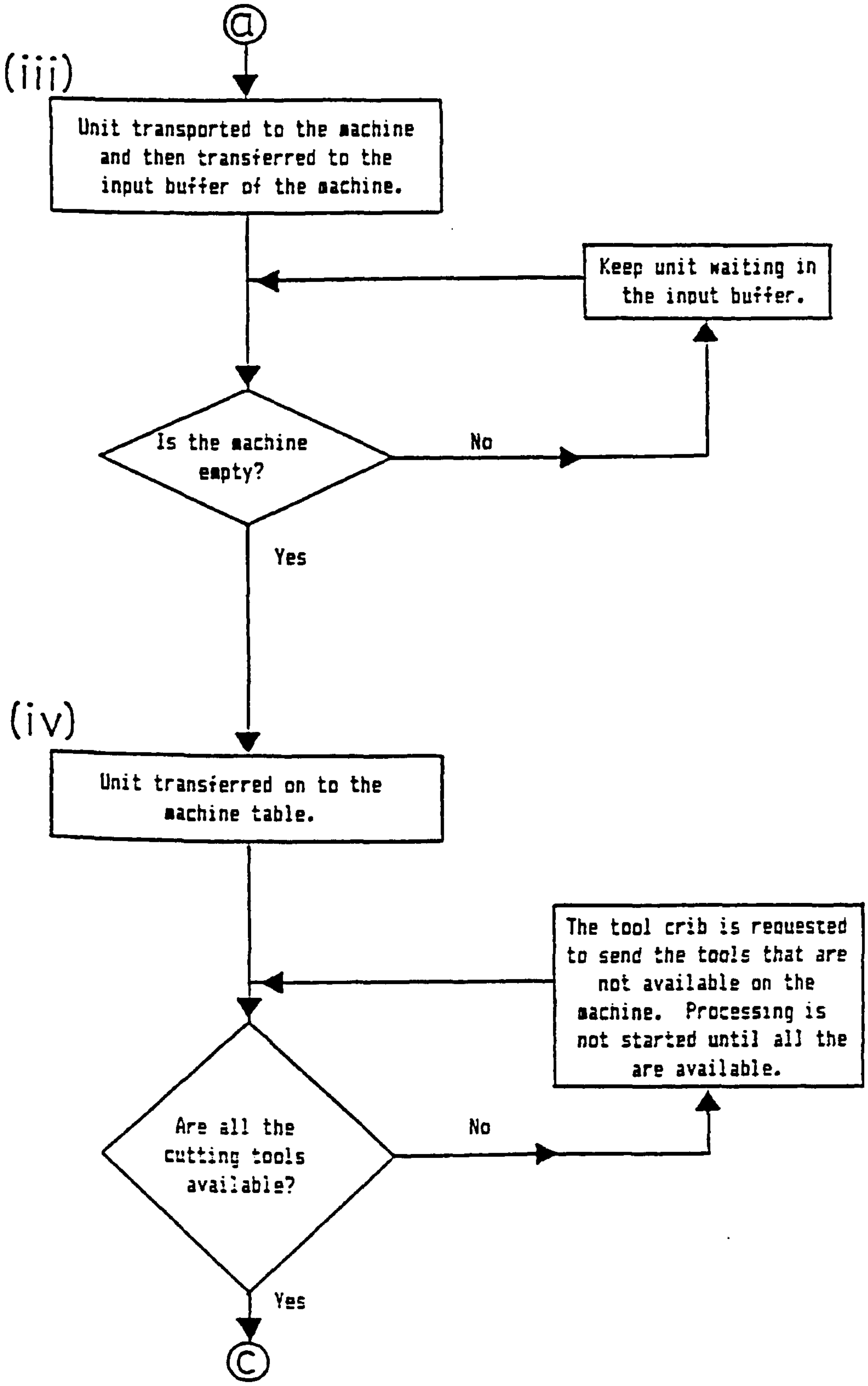

a
(iii)
Unit transported to the machine and then transferred to the input buffer of the machine.
Keep unit waiting in the input buffer.
Is the machine empty?
No
Yes
(iv)
Unit transferred on to the machine table.
The tool crib is requested to send the tools that are not available on the machine. Processing is not started until all the are available.
Are all the cutting tools available?
No
Yes
C

Fig.5 (continued)

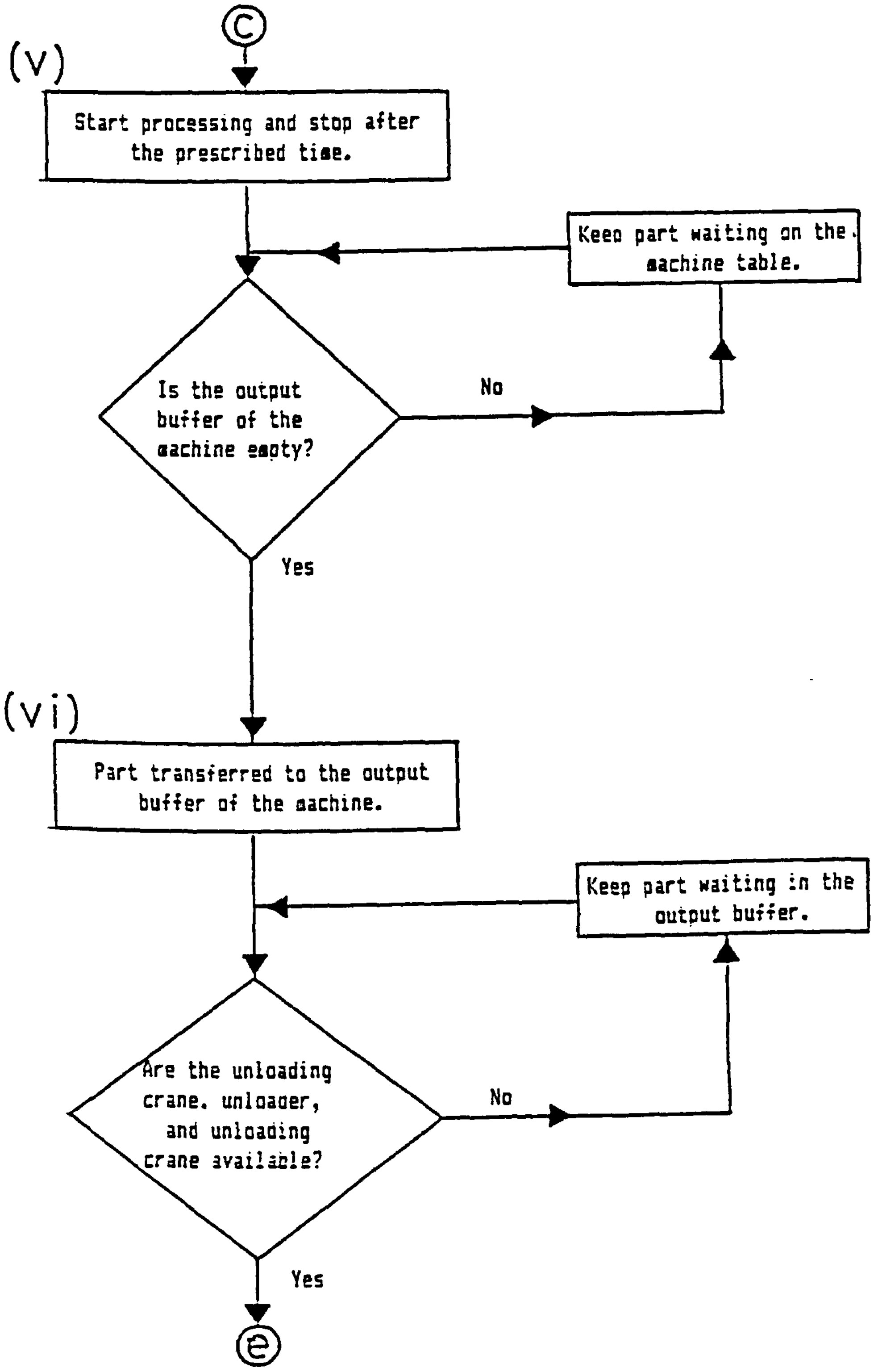

C
(v)
Start processing and stop after the prescribed tiae.
Keep part waiting on the aachine table.
Is the output buffer of the aachine eapty?
No
Yes
(vi)
Part transferred to the output buffer of the aachine.
Keep part waiting in the output buffer.
Are the unloading crane, unloader, and unloading crane available?
No
Yes
e

Fig.5 (continued)

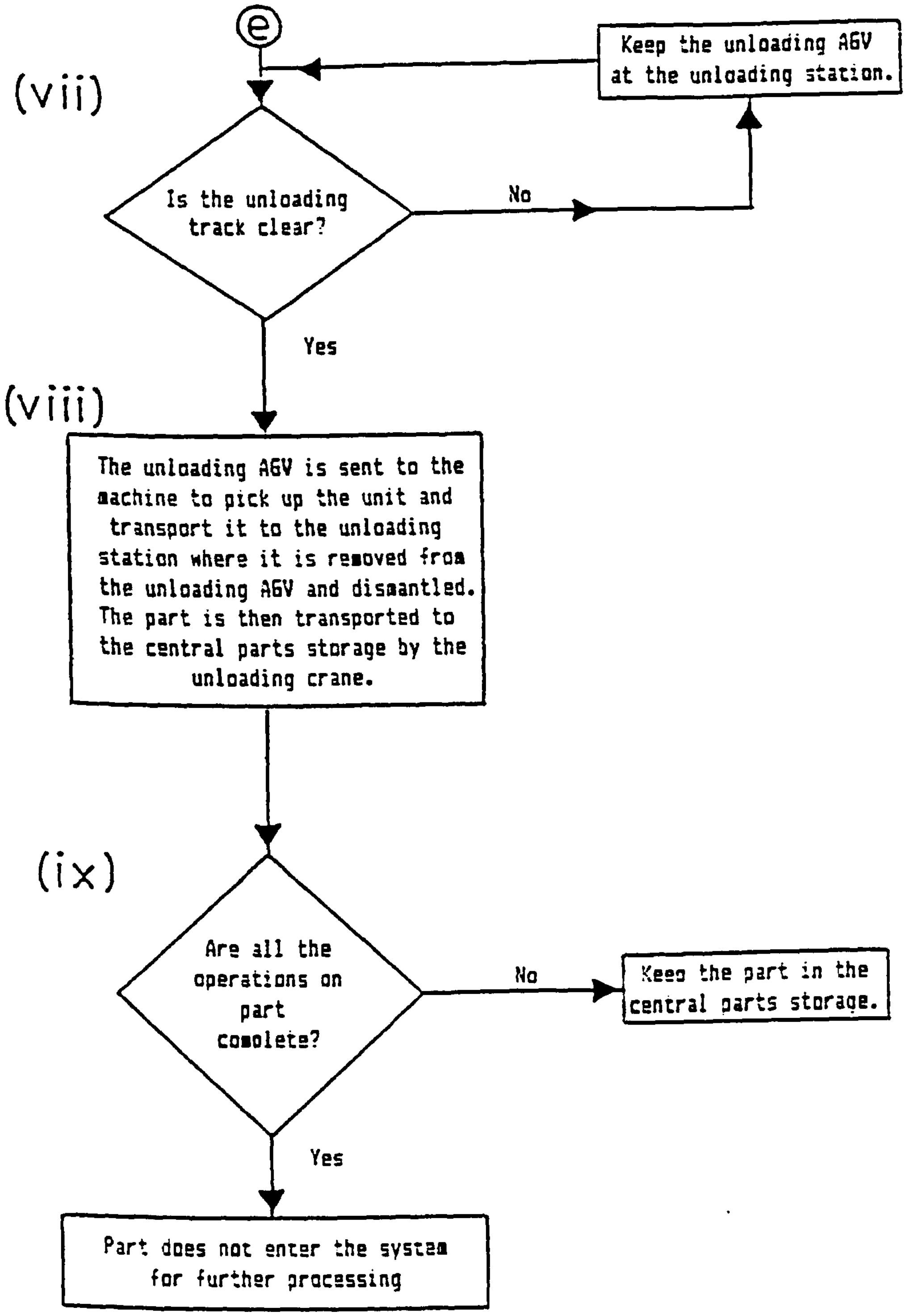

e
Keep the unloading AGV at the unloading station.
(vii)
Is the unloading track clear?
No
Yes
(viii)
The unloading AGV is sent to the machine to pick up the unit and transport it to the unloading station where it is removed from the unloading AGV and dismantled. The part is then transported to the central parts storage by the unloading crane.
(ix)
Are all the operations on part complete?
No
Keep the part in the central parts storage.
Yes
Part does not enter the system for further processing

Fig.5 (continued)

systems. The microcomputer version of SLAM is based on Microsoft Fortran (MS FORTRAN).

The FMS under consideration can be modeled as a discrete change system since the state variables that describe the system, e.g., the number of parts inside the system, the number of busy machines in the system, etc., change at discrete points in time. Discrete change systems are modeled in SLAM using one of the following three approaches (Pritsker, 1984):

(i) Process orientation (Network modeling)

(ii) Event orientation

(iii) Combined process-event orientation.

The combined process-event orientation approach was used to develop the simulation model of the proposed FMS. This approach combines the advantages of the process orientation and the event orientation approaches. Using this approach the entire system was modeled using a network model and at instances where increased flexibility is required, the event orientation approach was used.

The software that was written was divided into two major programs (Fig.6), namely:

(i) Program 1 - This program creates the data files that are later accessed by the SLAM program as the simulation progresses.

(ii) Program 2 - This is the main simulation program written in SLAM. The program is further subdivided into

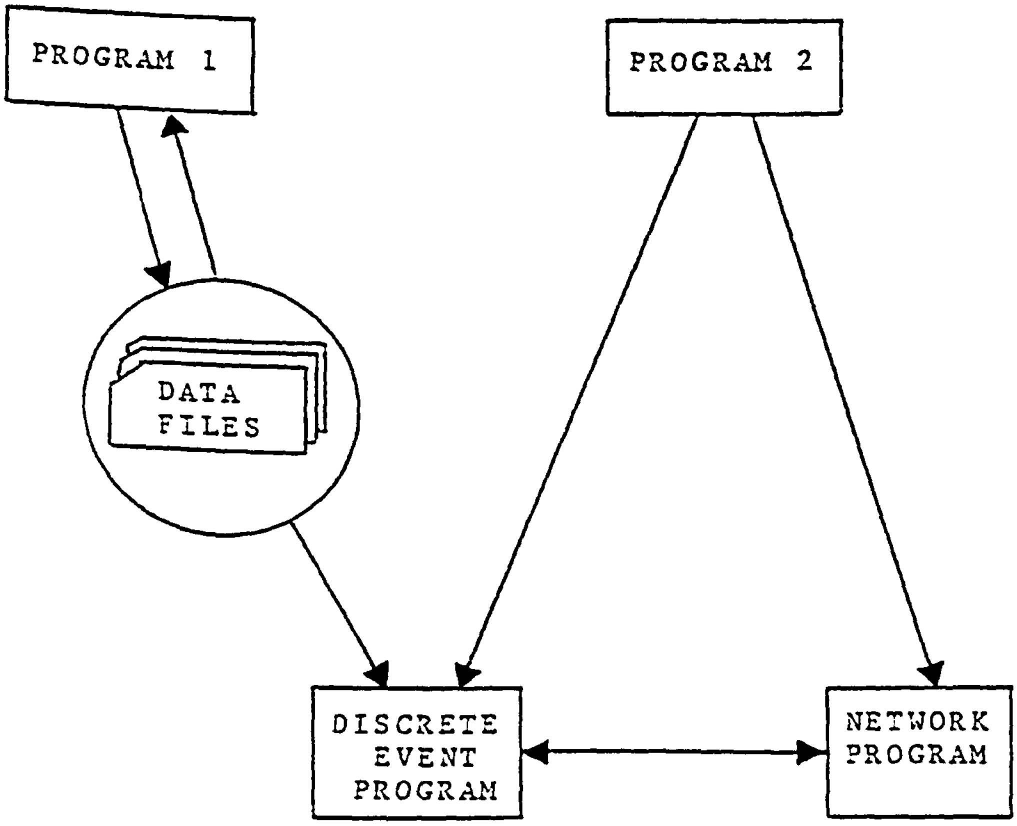


Fig.6 Organisation of the computer program

two different modules, namely:

(i) SLAM discrete event program

(ii) SLAM network program.

4.2.1 Program 1 - Input data creation

For each part type, its priority, its batch size, the number of operations, the fixture type for each operation, the machine(s) on which each operation could be performed and the cutting tools required for each operation have to be provided as input data. Moreover, the number of machines in the system, the distances of these machines from the loading station, the unloading station and the tool crib, and the speed of the different material handling systems form part of the input data. As a result of this, it is a cumbersome procedure to input data whenever a scheduling rule is analysed or the resource levels are changed. In order to prevent this inconvenience to the user, the above data is stored in different data files which are later accessed by the simulation program. A user interactive program was written in Microsoft Advanced Basic (MS BASICA) which creates these different data files as the user responds to a series of queries during the execution of the program. This program is a menu driven program and consists of a set of subprograms that are used to:

(i) create new data files

(ii) review and alter existing data files

(iii) add details to the existing data files

(iv) print out the details of existing data files.

A computer listing of this program is given in Appendix-A.

4.2.2 Program 2 - SLAM simulation program

Since a combined network-discrete event modeling approach was followed, the simulation program in SLAM consists of two modules, namely:

(i) SLAM discrete event program

(ii) SLAM network program.

The SLAM network program is used as a driver program for the simulation. The flexibility in terms of the part-machine selection algorithms, the multiple resource allocation procedures and the automated material handling equipment travel times are provided by the SLAM discrete event program.

4.2.2.1 SLAM discrete event program

This program (Appendix-B) was written in MS FORTRAN since SLAM is a Fortran based simulation language. A modular approach was followed during the development of the program which resulted in the entire program being organised into a number of subroutines.

The input data that is required by the simulation program is loaded into the computer memory by the MAIN routine. This routine calls the SLAM execution processor to start the simulation. Commands that require the user to input the scheduling rule, the number of runs, the names of the files from which the input data has to be retrieved and the quantities of the various resources to be used, on an interactive basis, are contained in this routine (Appendix-C).

The subroutine (INTLC) initialises the simulation to the same initial conditions at the start of each run when multiple runs are used. Part selection based on the scheduling rule followed, checking the tools in the tool magazines of the machines, deciding the re-entry of a part into the system, collecting data on utilizations of the resources and the machines, and printing out the results are some of the important EVENT routines that are employed in the program. These routines are called when a part enters the corresponding EVENT node in the network.

Allocation of the multiple resources is handled by ALLOC(I) subroutine that is invoked as a part enters the AWAIT node in the network model. The USERF(I) subroutine returns the value of the travel times of the different material handling equipment when a part passes through the USERF branch.

4.2.2.2 SLAM network program

The network model of the FMS consists of the SLAM network symbols, namely, the nodes and branches. The program consists of statements that are classified as control statements and network statements. The control statements are used to control the number of simulation runs, to initialise the seeds at the start of each run, to store the entities in files based on certain priorities, to start and end the simulation, etc. The network statements represent the various nodes and branches used in the network model. The SLAM network program, whose listing is given in Appendix-D, consists of a number of segments as shown in Fig.7.

Segment 1 (Fig.8) - The CREATE node starts the simulation and the EVENT node (EVE1) selects the parts based on the rule being followed. The selected part waits for the following resources - the loading crane, loading AGV, loader and pallet - at the AWAIT node (MULT). Once these resources are available, they are allocated to the first part waiting at this node.

Segment 2 (Fig.9) - The transportation time from the parts storage to the unloading station is provided by the USERF routine (USERF(1)). The loading time at the loading station is represented as a service activity. Due to the stochastic nature of the loading activity, the loading time follows a probability distribution. Depending on the system under consideration and the distribution that is appropriate, the exact function that generates random variables for this distribution with the assumed parameters can be used in the corresponding network statement. After the unit has been mounted on the loading AGV, the loading crane and the loader are freed by the FREE nodes (FCRA and FLOA). The loading AGV waits in the AWAIT node (AW25) until the loading track on which it is to travel is clear. The ASSIGN node (AS20) increments the number of parts being processed (XX(20)) every time a part flows through this node. The travel time from the loading station to the machine is returned by the USERF routine (USERF(2)). At the machine the pallet is transferred from the loading AGV to the input buffer of the machine. This transfer time is very small and hence assumed to be zero. The travel time of the loading AGV from the machine back to the loading station is returned by USERF(3). The EVENT node (EVE2) releases the loading AGV and the loading track, i.e., it indicates that the loading track is clear, while EVENT 3 (EVE3) is used to check for the tools in the tool magazine of the machine and make a request to the tool crib to send the required tools that are not available in the tool magazine of the machine.

Segment 3 (Fig.10) - The input buffers of the machines are represented as AWAIT nodes (BUF1 to BUF9). The machines are treated as resources and the part waits in the appropriate AWAIT node for the machine (as an example, BUF1 for machine 1). The capacity of each input buffer is set at one unit. The EVENT node (EVE4) is used to free an input buffer once the corresponding machine becomes available. It is also used to request the parts storage to send a part to this machine whose input buffer is empty. At the machine table, the part waits for the tools from the tool crib if they are not in the tool magazine of the machine. This waiting time is modeled as an activity, USERF(4), which returns the value of the waiting time. The SLAM variables, XX(11) to XX(19), represent the status of the machines. A value of 1 indicates that the respective machine is busy, while a value of 0 indicates that the machine is idle. This facilitates automatic collection of statistics on the utilization of the machines. The processing times on the machines are represented as activities (ACT 1 to ACT 9). After a part has been processed on a particular machine, the FREE node (FREM) sets the machine free. The output buffers at the machines are denoted by the QUEUE nodes (QU10 to QU18). These queues have a capacity of one unit each and they block any entities from entering when they are full, i.e., when there is a part already in the output buffer waiting for the unloading AGV.

Segment 4 (Fig.11) - The AWAIT node (AW23) is a dummy node in which a part waits until the unloader, unloading AGV and the unloading crane are free. USERF(5) returns the value of the travel time from the appropriate machine to the unloading station. The travel time for the unloading crane is returned by USERF(6). The unloading time to dismantle the entire unit at the unloading station is modeled as an activity. As in the case of the loading time, the unloading time is also stochastic and the same procedure is adopted to generate random variables that follow the specified distribution. Once the dismantling is over, the FREE nodes, namely FPAL, FULO, and FRUG, are used to free the pallet, the unloader and the unloading AGV. The EVENT node (EVE5) is a user written EVENT routine that releases the necessary fixture type and indicates that the unloading track is clear. The part is then transported to the parts storage. USERF(7) provides the value of the travel time elapsed. The FREE node (FUCR) releases the unloading crane at the central parts storage. The check as to whether a part that returns to the parts storage has finished all its operations or not is handled by the EVENT routine (EVE6). This routine also collects the necessary data which are printed out at the end of the simulation.

Segment 5 (Fig.8) - This is an intermediate segment that requests the parts storage to select a part when one or more input buffers are empty. It triggers the EVENT routine (EVE1) when the GATE (CLO2) is opened and a dummy entity waiting in the AWAIT node (BAIN) is released.

Five different network programs, one for each rule, were written in SLAM. These programs are the same except for the PRIORITY statement. This is a SLAM control statement that is used to store the parts in the central parts storage according to the scheduling rule being followed. As a result of this, five different network programs, one for each rule, were written and compiled. The SLAM network program relating to the FOPR rule is listed in Appendix-D. During the execution of the program, depending on the rule (policy) to be followed, the user is requested to enter the name of the appropriate network program that has been compiled.

4.3 Model verification and validation

Verification is the process of determining whether the simulation model is performing as intended,

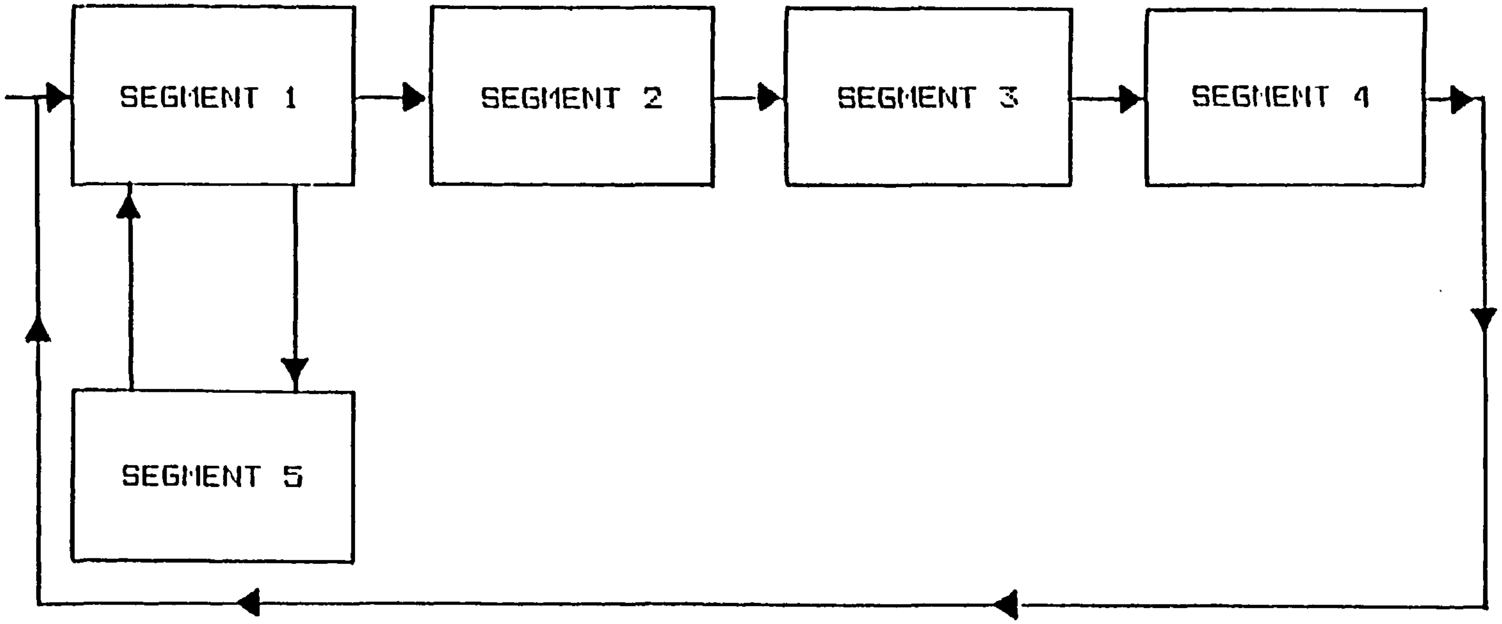


Fig.7 Block diagram of the SLAM network model

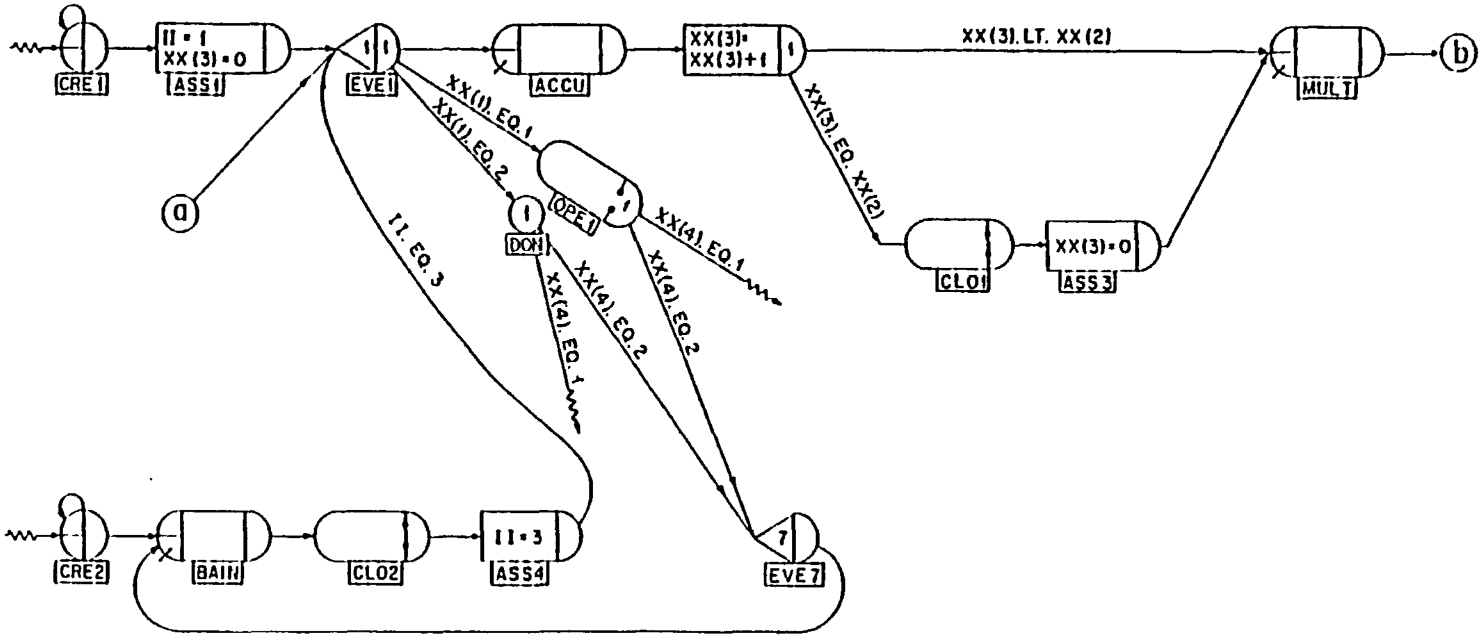


Fig.8 Segment 1 and segment 5 of the SLAM network model

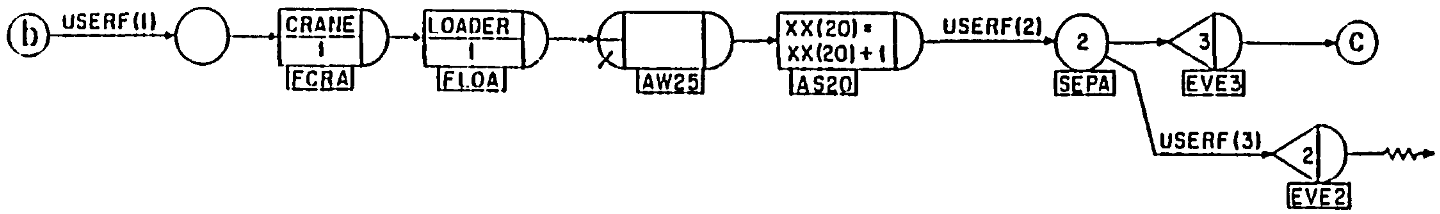


Fig.9 Segment 2 of the SLAM network model

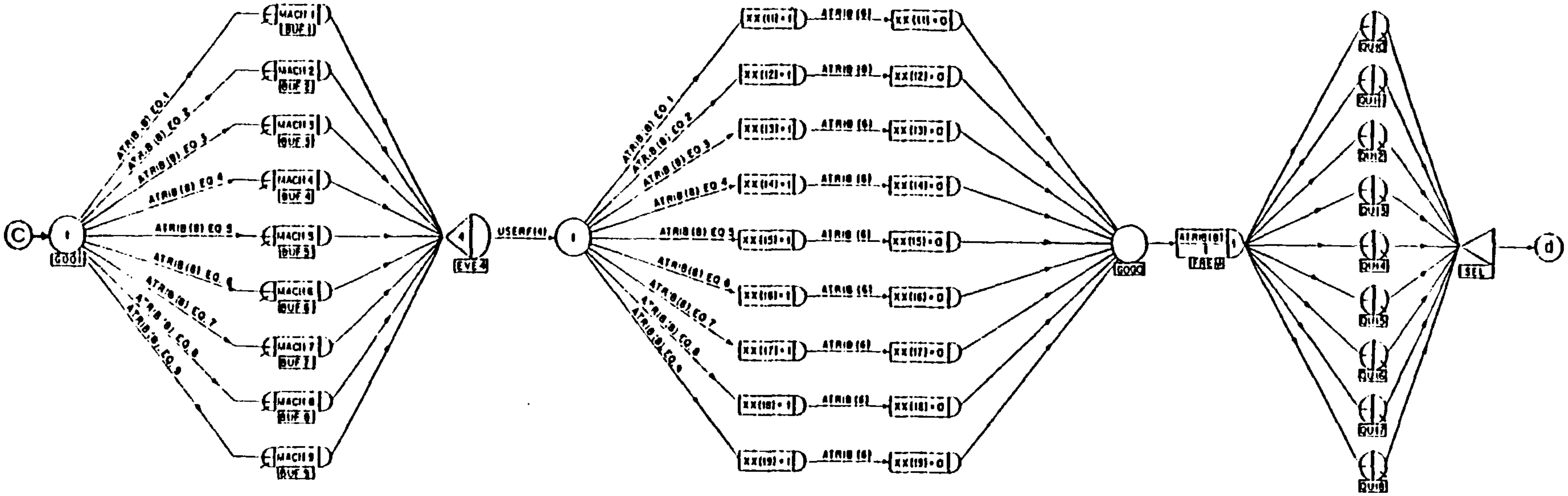


Fig.10 Segment 3 of the SLAM network model

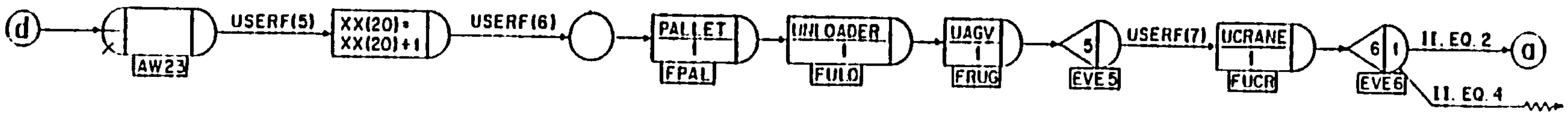


Fig.11 Segment 4 of the SLAM network model

i.e., debugging the computer program (Law & Kelton, 1982). The following techniques are generally used to debug computer programs of simulation models. These techniques include:

(i) the use of a modular approach to develop programs. However, this approach is dependent to a large extent on the language being used and many simulation languages have limitations in this regard.

(ii) the use of a trace in which the events that occur as the simulation progresses are printed out.

(iii) using results from analytical models that have already been developed for the system being simulated. In some cases by making certain simplifying assumptions, a suitable analytical model can be approximated. The results from the simulation model are then compared with those of the analytical model.

(iv) the use of animation that displays the simulation on the terminal as the simulation progresses. Presently a number of simulation softwares, for example, TESS, SIMAN, SEE-WHY, etc., that represent the system graphically on the terminal and display the changes that take place as the simulation progresses, are available.

The program that was written for simulating the

FMS contains thousands of lines of computer code. There exists no analytical model that is similar to the FMS under consideration. Verification of the system by equating it to an existing analytical model and comparing the results of both is out of consideration. Hence, the model was verified by tracing the program as the simulation progressed. At the user's request for a trace of the simulation, the WRITE statements are activated and the various events that take place at discrete points in time are printed out. Using one or two of machines and and a maximum of three part types, this procedure was followed for different number of combinations. From the printouts it was found that model was performing as intended. This was the main debugging tool used to verify the simulation model. A typical traced output that was used to verify the program is shown in Appendix-E.

Validation is the process of determining whether the simulation model is an accurate representation of the real world system. Complete validation for the FMS under consideration is extremely difficult due to the non-existence of such a system. Hence, the general validation procedure in which output from the simulation model is compared with the output from the real system for the same input details, is not possible with this system.

CHAPTER V

MODEL APPLICATION

In this chapter the application of the microcomputer-based simulation model is demonstrated through the use of three examples. Section 5.1 explains the statistical technique that is used to select a policy from the set of policies under consideration. The examples considered in Section 5.2 illustrate this selection technique. The factorial design set up that is used to aid the study of the effects of the important factors (resources) on system performance measure is explained in the same section.

5.1 Selection of the policy

The FMS under consideration processes different batches of part types during different periods. The system starts operating at time 0 with all the machines in an idle state and the raw materials for the entire set of part types to be machined during the period made available in the central parts storage. The system stops operating when all the parts in the entire batch have been processed, thereby completing a cycle. The next batch of parts to be processed are introduced into the system and the system starts all over again for the next cycle. Thus the system is a terminating type of system that starts with certain initial conditions at a time 0 and terminates at a time T_E on the occurence of event E. Hence, statistical techniques that are generally applied to analyse output data for terminating simulations can be applied to this system.

Given all the necessary details of the FMS under consideration, i.e., the configuration, the part types to be processed and their details, the levels of the various resources that are available, etc., our objective is to select a scheduling rule (policy) from the five rules under consideration. The statistical procedure that is used for this purpose is based on the method developed by Dudewicz and Dalal (1975) for selecting a policy out of, say, 'k' policies. The entire procedure is a comparison of alternate policies based on some criterion, namely, the system performance measure. The policy to be selected is the one that results in either a low or a high value of the system performance measure under consideration, as the case maybe. For example, if makespan, which is defined as the total time in system for the set of part types to be processed by the system, is chosen as the system performance measure, we would select the policy that results in the shortest makespan. However, if machine utilization, which is defined as the ratio of busy time of the machine to the total simulation time, is chosen as the system performance measure, we would select the policy that results in the highest machine utilization.

5.1.1 Notations used

X_{ij} - the random variable of interest (some measure of performance) from the j^{th} independent replication of the i^{th} policy, (i = 1,2,...,k) and (j = 1,2...,N)

k - number of policies under consideration

CS - the event of 'correct selection' of a policy

P{CS} - the probability of selecting the correct policy

μ_i - the population mean of the random variable X_{ij} for the policy 'i'.

$\mu_i = E(X_{ij})$

Depending on the measure used to evaluate the system performance, we would select the policy that results in either the smallest expected response or the largest expected response. Let us assume that we are interested in selecting the policy that results in the smallest expected response.

Let μ_{in} be the n^{th} smallest of the μ_i's, such that:

$$\mu_{i1} \leq \mu_{i2} \leq \text{......} \leq \mu_{ik}$$

Our objective now is to select the policy with the smallest expected response, μ_{i1}. Since the output from a simulation model is of a stochastic nature, there is an inherent randomness in the system performance measure X_{ij}. Thus, it is difficult to be absolutely sure that a correct selection (CS) from amongst the alternatives can be made. However, we can select the alternative whose population mean is less than the next best population mean by a prescribed value d^*, with a given probability P^*. The exact formulation then becomes,

$$P\{CS\} \geq P^*, \text{ such that } \mu_{i2} - \mu_{i1} \geq d^*$$

where P^* and d^* are specified by the analyst. The procedure involves a "two-stage" sampling. In the first stage we make a fixed number of replications of each policy, and then we use the resulting variances to determine the number of additional replications needed for each policy, to reach a decision. Since we are considering independent replications, it can be assumed that the X_{ij}'s are independent.

Let n_O - number of replications for each of the k policies in the first stage. The recommended value of n_O is at least 15 (Law & Kelton, 1982).

The first stage sample means and variances are given by Eqn (1) and Eqn (2) as

$$\bar{X}_i^{(1)}(n_O) = \frac{\sum_{j=1}^{n_O} X_{ij}}{n_O} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$s_i^2(n_O) = \frac{\sum_{j=1}^{n_O} [\, X_{ij} - \bar{X}_i^{(1)}(n_O) \,]^2}{[\, n_O - 1 \,]} \quad \ldots\ldots\ldots(2)$$

for i = 1,2,....,k.

Now, if N_i is the total sample size needed for policy 'i', we have:

$$N_i = \left\lceil \max\left(n_O + 1, \frac{h_1^2 \; s_i^2(n_O)}{(d^*)^2} \right) \right\rceil \ldots\ldots\ldots(3)$$

where $\lceil z \rceil$ is the smallest integer that is greater than or equal to the real number z, and h_1 (which depends on k, P^* and n_O) is a constant. The values of h_1 for different values of k, P^*, and n_O are provided in the tables listed by Law & Kelton (1982).

To obtain the second stage sample means we need $(N_i - n_O)$ more replications of system 'i' (i = 1,2,...

....,k). The second stage sample means is calculated using the formula in Eqn(4).

$$\bar{X}_i^{(1)}(N_i - n_0) = \frac{\sum_{j=n_0+1}^{N_i} X_{ij}}{[\,N_i - n_0\,]} \quad ....(4)$$

The number of runs, n_0 and $(N_i - n_0)$, for the two stages are different. As a result of this the first stage and the second stage sample means have to be weighted to obtain the overall mean. The weights W_{i1} and W_{i2}, assigned to the first stage and the second stage, are defined as:

$$W_{i1} = \frac{n_0}{N_i}\left(1 + \left\{1 - \frac{N_i}{n_0}\left[1 - \frac{(N_i - n_0)(d^*)^2}{h_1^2 s_i^2(n_0)}\right]\right\}^{1/2}\right) \quad ...(5)$$

and

$$W_{i2} = 1 - W_{i1}, \quad (i=1,2,.....,k) \quad ....(6)$$

The weighted sample means for policy 'i' is given by Eqn(7) as

$$\tilde{X}_i(N_i) = W_{i1}\bar{X}_i^{(1)}(n_0) + W_{i2}\bar{X}_i^{(2)}(N_i - n_0) \quad ....(7)$$

These calculations that are given in Eqn(1) through Eqn(7) are performed for all the 'k' policies. The weighted sample means $\tilde{X}_i(N_i)$ are then used to select the policy. Depending on the system performance measure under consideration, the policy that results in the smallest or largest value of $\tilde{X}_i(N_i)$ is selected. One important point to be noted here is the selection of P^* and d^* which ultimately depends on the analyst's goal and the particular system under investigation. P^* and d^* could be chosen after the first stage sampling.

This selection procedure is explained in Section 5.2.1, Section 5.2.2 and Section 5.2.3 with the help of examples. Since five scheduling rules (policies) are under consideration, we have k=5. Policy 'i' (i=1,2,3,4,5) is coded as follows:

i=1 -- Random selection rule (RANDOM)

i=2 -- Fewest operations remaining (FOPR)

i=3 -- Most operations remaining (MOPR)

i=4 -- Shortest processing time (SPT)

i=5 -- Longest processing time (LPT)

5.2 Problem data

The problem data considered in this section, to illustrate the application of the simulation model, represent the typical parts spectrum and machine variety found in flexible manufacturing systems. The parts considered are of the non-rotational (prismatic) type and the machines used in the system are general purpose horizontal and vertical machining centers.

5.2.1 Problem 1

The FMS assumed for this problem consists of five machining centers, viz., three vertical machining centers (Machine # 1, 2 and 5) and two horizontal machining centers (Machine # 3 and 4). All these machines are laid out in a single lane as shown in Fig.12. The sketches and process sheets of the set of parts to be processed by this system are given in Appendix-F. The list of all the fixture types and the cutting tools that are required for the different operations of these parts are provided in Appendix-G. Six part types form the parts spectrum for this problem. The summary of the part type data for these six parts are shown in Table 1. Table 2 gives the summary of the system layout data and Table 3 lists the quantities of the various resources available. The travel times and the speed details of the different material handling systems are shown in Table 4.

5.2.1.1 Selection of the scheduling rule

These input data were organised into different data files using Program 1. The simulation of the FMS was carried out for each of the five scheduling rules. Following the observations made by Law & Kelton (1982) on the value of the first stage sample size, a value of 20

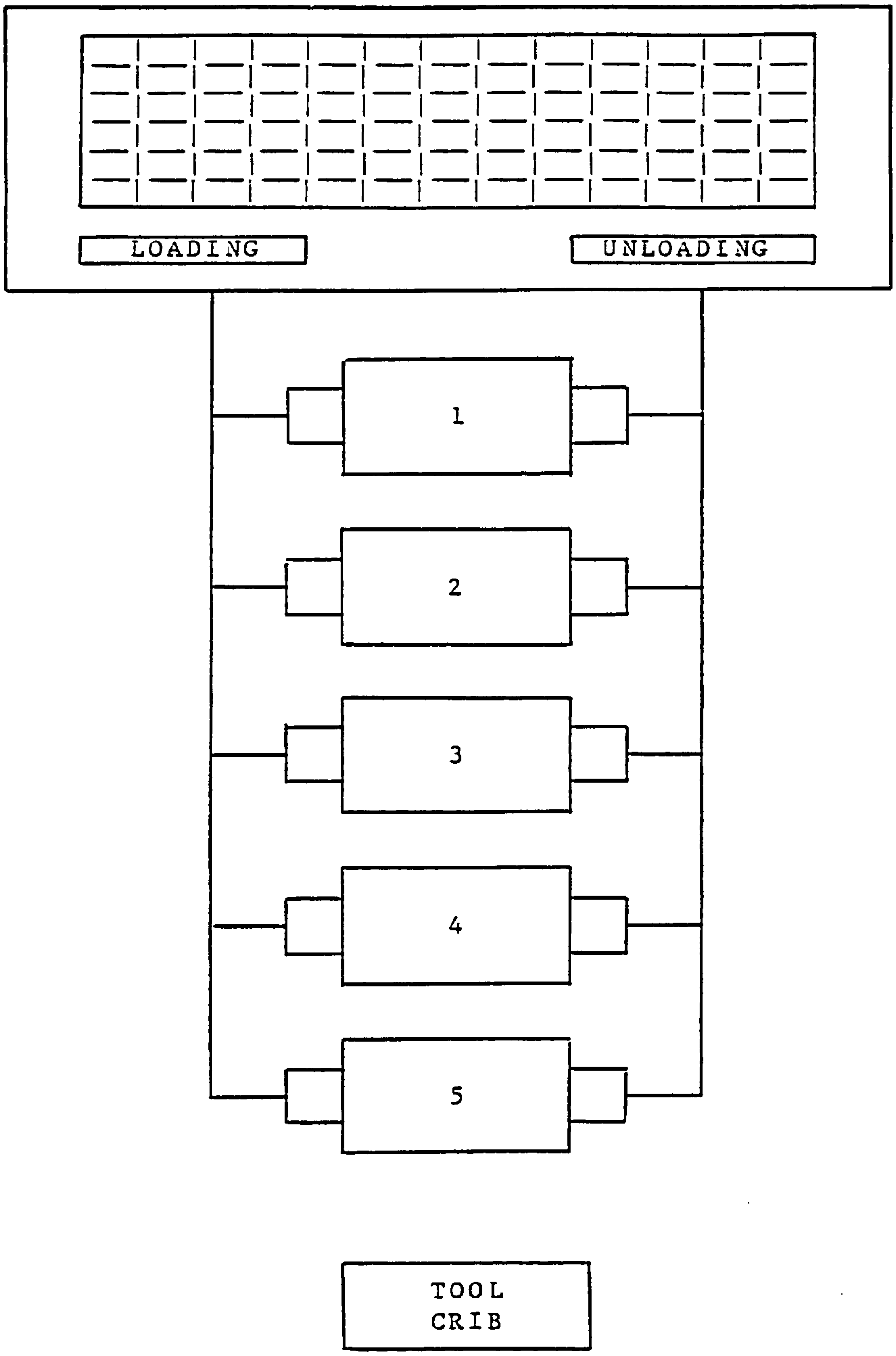


Fig.12 Layout of the FMS - Problem 1

Table 1

Summary of part type data - Problem 1

Part Type	Batch Size	Prty	Opn No.	Proc. Time (mins)	FT	Machine(s)	Tools required
1	5	NO	1	24.00	1	1 or 2 or 5	13,14,8,24
			2	24.00	1	1 or 2 or 5	13,14,8,24
			3	18.00	1	3 or 4	15,16,8,24
2	9	NO	1	31.00	2	1 or 2 or 5	17,18,9,25,7
3	7	NO	1	6.00	3	1 or 2 or 5	19,20
			2	23.00	3	3 or 4	19,20,17,18,21,22
4	12	NO	1	9.00	4	1 or 2 or 5	8,10,26
			2	17.00	4	3 or 4	15,16,5,6,3,4,11,27,29
5	10	NO	1	36.00	3	1 or 2 or 5	17,18,12,28,30
6	7	NO	1	8.00	5	1 or 2 or 5	19,20
			2	14.00	5	1 or 2 or 5	19,20,8,10,26
			3	21.00	6	4	21,23,1,2,5,6

Note : Prty - Priority, Opn. No. - Operation Number
Proc. Time - Processing Time, FT - Fixture type

Table 2

Summary of system layout data - Problem 1

MACHINE	LANE	DISTANCE FROM MACHINE TO		
		LOADING STATION	UNLOADING STATION	TOOL CRIB
1 - VMC	1	10.00	10.00	50.00
2 - VMC	1	20.00	20.00	40.00
3 - HMC	1	30.00	30.00	30.00
4 - HMC	1	40.00	40.00	20.00
5 - VMC	1	50.00	50.00	10.00

Note : All the distances are in meters

VMC - Vertical Machining Center

HMC - Horizontal Machining Center

Table 3

Summary of resource data - Problem 1

Resource name	Quantity
Loading crane	1
Unloading crane	1
Loaders	1
Unloaders	1
Loading AGVs	1
Unloading AGVs	1
Pallets	5
Fixture type 1	2
Fixture type 2	1
Fixture type 3	2
Fixture type 4	1
Fixture type 5	1
Fixture type 6	1

Table 4

Summary of speed and time details of the material handling systems - Problem 1

Speed of the loading AGV	= 20.00 mts/min
Speed of the unloading AGV	= 20.00 mts/min
Speed of the tool conveyor	= 10.00 mts/min
Travel time for the loading crane	= 2.00 mins
Travel time for the unloading crane	= 2.00 mins
Loading time	- Uniformly distributed between 2.00 mins and 3.00 mins
Unloading time	- Uniformly distributed between 1.00 and 2.00 mins

was chosen for n_O. Twenty independent runs for each scheduling rule were made using different random number streams for each run. To ensure that each of these policies was tested under identical conditions, the random number streams for each policy were made identical. The makespan was chosen as the measure of performance of the system.

The first stage sampling for the five policies was done and the results are shown in Table 5. The following values of P^* and d^* were chosen for the analysis. $P^* = .90$, and $d^* = 15$ minutes

These values for P^* and d^* indicate that we want to be 90% confident that the difference in the makespan values for the selected policy and the next policy would be greater than or equal to 15 minutes. From the tables, for a value of k=5 and the above mentioned values of P^* and n_O, h_1 was found to be 2.747. The number of additional runs required $(N_i - n_O)$ for the i^{th} policy (i=1,2,..,5) was calculated based on the results of the first-stage sampling and was found to be one. The additional run was conducted for each policy and the second-stage sampling procedure was applied. The weights W_{i1} and W_{i2} for each policy were calculated. From these weights, the weighted sample means for the system performance measure under consideration were determined. Table 5 shows the weighted sample means of

Table 5

Weighted sample means of makespan - Problem 1

i	$\bar{X}_i^{(1)}(20)$	$s_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i - 20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$ (mins)	90% confidence interval (mins)
1	1145.676	620.110	21	1159.036	.9734	.0266	1146.031	1145.676 $\pm$ 43.08
2	1128.065	152.677	21	1127.931	1.3274	-.3274	1128.109	1128.065 $\pm$ 21.38
3	1177.570	288.320	21	1201.931	1.1829	-.1829	1173.152	1177.570 $\pm$ 29.38
4	1158.917	443.128	21	1155.390	1.0892	-.0892	1159.232	1158.917 $\pm$ 36.42
5	1173.930	462.621	21	1173.800	1.0790	-.0790	1173.940	1173.930 $\pm$ 37.20

|—First stage sampling—|—Second stage sampling—|

[`i´ refers to policy]

i = 1 - Random selection rule
i = 2 - Fewest operations remaining
i = 3 - Most operations remaining
i = 4 - Shortest processing time
i = 5 - Longest processing time

$P^* = .90$, $d^* = 15$ mins, $k=5$ and $h_1 = 2.747$

the makespan for each rule. From the column in Table 5 corresponding to the weighted sample means of the makespan $[\tilde{X}_i(N_i)]$ for each scheduling policy, it can be seen that the Fewest Operations Remaining (FOPR) rule results in the shortest expected makespan. Hence, the FOPR rule is selected. The last column in Table 5 gives the 90% confidence interval of the makespan for the five policies. The formula to obtain the 90% confidence interval is given in Eqn (8) as

$$\bar{X} \pm (s/\ n\ ).(t_{n-1,\ 1-\alpha/2}) \quad ....(8)$$

where, n - number of simulation runs

X - mean of the n runs

s - standard deviation of X in the n runs

α - level of significance.

The 90% confidence interval for policy 2, the FOPR rule, is 1128.065 ± 21.38 minutes, i.e., we can be 90% confident the makespan will be between 1106.685 minutes and 1149.445 minutes, when we are using this policy.

5.2.1.2 Experimental design setup

As proposed earlier, after the selecting the scheduling policy, the next stage is to study the effects of the important resources on the makespan, using this policy. This is done by setting up a suitable experimental design. A 2^k factorial design was chosen for this purpose. The 'k' factors, or important resources, are set at two levels - a high level and a low level. At the outset, though it might seem that each resource can have more than two levels, in reality, the hardware constraints, the operational constraints, the technical constraints, etc. do not permit certain resources to have more than two levels. For example, due to operational constraints, it is not feasible to have more than one loading or unloading crane to move the parts between the central parts storage and the loading and unloading stations.

The selection of the important resources is based on their respective utilizations. Table 6 gives the list of the various resources that were used by the system, their quantities, and their respective average utilizations for twenty one runs using the FOPR rule. Resources used in a FMS are classified as shared resources, those that are used by all the part types and specific resources, those that are used by only a certain part type. The fixture types belong to the latter category while the other resources used by this system belong to the former category.

Consider the shared resources that are at their minimum level, i. e., having a quantity of one. We can see that the loading AGV was utilized on the average to a larger extent than the rest, say, loading crane or

Table 6

Resource quantities and utilization - FOPR rule

Resource	Quantity	Average Utilization
Loading crane	1	53.3%
Unloading crane	1	66.3%
Loader	1	53.3%
Unloader	1	49.8%
Loading AGV	1	74.3%
Unloading AGV	1	49.8%
Pallet	5	69.8%
Fixture type 1	2	30.6%
Fixture type 2	1	42.4%
Fixture type 3	2	50.8%
Fixture type 4	1	72.5%
Fixture type 5	1	44.2%
Fixture type 6	1	27.0%

loader. It can also be seen that there are five pallets that were on the average utilized at approximately 70%, i.e., only 3.5 pallets out of the five pallets were utilized on the average. Considering the specific resources, namely, the different fixture types, we can see that there are two units of fixture type 1 which were utilized about 30% on the average. This again is a low value considering the fact that there are two fixtures of the same type. These three resources - loading AGV, pallet and fixture type 1 - were considered as important resources for this problem. To study the effects of these resources a 2^3 factorial design was set up with each factor at two levels. Table 7 gives the different treatment combinations of these three factors. The design matrix that facilitates calculations of the factor effects and interactions is shown in Table 8. Ten simulation runs were made for the eight treatment combinations of these three factors (resources). The makespan (response) is also given in Table 8 for each treatment.

With the help of this design matrix, the main effects and the interactions were calculated and listed as shown in Table 9. The main effect of a factor j is the average change in response due to moving j from its '-' level to its '+' level while holding all the other factors fixed and is calculated using the notations in the design matrix. For example, if R_1, R_2, ... ,R_8 are the

Table 7

Treatment combinations for the three resources

TREATMENT #	RESOURCE LEVELS		
	LOADING AGV	PALLET	FIXTURE TYPE 1
1	1 (-)	3 (-)	1 (-)
2	2 (+)	3 (-)	1 (-)
3	1 (-)	5 (+)	1 (-)
4	2 (+)	5 (+)	1 (-)
5	1 (-)	3 (-)	2 (+)
6	2 (+)	3 (-)	2 (+)
7	1 (-)	5 (+)	2 (+)
8	2 (+)	5 (+)	2 (+)

Note : The signs within parantheses indicate a high or low level of each resource

(+) - High level

(-) - Low level

Table 8

2^3 factorial design matrix - FOPR rule

TREATMENT	A	B	C	AB	AC	BC	ABC	Makespan (mins) 90% confidence interval
1	−	−	−	+	+	+	−	1401.599 ± 9.36
a	+	−	−	−	−	+	+	1382.950 ± 15.87
b	−	+	−	−	+	−	+	1115.009 ± 33.50
ab	+	+	−	+	−	−	−	1087.655 ± 28.78
c	−	−	+	+	−	−	+	1421.984 ± 8.98
ac	+	−	+	−	+	−	−	1390.228 ± 16.92
bc	−	+	+	−	−	+	−	1127.930 ± 26.09
abc	+	+	+	+	+	+	+	1089.061 ± 16.66

Factor A - Loading AGV
Factor B - Pallet
Factor C - Fixture type 1

Levels of Factor A - 1 and 2
Levels of Factor B - 3 and 5
Levels of Factor C - 1 and 2

Table 9

Effects of factors on system response - FOPR rule

Effect	Makespan
Main effects	
Loading AGV A	-29.1500
Pallet B	-294.2765
Fixture type 1 C	10.4975
Two factor interactions	
A X B	-3.9545
A X C	-6.1555
B X C	-3.3340
Three factor interactions	
A X B X C	.3993

responses for each of the eight treatments, respectively, the main effect of factor A (e_A) is given by the formula, (Law & Kelton, 1982)

$$e_A = \frac{(R_2-R_1) + (R_4-R_3) + (R_6-R_5) + (R_8-R_7)}{4} \quad .....(9)$$

Similarly, the interaction effect for the factors A and B is given by Eqn (10),

$$e_{AB} = \frac{R_1 - R_2 - R_3 + R_4 + R_5 - R_6 - R_7 + R_8}{4} \quad ..(10)$$

5.2.1.3 Discussion

(i) From Table 9 it is clear that the two factor interactions are small compared to the main effects of factors A and B. The three factor interaction is also negligible. However, the main effect due to factor B, i.e., the pallet, is quite significant in decreasing the response (makespan). It can be seen that an increase in the number of pallets from a low level of three to a high level of five resulted in a considerable decrease in the makespan. The number of parts circulating inside the system is the same as the number of pallets used by the system because every fixtured part type is mounted on a pallet irrespective of the fixture type or part type. Hence, increasing the number of pallets in the system increases the number of parts circulating in the system thereby decreasing the makespan. The FMS consists of five machines and with only three pallets being used, at least two machines are idle at any point in time during the operation of the system. Due to the non-availability of the pallets, parts that are selected to enter the system for processing are forced to wait, thereby increasing the total time the parts spend in the system.

(ii) The main effect of factor A, i.e., the loading AGV, is not as much as that of factor B, the pallet (Table 9). Increasing the number of loading AGV's resulted in a decrease in the makespan. However, the decrease in the makespan due to the increase in the number of pallets is ten times larger than that due to the increase in the number of loading AGV's. This can be attributed to the traffic congestions on the loading track. Though an increase in the number of loading AGVs increases the number of parts circulating inside the system, a loading AGV spends some time waiting at the loading station whenever there is another AGV travelling on the same track. As a result of this, increasing the number of loading AGV's did not result in a significant reduction in the makespan.

(iii) Increasing factor C, i.e., the fixture type 1, resulted in increasing the makespan (Table 9). The fixture, being a more specific resource, is used only by part type 1 and probably during the entire operation of the system, there was no part type 1 that was waiting for fixture type 1. However, it should be mentioned here that the effect of fixture type 1 depends also on the batch size of part type 1. If the batch size of this part type is very high compared to the batch sizes of the rest of the parts spectrum, the number of parts waiting for this fixture type would be high. In such conditions, increasing the quantity of this fixture would tend to decrease the makespan. In this example, since the batch size of part type 1 was small compared to the other part types, increasing the number of units of fixture type 1 did not result in a decrease in the makespan.

(iv) Table 10 lists the average machine utilizations for the different treatment combinations. Considering all the treatments, it can be seen that Machine 1 was utilized to a larger extent compared to the other machines. This is due to the proximity of this machine to the loading and unloading stations that always results in Machine 1 being selected whenever it is available. Higher levels of the pallets (Treatment 3, 4, 7, 8 - Table 10) resulted in higher utilizations for all the machines. This is because of the significant decrease in the makespan whenever there is an increase in the number of pallets, as seen earlier. Data of this type on machine utilizations are useful in deciding whether a particular machine should be kept running or shut down.

Table 10

Average machine utilizations - Problem 1

TRTMNT.	MACHINE				
	1	2	3	4	5
1	33.4	27.7	25.0	18.0	23.3
2	33.0	28.3	26.5	17.0	24.2
3	49.2	37.9	33.3	20.7	19.3
4	51.3	37.9	35.5	19.9	19.6
5	31.3	28.2	24.7	17.6	23.7
6	32.0	29.4	25.7	17.6	23.7
7	43.0	35.0	32.6	20.8	26.8
8	46.9	38.1	32.8	22.5	23.7

From Table 10, we can see that Machine 4 has been utilized to the lowest extent compared to the other machines. Under such conditions a decision could be made whether this machine should be used to process this particular batch of part types. The simulation model could be used to evaluate the performance of the system without Machine 4.

(v) The average utilizations of the resources for the different treatment combinations are shown in Table 11. The loading crane, loader and loading AGV are resources that are seized simultaneously. After the unit has been mounted on the loading AGV, the loading crane and the loader are released while the loading AGV is not released until it has transported the unit to the respective machine. As a result of this, it can be found from Table 11 that the loading AGV is utilized to an higher extent than the loading crane and the loader.

The unloading crane, unloader and unloading AGV are seized simultaneously. However, after dismantling the unit at the unloading station, the unloader and the unloading AGV are released while the unloading crane is not released until it has transported the part to the central parts storage. Hence, the unloading crane is utilized to an higher extent than the unloading AGV and the unloader.

Considering the pallets and the loading AGV, it

Table 11

Average resource utilizations - Problem 1

	RESOURCES						
TRT.	Crane	Ucrane	Load	Uload	AGV-L	AGV-UL	Pallet
1	43.0	54.2	43.0	40.1	60.7	40.1	85.3
2	43.7	54.9	43.5	41.5	30.8	41.5	86.9
3	54.0	66.2	54.0	49.5	74.4	49.5	72.8
4	55.4	67.7	55.4	50.6	38.1	50.6	76.1
5	42.3	53.3	42.3	40.3	59.7	40.3	84.0
6	43.3	54.4	43.3	41.1	30.5	41.1	85.8
7	53.3	66.3	53.3	49.8	74.3	49.8	69.8
8	55.2	68.8	55.2	51.7	38.5	51.7	73.9

Note :	Crane	--	Loading crane
	Ucrane	--	Unloading crane
	Load	--	Loader
	ULoad	--	Unloader
	AGV-L	--	Loading AGV
	AGV-UL	--	Unloading AGV

can be seen that increasing their levels significantly decreases the makespan (Table 8) and their respective utilizations. A similar observation has been reported by Chang et al. (1986).

(vi) The average utilizations of the specific resources, namely, the fixture types, are shown in Table 12. It can be seen that the high utilizations for all the fixture types, except fixture type 1, occurs for the four treatments (Treatments 3, 4, 7 and 8 - Table 12). However, for the fixture type 1, it can be seen that high utilisations occur only for treatments 3 and 4. Increasing this fixture type from one to two units resulted in a decrease in utilization per unit (Treatments 5, 6, 7 and 8 - Table 12).

The above discussion based on the results of the experimental design can be used as a basis to decide the quantities of the resources to be used to process a particular batch of part types. Due considerations should be given to the cost factors such as, the cost of a pallet, or a loading AGV, or fixture, etc., before a final decision can be made.

5.2.2 Problem 2

The parts spectrum and the machine variety considered for this problem are the same as that of

Table 12

Average fixture utilizations - Problem 1

	FIXTURE TYPES					
TRT	1	2	3	4	5	6
1	43.5	32.2	36.2	56.0	31.1	20.8
2	44.2	32.9	37.0	57.5	31.5	20.8
3	61.3	44.2	52.6	78.1	47.6	27.9
4	66.4	44.4	54.8	82.7	49.0	28.5
5	21.5	30.9	36.3	54.2	30.6	20.5
6	22.1	31.6	37.5	55.1	30.9	20.6
7	30.6	42.3	50.8	72.5	44.2	27.0
8	32.6	45.2	53.8	77.2	46.5	28.1

Problem 1. However, one of the part types has a priority over the other part types. This type of situation is common in practice. Part types ordered by important customers, part types to be processed urgently, etc. are always given a priority over the other part types in the batch. The simulation model is designed such that it can be adapted to such situations.

The summary of the part data for Problem 2 is the same as that for Problem 1 (Table 1) except for part type 5 that has been assigned a priority. The system layout data, resource data, and the speed and time details of the different material handling systems for this problem are also the same as that for Problem 1 (Tables 2, 3 and 4).

Table 13 shows the results of the first stage and second stage sampling procedures applied to select the scheduling policy. Based on the weighted sample means of the makespan, the RANDOM rule is selected. The only difference in the input data for Problems 1 and 2 is the inclusion of a part type (part type 5) a higher priority. However, the selected scheduling policy for both these problems are different.

5.2.3 Problem 3

Problem 3 demonstrates the application of the simulation model to multiple-lane layouts. The layout of

Table 13

Weighted sample means of makespan - Problem 2

i	$\bar{X}_i^{(1)}(20)$	$s_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i - 20)$	W_{i1}	W_{i2}	$\tilde{X}_i(N_i)$ (mins)	90% confidence interval (mins)
1	1115.393	654.923	21	1137.398	1.1305	-.1305	1112.527	1115.393 ± 44.27
2	1123.847	606.492	21	1117.983	1.1470	-.1470	1124.708	1123.847 ± 42.60
3	1142.270	317.766	21	1156.639	1.2893	-.2893	1138.113	1142.270 ± 30.84
4	1145.226	230.330	21	1119.380	1.3693	-.3693	1154.771	1145.226 ± 26.26
5	1128.840	621.140	21	1120.845	1.1419	-.1419	1129.974	1128.840 ± 43.11

|——First stage sampling——|——Second stage sampling——|

[`i´ refers to policy]

i = 1 - Random selection rule
i = 2 - Fewest operations remaining
i = 3 - Most operations remaining
i = 4 - Shortest processing time
i = 5 - Longest processing time

$P^* = .90$, $d^* = 20$ mins, $k=5$ and $h_1 = 2.747$

the FMS for this problem consists of two lanes, with three machines on lane 1 and two machines on lane 2 (Fig.13). The first four part types form the parts spectrum for this problem. The summary of the part type data for these four parts are shown in Table 14. Table 15 gives the summary of the system layout data and Table 16 lists the quantities of the various resources available. The travel times and the speed details of the different material handling systems are shown in Table 17.

Table 18 shows the results of the first stage and second stage sampling procedures applied to select the scheduling policy. Based on the weighted sample means of the makespan, the Longest Processing Time (LPT) rule is selected. From the analysis of these three problems it can be seen that variations in the input data result in the selection of different scheduling. This is similar to the observation made by Stecke and Solberg (1981).

The use of the simulation model to address the operating problems of a FMS, namely, the scheduling policy to be followed, has been explained through the use of examples in Section 5.2.1, Section 5.2.2 and Section 5.2.3. A detailed analysis was conducted for Problem 1. The 2^k factorial design that was proposed to study the effects of the important resources on the system performance was applied to Problem 1. The value of k was equal to 3. However, for higher values of k the number of

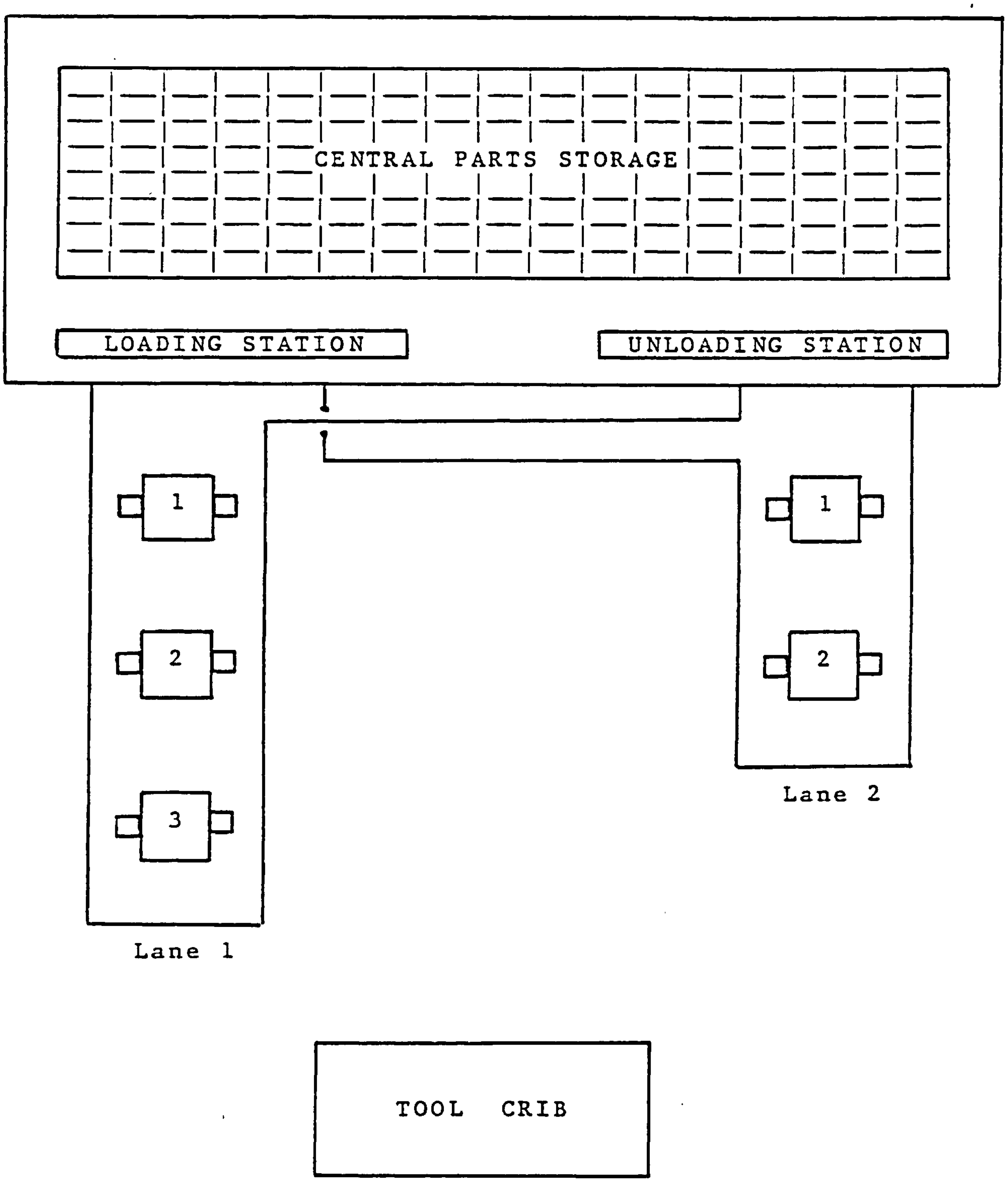


Fig.13 Layout of the FMS - Problem 3

Table 14

Summary of part type data - Problem 3

Part Type	Batch Size	Prty	Opn No.	Proc. Time (mins)	FT	Machine(s)	Tools required
1	2	NO	1	24.00	1	1 or 2 or 5	13,14,8,24
			2	24.00	1	1 or 2 or 5	13,14,8,24
			3	18.00	1	3 or 4	15,16,8,24
2	9	NO	1	31.00	2	1 or 2 or 5	17,18,9,25,7
3	3	NO	1	6.00	3	1 or 2 or 5	19,20
			2	23.00	3	3 or 4	19,20,17,18,21,22
4	41	NO	1	9.00	4	1 or 2 or 5	8,10,26
			2	17.00	4	3 or 4	15,16,5,6,3,4,11, 27,29

Note : Prty - Priority, Opn. No. - Operation Number
Proc. Time - Processing Time, FT - Fixture type

Table 15

Summary of system layout data - Problem 3

MACHINE	LANE	DISTANCE FROM MACHINE TO		
		LOADING STATION	UNLOADING STATION	TOOL CRIB
1 - VMC	1	10.00	16.00	30.00
2 - VMC	1	20.00	26.00	20.00
3 - HMC	1	30.00	36.00	10.00
4 - HMC	2	15.00	10.00	30.00
5 - VMC	2	25.00	20.00	20.00

Note : All the distances are in meters

VMC - Vertical Machining Center

HMC - Horizontal Machining Center

Table 16

Summary of resource data - Problem 3

Resource name	Quantity
Loading crane	1
Unloading crane	1
Loaders	1
Unloaders	1
Loading AGVs	1
Unloading AGVs	1
Pallets	5
Fixture type 1	1
Fixture type 2	1
Fixture type 3	1
Fixture type 4	1

Table 17

Summary of speed and time details of the material handling systems - Problem 3

Speed of the loading AGV	= 20.00 mts/min
Speed of the unloading AGV	= 20.00 mts/min
Speed of the tool conveyor	= 10.00 mts/min
Travel time for the loading crane	= 2.00 mins
Travel time for the unloading crane	= 2.00 mins
Loading time	- Uniformly distributed between 2.00 mins and 3.00 mins
Unloading time	- Uniformly distributed between 1.00 and 2.00 mins

Table 18

Weighted sample means of makespan - Problem 3

i	$\bar{X}_i^{(1)}(20)$	$s_i^2(20)$	N_i	$\bar{X}_i^{(2)}(N_i - 20)$	W_{i1}	W_{i2}	$X_i(N_i)$ (mins)	90% confidence interval (mins)
1	2577.272	958.842	21	2565.385	1.0738	-.0738	2577.721	2577.272 ± 53.57
2	2551.960	194.847	21	2551.807	1.4147	-.4147	2552.023	2551.965 ± 24.15
3	2634.196	252.121	21	2636.078	1.3459	-.3459	2633.545	2634.196 ± 27.47
4	2639.441	155.563	21	2622.835	1.4087	-.4087	2647.423	2639.441 ± 21.57
5	2543.338	203.484	21	2538.988	1.4027	-.4027	2545.091	2543.338 ± 24.68

|— First stage sampling ——|—— Second stage sampling ——|

[\`i´ refers to policy]

i = 1 - Random selection rule
i = 2 - Fewest operations remaining
i = 3 - Most operations remaining
i = 4 - Shortest processing time
i = 5 - Longest processing time

$P^* = .90$, $d^* = 20$ mins, $k=5$ and $h_1 = 2.747$

treatment combinations is large. In such situations a 2^k factorial design with replicates would involve a large amount of computer time. To avoid this, a fractional factorial design could be employed in which only 'p' factors out of the k important factors are analysed (Law & Kelton, 1982). The choice of p is subjective and depends on the analyst.

Except for the loading and unloading times which are stochastic, the remaining times, like the processing time, travel times, etc., are deterministic. However, if these times are made stochastic the amount of randomness that would be introduced into the output would be high. Hence, a larger number of runs would be necessary to reduce the variance of the output measures. Variance reduction techniques could be applied to control the variance under such conditions.

Chapter VI

SUMMARY

A microcomputer-based simulation model of an idealised FMS has been developed. Some features such as alternate routings, treatment of fixtures as resources and part types with priorities were included to make the model realistic. The model was developed on a microcomputer using SLAM and was made user-interactive. Five different scheduling rules were selected to be investigated by the model for each set of part types to be processed by the system. The simulation model has been designed such that it is adaptable to variations in the layout as shown in Fig.2.

The statistical approach to select the scheduling policy and the experimental design setup used to aid the study of the effects of the important resources on system performance were explained through the use of examples. The simulation model is general enough to be applied to small- or medium-sized problems. The primary function of the model is to act as a decision tool in selecting the scheduling policy for the part types that are processed by the system under consideration.

Though the model attempted to include some realistic aspects mentioned earlier, breakdown of machines or resources which is a common feature in most manufacturing systems has not been considered. Also, complicated FMS layouts could not be incorporated into a general program. Complicated scheduling rules were not considered by this model. It is in these directions that this model could be used for further work and enhanced to include additional features.

REFERENCES

Acree, E. S.,(1983), Part and Tool Scheduling Rules for a Flexible Manufacturing System, Ph. D. Thesis, Dept. of Ind. Engg., Texas Tech University.

Baskett, F., Chandy, K. M., Muntz, R. R., Palacios, F. G., (1975), Open,Closed and Mixed Network of Queues with Different Classes of Customers, JACM, Vol 22, No. 2, 248-260.

Bilalis, N. G., Mamalis, A. G.,(1985), The Flexible Manufacturing Systems in Metal Removal Processing : An Overview, American Society for Metals, Vol 3, No. 4, January, 400-409.

Browne, J., Rathmill, K.,(1983), The Use of Simulation Modeling as a Design Tool for FMS, Proceedings of the II International Conference on Flexible Manufacturing Systems, London, U. K., 197-214.

Buzacott, J. A., Shantikumar, J. G.,(1980), Models for Understanding Flexible Manufacturing Systems, AIIE Trans., Vol 12, No. 4, 339-350.

Buzen, J. P.,(1973), Computational Algorithms for Closed Queueing Networks with Experimental Servers, Communications of ACM, Vol 16, No. 9, 527-631.

Carrie, A. S., Adhami, E., Stephens, A., Murdoch, I. C., (1984), Introducing a Flexible Manufacturing System, International Journal of Production Research, Vol 22, No. 6, 907-916.

Chang, Y., Sullivan, R. E., Wilson, J. R.,(1986), Using SLAM to design the material handling system of a flexible manufacturing system, International Journal of Production Research, Vol 24, No. 1, 15-26.

Cook, N. H.,(1975), Computer-Managed parts manufacture, Scientific American, Vol 232, No. 2, February, 22-28.

Drake, A.,(1984) Fixture design: Working with modules Manufacturing Engineering, January, 35-38.

Dudewicz, E. J., Dalal, S. R.,(1975), Allocation of Measurements in Ranking and Selection with Unequal Variances, Sankhya, B37, 28-78.

El Maraghy, H.,(1982), Simulation and Graphical Animation of Advanced Manufacturing Systems, Journal of Manufacturing Systems, Vol 1, No. 1, 53-63.

Groover, M. P., Zimmers, E. W.,(1984), CAD/CAM: Computer Aided Design and Manufacturing, Prentice Hall, Inc. U. S. A.

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

Jackson, J. R.,(1957), Networks of Waiting Lines, Operations Research, Vol 5, 518-521.

Kimemia, J., Gershwin, S. B.,(1985), Flow Optimization in Flexible Manufacturing Systems, International Journal of Production Research, Vol 23, No. 1, 81-96.

Kusiak, A.,(1983), Loading Models in Flexible Manufacturing Systems, Proceedings of VIII International Conference on Production Research, Windsor, Canada.

Law, A. M.,(1986), Introduction to Simulation: A Powerful Tool for Analysing Complex Manufacturing Systems Industrial Engineering, Vol 18, No.5, May, 46-86.

Law, A. M., Kelton, W. D.,(1982), Simulation Modeling and Analysis, McGraw-Hill, Inc., U. S. A.

Pritsker, A. A. B.,(1977), Modeling and Analysis using Q-GERT networks, Halstead Press and Pritsker Assosciates, Inc., U. S. A.

Pritsker, A. A. B.,(1984), Introduction to Simulation and SLAM II, Systems Publishing Corporation, West Lafayette, Indiana, U. S. A.

Ranky, P. G.,(1986), Computer Integrated Manufacturing - An Introduction with Case Studies, Prentice Hall Ltd., U. K.

Rathmill, K., Greenwood, N., Houshmand, H.,(1983), Computer Simulation of FMS, Proceedings of the II International Conference on Flexible

Manufacturing Systems, London, U. K., 251-280.

Schriber, T.,(1974), Simulation using GPSS, John Wiley, U. S. A.

Shanker, K., Tzen, Y. J.,(1985), A Loading and Dispatching Problem in a Random Flexible Manufacturing System, International Journal of Production Research, Vol 23, No. 3,579-599.

Solberg, J. J.,(1977), A Mathematical Model of Computerized Manufacturing Systems, Proceedings of the IV International Conference on Production Research, Tokyo, Japan, 1265-1275.

Stecke, K. E.,(1981), Production Planning Problems of Flexible Manufacturing Systems, Ph. D. Thesis, Purdue University.

Stecke, K. E., Solberg, J. J.,(1981), Loading and Control Policies for a Flexible Manufacturing System, International Journal of Production Research, Vol 19, No. 5, 481-490.

Stecke, K. E.,(1984), Design, Planning, Scheduling and Control Problems of Flexible Manufacturing Systems, Proc. of ORSA/TIMS Special International Conference on FMS: Operations Research Models and Applications, August, Ann Arbor, Michigan.

Wilhelm, W. E., Sarin, S. C.,(1983), Models for the Design of Flexible Manufacturing Systems, Proceedings of AIEE Spring Annual Conference, Louisville, Kentucky, 564-573.

Wortman, D. B., Miner, R. J.,(1984), Designing Flexible Manufacturing Systems Using Simulation, Autofact - 6, 24-29 to 24-55.

APPENDIX-A

COMPUTER PROGRAM LISTING - PROGRAM 1

```
10 REM **********************************************************************
20 REM *                                                                    *
30 REM * PROGRAM TO INPUT THE PART DETAILS OF ALL THE PARTS THAT ARE USED   *
40 REM *    IN THE SIMULATION MODEL.  THESE DETAILS WILL BE STORED IN       *
50 REM *        SEPARATE DATA FILES AND WILL BE ACCESSED BY THE             *
60 REM *              S L A M   M A I N  P R O G R A M                      *
70 REM *                                                                    *
80 REM **********************************************************************
90 REM
100 REM *** PROGRAM WRITTEN BY -----> THIRUVENGADAM RAVI ***
110 REM
120 OPTION BASE 1
130 DIM IGD(3),IFT(10),PRT(10),IALM(90),ITOOL(100),JPRPT(25),NF(30),PARR(25,220),DULOM(9),DLOM(9)
,DSTM(9),ITRACK(9)
140 REM ------------------------------
150 REM CALLING THE VARIOUS SUBROUTINES
160 REM ------------------------------
170 GOSUB 200
180 GOTO 350
190 CLS:COLOR 7,0,0:END
200 REM ****************************************************
210 REM *  SUBROUTINE TO DISPLAY FEATURES OF THE SEGMENT  *
220 REM ****************************************************
230 CLS:KEY OFF:COLOR 11,0,9:LOCATE 10,16:PRINT "┌";STRING$(48,205);"┐"
240 LOCATE 11,16:PRINT "│";STRING$(48,32);"│"
250 LOCATE 12,16:PRINT "│";:LOCATE 12,22:PRINT "PROGRAM TO INPUT THE DATA THAT DRIVES ";:LOCATE 1
2,65:PRINT "│"
260 LOCATE 13,16:PRINT "│";STRING$(48,32);"│"
270 LOCATE 14,16:PRINT "│";:LOCATE 14,20:PRINT "THE SIMULATION MODEL DEVELOPED IN SLAM II";:LOCAT
E 14,65:PRINT "│"
280 LOCATE 15,16:PRINT "│";STRING$(48,32);"│"
290 LOCATE 16,16:PRINT "└";STRING$(48,205);"┘"
300 COLOR 30,0:LOCATE 25,14:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
310 IF INKEY$ < > "" THEN GOTO 310
320 PRO$=INKEY$
330 IF PRO$ < > " " THEN GOTO 320
340 RETURN
350 REM ******************************************
360 REM *  SUBROUTINE TO DISPLAY THE MAIN MENU  *
370 REM ******************************************
380 CLS:KEY OFF:COLOR 10,0,9
390 LOCATE 3,16:PRINT "┌";STRING$(46,205);"┐"
400 LOCATE 4,16:PRINT "│";STRING$(46,32);"│"
410 LOCATE 5,16:PRINT "│";"    PROGRAM TO INPUT DATA THAT DRIVES THE      │"
420 LOCATE 6,16:PRINT "│";"            SLAM SIMULATION PROGRAM            │"
430 LOCATE 7,16:PRINT "│";STRING$(46,32);"│"
440 LOCATE 8,16:PRINT "└";STRING$(46,205);"┘"
450 LOCATE 10,16:PRINT "OPTIONS THAT ARE AVAILABLE : "
460 LOCATE 12,21:PRINT "1.  ENTER A NEW SET OF PARTS DATA "
```

```
470 LOCATE 14,21:PRINT "2.  REVIEW AND ALTER EXISTING DATA "
480 LOCATE 16,21:PRINT "3.  APPEND NEW DATA TO EXISTING ONE "
490 LOCATE 18,21:PRINT "4.  PRINT THE DATA FROM EXISTING FILES "
500 LOCATE 20,21:PRINT "5.  QUIT THE PROGRAM"
510 LOCATE 23,16,1,0,7:INPUT "ENTER YOUR CHOICE : ",CHO
520 IF CHO > 0 AND CHO < 6 THEN GOTO 550
530 SOUND 500,10:CHO=0
540 LOCATE 23,16:PRINT STRING$(50,32):GOTO 510
550 REM ******************************************************
560 REM *  CALLING THE EXACT SUBROUTINE FOR THE CHOICE MADE  *
570 REM ******************************************************
580 ON CHO GOTO 600,2440,5080,6330,590
590 GOTO 190
600 REM **********************************************
610 REM *  SUBROUTINE THAT ENTERS A SET OF NEW DATA  *
620 REM **********************************************
630 CLS:COLOR 27,0
640 LOCATE 4,30:PRINT "INSTRUCTIONS":COLOR 11,0
650 LOCATE 6,11:PRINT "YOU ARE REQUESTED TO INPUT THE PART DETAILS AND THE "
660 LOCATE 7,11:PRINT "OTHER DETAILS THAT ARE USED TO RUN THE SIMULATION"
670 LOCATE 8,11:PRINT "MODEL OF THE FLEXIBLE MANUFACTURING SYSTEM.  THESE"
680 LOCATE 9,11:PRINT "DATA WILL BE ENTERED ON AN INTERACTIVE BASIS.  PLEASE "
690 LOCATE 10,11:PRINT "RESPOND CAREFULLY TO THE QUERIES.  FILES ARE CREATED"
700 LOCATE 11,11:PRINT "TO STORE THESE DATA.  THE NAME FOR THESE FILES SHOULD"
710 LOCATE 12,11:PRINT "NOT BE GREATER THAN EIGHT CHARACTERS LONG.  THESE FILE"
720 LOCATE 13,11:PRINT "NAMES WILL BE PROVIDED WITH AN APPEND DEPENDING ON THE"
730 LOCATE 14,11:PRINT "DATA CONTAINED IN THE FILE"
740 LOCATE 16,11:PRINT "FILE CONTAINING PART DETAILS---------------------[.PAR]"
750 LOCATE 17,11:PRINT "FILE CONTAINING PRIORITY DETAILS-----------------[.PRI]"
760 LOCATE 18,11:PRINT "FILE CONTAINING FIXTURE TYPE DETAILS-------------[.FIX]"
770 LOCATE 19,11:PRINT "FILE CONTAINING SYSTEM LAYOUT DETAILS------------[.LAY]"
780 LOCATE 20,11:PRINT "FILE CONTAINING SPEED AND TRAVEL TIME DETAILS---[.TIM]"
790 COLOR 10,0:LOCATE 22,11:INPUT "ENTER NAME OF FILE TO STORE THE PART DETAILS : ",FILE1$
800 IF LEN(FILE1$) < = 8 THEN GOTO 830
810 SOUND 500,10:FILE1$="        "
820 LOCATE 22,11:PRINT STRING$(68,32):GOTO 790
830 COLOR 30,0:LOCATE 25,11
840 PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E "
850 IF INKEY$ < > "" THEN GOTO 850
860 PRO$=INKEY$
870 IF PRO$ < > " " THEN GOTO 860
880 FILE1$=FILE1$+".PAR"
890 OPEN FILE1$ FOR OUTPUT AS #1
900 CLS:COLOR 10,0:LOCATE 5,2
910 INPUT "TOTAL NUMBER OF PART TYPES TO BE PRODUCED : ",NPT
920 IF NPT < = 25 THEN GOTO 970
930 LOCATE 7,2:COLOR 12,0:SOUND 500,20
940 PRINT "THE PROGRAM CAN HANDLE A MAXIMUM OF 25 PART TYPES ONLY"
950 FOR IL=1 TO 5000:NEXT IL
960 NPT=0:GOTO 900
```

```
970 FOR I=1 TO NPT
980 FOR IQ=1 TO 220:PARR(I,IQ)=0!:NEXT IQ
990 COLOR 10,0:LOCATE 7,2:PRINT "DATA FOR PART TYPE ";I
1000 LOCATE 8,2:PRINT "---------------------"
1010 LOCATE 9,2:PRINT "BATCH SIZE          = ";
1020 INPUT " ",NBS
1030 LOCATE 10,2:PRINT "NUMBER OF OPERATIONS = ";
1040 INPUT " ",NOPS
1050 PARR(I,1)=I:PARR(I,2)=NBS:PARR(I,3)=NOPS
1060      FOR K=1 TO NOPS
1070           J11=4+(K-1)*21
1080          PRINT:PRINT TAB(2);"PROCESSING TIME FOR OPERATION ";K;"   = ";
1090          INPUT " ",PARR(I,J11)
1100          PRINT TAB(2);"FIXTURE TYPE FOR OPERATION ";K;"      = ";
1110          INPUT " ",PARR(I,J11+1)
1120          PRINT TAB(2);"POSSIBLE MACHINES FOR OPERATION ";K;" = ";
1130          INPUT " ",NALM
1140          LJ=1
1150          FOR L=J11+2 TO J11+NALM+1
1160               PRINT TAB(20);"POSSIBILITY # ";LJ;" = ";
1170               INPUT " ",PARR(I,L)
1180               LJ=LJ+1
1190          NEXT L
1200          PRINT TAB(2);"TOOLS REQUIRED FOR OPERATION ";K;"    = ";
1210          INPUT " ",NTOOL
1220          LN=1
1230          FOR M=J11+11 TO J11+NTOOL+10
1240               PRINT TAB(22);"TOOL NUMBER ";LN;" = ";
1250               INPUT " " ,PARR(I,M)
1260               LN=LN+1
1270          NEXT M
1280     NEXT K
1290 REM ---------------------------------------------
1300 REM  WRITING THE DATA ONTO THE FILES ON HARD DISK
1310 REM ---------------------------------------------
1320 PRINT #1,USING "## ";PARR(I,1);:PRINT #1,USING "### ";PARR(I,2);:PRINT #1,USING "## ";PARR(I
,3);
1330 J1=4:KTOT=0
1340 KTOT=KTOT+1
1350     PRINT #1,USING "###.## ";PARR(I,J1);
1360     PRINT #1,USING "## ";PARR(I,J1+1);
1370     FOR JA=J1+2 TO J1+10
1380         PRINT #1,USING "# ";PARR(I,JA);
1390     NEXT JA
1400     FOR JAB=J1+11 TO J1+20
1410         PRINT #1,USING "## ";PARR(I,JAB);
1420     NEXT JAB
1430     J1=J1+21
1440     IF KTOT < 10 THEN GOTO 1340
1450     A$=" ":PRINT #1,USING "\ \";A$
```

```
1460 COLOR 30,0:LOCATE 25,14:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
1470 IF INKEY$ < > "" THEN GOTO 1470
1480 PRO$=INKEY$
1490 IF PRO$ < > " " THEN GOTO 1480
1500 CLS:NEXT I
1510 CLOSE #1
1520 COLOR 11,0:LOCATE 9,16:PRINT STRING$(45,205)
1530 LOCATE 10,17:PRINT "DATA FOR ";NPT;" PART TYPE(S) HAVE BEEN ENTERED"
1540 LOCATE 11,16:PRINT STRING$(45,205):COLOR 30,0
1550 COLOR 30,0:LOCATE 25,14:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
1560 IF INKEY$ < > "" THEN GOTO 1560
1570 PRO$=INKEY$
1580 IF PRO$ < > " " THEN GOTO 1570
1590 REM ------------------------------------------
1600 REM  CREATING THE FILE FOR THE PRIORITY DETAILS
1610 REM ------------------------------------------
1620 CLS:COLOR 10,0
1630 LOCATE 5,2:INPUT "ENTER FILE NAME FOR STORING PRIORITIES OF THE PART TYPES : ",FILE2$
1640 IF LEN(FILE2$)< = 8 THEN GOTO 1670
1650 SOUND 500,10:FILE2$="        "
1660 LOCATE 5,2:PRINT STRING$(70,32):GOTO 1630
1670 FILE2$=FILE2$+".PRI"
1680 FOR IJK=1 TO 25:JPRPT(IJK)=0:NEXT IJK
1690 PRINT:PRINT:PRINT
1700 OPEN FILE2$ FOR OUTPUT AS #1
1710 FOR IPM=1 TO NPT
1720     PRINT TAB(2);"ANY PRIORITY FOR PART TYPE ";IPM;" [ 1-Yes , 2-No ] ? ";
1730     INPUT " ",IPRIOR
1740     IF IPRIOR=1 THEN JPRPT(IPM)=1
1750 NEXT IPM
1760 FOR IPP=1 TO NPT
1770      PRINT #1,USING "          ###";IPP;:PRINT #1,USING "     ##";JPRPT(IPP)
1780 NEXT IPP
1790 CLOSE #1
1800 COLOR 30,0
1810 LOCATE 25,14:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E
1820 IF INKEY$ < > "" THEN GOTO 1820
1830 PRO$=INKEY$
1840 IF PRO$ < > " " THEN GOTO 1830
1850 REM ----------------------------
1860 REM  FILE TO STORE LAYOUT DETAILS
1870 REM ----------------------------
1880 CLS:COLOR 10,0
1890 LOCATE 5,2:INPUT "ENTER FILE NAME TO STORE FMS LAYOUT DETAILS : ",FILE4$
1900 IF LEN(FILE4$) < = 8 THEN GOTO 1930
1910 SOUND 500,10:FILE4$="        "
1920 LOCATE 5,2:PRINT STRING$(70,32):GOTO 1890
1930 LOCATE 7,2:INPUT "NUMBER OF MACHINES IN THE SYSTEM = ",NMAC
1940 IF NMAC < = 9 THEN GOTO 1990
1950 SOUND 500,10:LOCATE 9,2:COLOR 12,0
```

```
1960 PRINT "THE PROGRAM CAN HANDLE A MAXIMUM OF 9 MACHINES ONLY":FOR IL=1 TO 5000:NEXT IL
1970 NMAC=0:LOCATE 7,2:PRINT STRING$(70,32):LOCATE 9,2:PRINT STRING$(70,32)
1980 COLOR 10,0:GOTO 1930
1990 FILE4$=FILE4$+".LAY"
2000 FOR ID=1 TO 9:DLOM(ID)=0!:DULOM(ID)=0!:DSTM(ID)=0!:ITRACK(ID)=0:NEXT ID
2010 OPEN FILE4$ FOR OUTPUT AS #1
2020 FOR NUM=1 TO NMAC
2030     PRINT:PRINT TAB(2);"THE LANE ON WHICH MACHINE ";NUM;" IS LOCATED                 = ";:IN
PUT " ",ITRACK(NUM)
2040     PRINT TAB(2);"DISTANCE BETWEEN LOADING STATION & MACHINE ";NUM;" [ mts ]   = ";
2050     INPUT " ",DLOM(NUM)
2060     PRINT TAB(2);"DISTANCE BETWEEN MACHINE ";NUM;" & UNLOADING STATION [ mts ] = ";
2070     INPUT " ",DULOM(NUM)
2080     PRINT TAB(2);"DISTANCE BETWEEN TOOL CRIB AND THE MACHINE ";NUM;" [ mts ]   = ";
2090     INPUT " ",DSTM(NUM)
2100     PRINT #1,USING "     ###";NUM;:PRINT #1,USING "     ##";ITRACK(NUM);:PRINT #1,USING "
      ####.##";DLOM(NUM);
2110     PRINT #1,USING "     ####.##";DULOM(NUM);:PRINT #1,USING "     ####.##";DSTM(NUM)
2120 NEXT NUM
2130 CLOSE #1
2140 PRINT:PRINT
2150 COLOR 30,0:LOCATE 25,14:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
2160 IF INKEY$ < > "" THEN GOTO 2160
2170 PRO$=INKEY$
2180 IF PRO$ < > " " THEN GOTO 2170
2190 REM -------------------------------------
2200 REM  FILE TO STORE TIME AND SPEED DETAILS
2210 REM -------------------------------------
2220 CLS:COLOR 10,0
2230 LOCATE 5,2:INPUT "ENTER FILE NAME TO STORE TIME AND SPEED DETAILS OF THE  M H S  : ",FILE5$
2240 IF LEN(FILE5$) < = 8 THEN GOTO 2270
2250 SOUND 500,10:FILE5$=FILE5$+"       "
2260 LOCATE 5,2:PRINT STRING$(70,32):GOTO 2230
2270 FILE5$=FILE5$+".TIM"
2280 OPEN FILE5$ FOR OUTPUT AS #1
2290 LOCATE 8,2:INPUT "ENTER SPEED OF THE LOADING AGV [ mts/min ]          = ",SPAGV
2300 LOCATE 9,2:INPUT "ENTER SPEED OF THE UNLOADING AGV [ mts/min ]        = ",SPUAGV
2310 LOCATE 10,2:INPUT "ENTER SPEED OF THE TOOL CONVEYOR [ mts/min ]        = ",SPCON
2320 LOCATE 11,2:INPUT "ENTER TRAVEL TIME FOR THE LOADING CRANE [ mins ]    = ",TTSLO
2330 LOCATE 12,2:INPUT "ENTER TRAVEL TIME FOR THE UNLOADING CRANE [ mins ]  = ",TTULSO
2340 PRINT #1,USING "          ###.###";SPAGV;:PRINT #1,USING "     ###.###";SPUAGV;
2350 PRINT #1,USING "     ###.###";SPCON;:PRINT #1,USING "                    ###.###";TTSLO;:PRI
NT #1,USING "     ###.###";TTULSO
2360 CLOSE #1
2370 COLOR 30,0
2380 LOCATE 21,14:PRINT "P R E S S   S P A C E   B A R   T O   R E T U R N"
2390 LOCATE 22,14:PRINT "         T O   T H E   M A I N   M E N U"
2400 IF INKEY$ < > "" THEN GOTO 2400
2410 PRO$=INKEY$
2420 IF PRO$ < > " " THEN GOTO 2410
```

```
2430 GOTO 350
2440 REM ********************************************************
2450 REM *  SUBROUTINE TO REVIEW AND ALTER THE EXISTING DATA  *
2460 REM ********************************************************
2470 CLS:COLOR 10,0:LOCATE 5,16:PRINT "OPTIONS THAT ARE AVAILABLE : "
2480 LOCATE 7,21:PRINT "1.  PART TYPE DETAILS"
2490 LOCATE 9,21:PRINT "2.  PART TYPE PRIORITIES"
2500 LOCATE 11,21:PRINT "3.  LAYOUT DETAILS"
2510 LOCATE 13,21:PRINT "4.  TIME AND SPEED DETAILS"
2520 LOCATE 15,21:PRINT "5.  QUIT THIS SEGMENT"
2530 LOCATE 18,16:INPUT "ENTER YOUR CHOICE : ",CHO1
2540 IF CHO1 > 0 OR CHO1 < 6 THEN GOTO 2570
2550 SOUND 500,10:CHO1=0
2560 LOCATE 18,16:PRINT STRING$(50,32):GOTO 2530
2570 ON CHO1 GOTO 2590,3910,4260,4720,350
2580 REM ------------------------------------
2590 REM  SUBROUTINE TO REVIEW PART DETAILS
2600 REM ------------------------------------
2610 IREC=0
2620 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN PART TYPE DETAILS : ";:FILES "C:*.PAR"
2630 PRINT:COLOR 10,0
2640 PRINT:PRINT TAB(2);"ENTER NAME OF FILE CONTAINING PART TYPE DETAILS : ";:INPUT " ",FILE1$:FI
LE1$=FILE1$+".PAR"
2650 OPEN FILE1$ FOR INPUT AS #1
2660 IREC=IREC+1:IF EOF(1) THEN GOTO 2760
2670 FOR INIR=1 TO 220:PARR(IREC,INIR)=0!:NEXT INIR
2680      INPUT #1,PARR(IREC,1),PARR(IREC,2),PARR(IREC,3)
2690      FOR K=1 TO 10
2700          J1=4+(K-1)*21
2710          INPUT #1,PARR(IREC,J1),PARR(IREC,J1+1)
2720          FOR JA=J1+2 TO J1+10:INPUT #1,PARR(IREC,JA):NEXT JA
2730          FOR JAB=J1+11 TO J1+20:INPUT #1,PARR(IREC,JAB):NEXT JAB
2740      NEXT K
2750 GOTO 2660
2760 NTPT=IREC-1
2770 CLOSE #1
2780 INDE=0:CLS:LOCATE 5,2:INPUT "ENTER THE PART TYPE WHOSE DETAIL YOU WANT TO REVIEW : " ,IPA
2790 IF IPA > 25 THEN GOTO 2810
2800 IF PARR(IPA,1) > 0! THEN GOTO 2840
2810 SOUND 500,10:LOCATE 7,2:COLOR 12,0:PRINT "PART TYPE ";IPA;" DOES NOT EXIST - TRY AGAIN "
2820 FOR KL=1 TO 3000:NEXT KL
2830 IPA=0:COLOR 10,0:GOTO 2780
2840 LOCATE 7,2:COLOR 10,0:PRINT "ENTER THE OPERATION NUMBER OF PART TYPE ";IPA;" YOU WANT TO REV
IEW : ";:INPUT " ",LNOP
2850 J11=4+(LNOP-1)*21
2860 IF PARR(IPA,J11) > 0! THEN GOTO 2890
2870 SOUND 500,10:LOCATE 9,2:COLOR 12,0:PRINT "OPERATION # ";LNOP;" DOES NOT EXIST - TRY AGAIN"
2880 FOR KI=1 TO 2500:NEXT KI:LOCATE 9,2:PRINT STRING$(75,32):LOCATE 7,2:PRINT STRING$(75,32):GOT
O 2840
2890 LOCATE 9,2:COLOR 11,0:PRINT "OPERATION NUMBER ";LNOP;" OF PART TYPE ";IPA
```

```
2900 PRINT TAB(2);"-----------------------------------"
2910 PRINT TAB(6);INDE+1;".   PROCESSING TIME ----- ";PARR(IPA,J11)
2920 IFTY=PARR(IPA,J11+1):PRINT TAB(6);INDE+2;".   FIXTURE TYPE -------- ";IFTY
2930 PRINT TAB(6);INDE+3;".   ALTERNATE MACHINES -- ";
2940 ICDE=0
2950 FOR JAY=J11+2 TO J11+10
2960     ICDE=ICDE+1
2970     IALMAC=PARR(IPA,JAY):IF IALMAC = 0 THEN GOTO 3020
2980     IF ICDE > 1 THEN GOTO 3000
2990     PRINT IALMAC;:GOTO 3010
3000     PRINT "or";IALMAC;
3010 NEXT JAY
3020 PRINT
3030 PRINT TAB(6);INDE+4;".   TOOLS REQUIRED  ----- ";
3040 ICTA=0
3050 FOR JAT=J11+11 TO J11+20
3060     ICTA=ICTA+1
3070     IALTO=PARR(IPA,JAT):IF IALTO = 0 THEN GOTO 3120
3080     IF ICTA > 1 THEN GOTO 3100
3090     PRINT IALTO;:GOTO 3110
3100     PRINT ",";IALTO;
3110 NEXT JAT
3120 PRINT
3130 LABS=PARR(IPA,2)
3140 PRINT TAB(6);INDE+5;".   BATCH SIZE ---------- ";LABS
3150 COLOR 10,0:LOCATE 18,2:INPUT "ANY CHANGES TO BE MADE [ Y-Yes, N-No ] ? ",AN$
3160 IF AN$="Y" OR AN$="y" THEN GOTO 3210
3170 IF AN$="N" OR AN$="n" THEN GOTO 3610
3180 SOUND 500,10:AN$="   "
3190 LOCATE 18,2:PRINT STRING$(50,32):GOTO 3150
3200 REM --------------------
3210 REM  CHANGING THE DATA
3220 REM --------------------
3230 LOCATE 19,2:INPUT "ENTER YOUR CHOICE [ Number before each line ] : ",CHO2
3240 ON CHO2 GOTO 3270,3310,3350,3450,3550
3250 SOUND 500,10:CHO2=0
3260 LOCATE 19,2:PRINT STRING$(70,32):GOTO 3230
3270 REM ----------------------
3280 REM  NEW PROCESSING TIME
3290 REM ----------------------
3300 CLS:LOCATE 22,2:PRINT  "NEW PROCESSING TIME FOR OPERATION ";LNOP;" OF PART TYPE ";IPA;" = ";
:INPUT " ",PRT1:PARR(IPA,J11)=PRT1:CLS:GOTO 2850
3310 REM ------------------
3320 REM  NEW FIXTURE TYPE
3330 REM ------------------
3340 CLS:LOCATE 22,2:PRINT  "NEW FIXTURE TYPE FOR OPERATION ";LNOP;" OF PART TYPE ";IPA;" = ";:IN
PUT " ",IFT1:PARR(IPA,J11+1)=IFT1:CLS:GOTO 2850
3350 REM --------------
3360 REM  NEW MACHINES
3370 REM --------------
```

```
3380 CLS:LOCATE 5,2
3390 FOR IMT=J11+2 TO J11+10:PARR(IPA,IMT)=0!:NEXT IMT
3400 PRINT "POSSIBLE MACHINES FOR OPERATION ";LNOP;" OF PART TYPE ";IPA;" = ";:INPUT " ",NALM
3410 FOR IAL2=1 TO NALM
3420     PRINT TAB(36);"POSSIBILITY # ";IAL2;"  = ";:INPUT " ",IM1:PARR(IPA,J11+IAL2+1)=IM1
3430 NEXT IAL2
3440 CLS:GOTO 2850
3450 REM -----------
3460 REM  NEW TOOLS
3470 REM -----------
3480 CLS:LOCATE 5,2
3490 FOR IMG=J11+11 TO J11+20:PARR(IPA,IMG)=0!:NEXT IMG
3500 PRINT "TOOLS REQUIRED FOR OPERATION ";LNOP;" OF PART TYPE ";IPA;" = ";:INPUT " ",NTOOL
3510 FOR IALT1=1 TO NTOOL
3520     PRINT TAB(36);"TOOL NUMBER ";IALT1;" = ";:INPUT " ",IT1:PARR(IPA,J11+IALT1+10)=IT1
3530 NEXT IALT1
3540 CLS:GOTO 2850
3550 REM ----------------
3560 REM  NEW BATCH SIZE
3570 REM ----------------
3580 CLS:LOCATE 22,2:PRINT "NEW BATCH SIZE FOR PART TYPE ";IPA;" = ";
3590 INPUT " " ,NB1:PARR(IPA,2)=NB1:CLS:GOTO 2850
3600 REM --------------------------
3610 REM  CHECK TO PROCEED FURTHER
3620 REM --------------------------
3630 LOCATE 19,2:INPUT "ANY OTHER PART DETAIL TO BE REVIEWED [ Y-Yes, N-No ] ? ",AN$
3640 IF AN$="Y" OR AN$="y" THEN GOTO 2780
3650 IF AN$="N" OR AN$="n" THEN GOTO 3690
3660 SOUND 500,10:AN$="  "
3670 LOCATE 19,2:PRINT STRING$(70,32):GOTO 3630
3680 REM -----------------------------------------
3690 REM  PUTTING BACK THE DETAILS IN THEIR FILES
3700 REM -----------------------------------------
3710 OPEN FILE1$ FOR OUTPUT AS #1
3720 FOR IPR=1 TO 25
3730     IF PARR(IPR,1) = 0! THEN GOTO 3890
3740     PRINT #1,USING "## ";PARR(IPR,1);:PRINT #1,USING "### ";PARR(IPR,2);:PRINT #1,USING "##
";PARR(IPR,3);
3750     J1=4:KTOT=0
3760     KTOT=KTOT+1
3770     PRINT #1,USING "###.## ";PARR(IPR,J1);
3780     PRINT #1,USING "## ";PARR(IPR,J1+1);
3790     FOR JA=J1+2 TO J1+10
3800         PRINT #1,USING "# ";PARR(IPR,JA);
3810     NEXT JA
3820     FOR JAB=J1+11 TO J1+20
3830         PRINT #1,USING "## ";PARR(IPR,JAB);
3840     NEXT JAB
3850     J1=J1+21
3860     IF KTOT < 10 THEN GOTO 3760
```

```
3870     A$=" ":PRINT #1,USING "\ \";A$
3880 NEXT IPR
3890 CLOSE #1
3900 GOTO 2440
3910 REM ----------------------------------------
3920 REM  SUBROUTINE TO REVIEW PRIORITY DETAILS
3930 REM ----------------------------------------
3940 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN PRIORITY DETAILS FOR VARIOUS PART TYPE
S : ";:FILES "C:*.PRI"
3950 COLOR 10,0:PRINT:PRINT TAB(2);"ENTER NAME OF FILE CONTAINING PRIORITY DETAILS : ";:INPUT " "
,FILE2$:FILE2$=FILE2$+".PRI"
3960 IPRIOR=0:FOR IPG=1 TO 25:JPRPT(IPG)=0:NEXT IPG
3970 OPEN FILE2$ FOR INPUT AS #1
3980 IPRIOR=IPRIOR+1:IF EOF(1) THEN GOTO 4010
3990 INPUT #1,IPX,JPRPT(IPRIOR)
4000 GOTO 3980
4010 CLOSE #1
4020 CLS:COLOR 11,0:LOCATE 5,2:PRINT "PRIORITY DETAILS FOR THE PART TYPES "
4030 LOCATE 6,2:PRINT "-----------------------------------"
4040 FOR IPM=1 TO IPRIOR-1
4050     IF JPRPT(IPM) = 1 THEN PRNT$="YES" ELSE PRNT$="NO"
4060     PRINT TAB(6);IPM;".  PRIORITY FOR PART TYPE ";IPM;" --- ";PRNT$
4070 NEXT IPM
4080 COLOR 10,0:LOCATE 22,2:INPUT "ANY CHANGES TO BE MADE [ Y-Yes, N-No ] ?  ",AN$
4090 IF AN$="Y" OR AN$="y" THEN GOTO 4130
4100 IF AN$="N" OR AN$="n" THEN GOTO 4200
4110 SOUND 500,10:AN$="  "
4120 LOCATE 22,2:PRINT STRING$(70,32):GOTO 4080
4130 LOCATE 23,2:INPUT "ENTER YOUR CHOICE [ Number before each line ] : ",ICHO3
4140 IF ICHO3 > 0 AND ICHO3 < IPRIOR THEN GOTO 4170
4150 SOUND 500,10:ICHO3=0
4160 LOCATE 23,2:PRINT STRING$(70,32):GOTO 4130
4170 CLS:LOCATE 22,2:PRINT "NEW PRIORITY FOR PART TYPE [ 1-Yes , 2-No ] ";ICHO3;" = ";:INPUT " ",
JPRPT(ICHO3)
4180 IF JPRPT(ICHO3) = 1 OR JPRPT(ICHO3) = 2 THEN GOTO 4020
4190 SOUND 500,10:LOCATE 22,2:PRINT STRING$(70,32):GOTO 4170
4200 OPEN FILE2$ FOR OUTPUT AS #1
4210 FOR IPT=1 TO IPRIOR-1
4220   PRINT #1,USING "          ###";IPT;:PRINT #1,USING "     ##";JPRPT(IPT)
4230 NEXT IPT
4240 CLOSE #1
4250 GOTO 2440
4260 REM  ----------------------------------------
4270 REM   SUBROUTINE FOR REVIEWING LAYOUT DETAILS
4280 REM  ----------------------------------------
4290 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN THE FMS LAYOUT DETAILS : ";:FILES "C:*
.LAY"
4300 COLOR 10,0:PRINT:PRINT TAB(2);"ENTER THE NAME OF THE FILE THAT CONTAINS THE LAYOUT DETAILS :
 ";:INPUT " ",FILE4$:FILE4$=FILE4$+".LAY"
4310 IMC=0:FOR IMB=1 TO 9:DLOM(IMB)=0!:DULOM(IMB)=0!:DSTM(IMB)=0!:ITRACK(IMB)=0:NEXT IMB
```

```
4320 OPEN FILE4$ FOR INPUT AS #1
4330 IMC=IMC+1:IF EOF(1) THEN GOTO 4360
4340 INPUT #1,NUM,ITRACK(IMC),DLOM(IMC),DULOM(IMC),DSTM(IMC)
4350 GOTO 4330
4360 CLOSE #1
4370 CLS:LOCATE 5,2:COLOR 10,0:PRINT "ENTER THE MACHINE WHOSE LAYOUT DETAIL YOU WANT TO REFER TO
: ";:INPUT " ",MAC
4380 TM=IMC-1
4390 IF MAC > 0 AND MAC < TM+1 THEN GOTO 4420
4400 SOUND 500,10:MAC=0
4410 COLOR 12,0:LOCATE 7,2:PRINT "SYSTEM HAS ONLY ";IMC-1;" MACHINES - TRY AGAIN":FOR KL = 1 TO 3
000:NEXT KL:LOCATE 5,2:PRINT STRING$(70,32):GOTO 4370
4420 COLOR 11,0:LOCATE 7,2:PRINT "1.  THE LANE ON WHICH MACHINE ";MAC;" IS LOCATED ---------- ";I
TRACK(MAC)
4430 LOCATE 8,2:PRINT "2.  DISTANCE BETWEEN MACHINE ";MAC;" AND LOADING STATION -- ";DLOM(MAC)
4440 LOCATE 9,2:PRINT "3.  DISTANCE BETWEEN MACHINE ";MAC;" UNLOADING STATION ---- ";DULOM(MAC)
4450 LOCATE 10,2:PRINT "4.  DISTANCE BETWEEN MACHINE ";MAC;" AND TOOL CRIB -------- ";DSTM(MAC)
4460 COLOR 10,0:LOCATE 15,2:INPUT "ANY CHANGES TO BE MADE [ Y-Yes, N-No ] ? ",AN$
4470 IF AN$="Y" OR AN$="y" THEN GOTO 4510
4480 IF AN$="N" OR AN$="n" THEN GOTO 4600
4490 SOUND 500,10:AN$=" "
4500 LOCATE 15,2:PRINT STRING$(50,32): GOTO 4460
4510 LOCATE 16,2:INPUT "ENTER YOUR CHOICE [ Number before each line ] : ",IMO1
4520 IF IMO1 > 0 AND IMO1 < 5 THEN GOTO 4550
4530 SOUND 500,10:IMO1=0
4540 LOCATE 16,2:PRINT STRING$(70,32):GOTO 4510
4550 ON IMO1 GOTO 4560,4570,4580,4590
4560 CLS:LOCATE 20,2:PRINT "THE NEW LANE ON WHICH MACHINE ";MAC;" IS LOCATED = ";:INPUT " ",ITRAC
K(MAC):CLS:GOTO 4420
4570 CLS:LOCATE 20,2:PRINT "NEW DISTANCE BETWEEN MACHINE ";MAC;" AND LOADING STATION = ";:INPUT "
 ",DLOM(MAC):CLS:GOTO 4420
4580 CLS:LOCATE 20,2:PRINT "NEW DISTANCE BETWEEN MACHINE ";MAC;" AND UNLOADING STATION = ";:INPUT
 " ",DULOM(MAC):CLS:GOTO 4420
4590 CLS:LOCATE 20,2:PRINT "NEW DISTANCE BETWEEN MACHINE ";MAC;" AND THE CENTRAL TOOL STORAGE = "
;:INPUT " ",DSTM(MAC):CLS: GOTO 4420
4600 LOCATE 16,2:INPUT "ANY MORE MACHINE LAYOUT DETAILS TO BE REVIEWED [ Y-Yes , N-No ] ? ",AN$
4610 IF AN$="Y" OR AN$="y" THEN GOTO 4370
4620 IF AN$="N" OR AN$="n" THEN GOTO 4650
4630 SOUND 500,10:AN$=" "
4640 LOCATE 16,2:PRINT STRING$(70,32):GOTO 4600
4650 OPEN FILE4$ FOR OUTPUT AS #1
4660 FOR NMA=1 TO IMC-1
4670   PRINT #1,USING "     ###";NMA;:PRINT #1,USING "     ##";ITRACK(NMA);:PRINT #1,USING "
    ####.##";DLOM(NMA);
4680   PRINT #1,USING "     ####.##";DULOM(NMA);:PRINT #1,USING "     ####.##";DSTM(NMA)
4690 NEXT NMA
4700 CLOSE #1
4710 GOTO 2440
4720 REM ----------------------------------------------------------
4730 REM  SUBROUTINE TO CHANGE THE SPEED AND TIME DETAILS OF THE MHS
```

```
4740 REM ------------------------------------------------------------
4750 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN TIME AND SPEED DETAILS OF THE  M H S
: ";:FILES "C:*.TIM"
4760 COLOR 10,0:PRINT:PRINT TAB(2);:INPUT "ENTER NAME OF FILE THAT CONTAINS THE SPEED DETAILS : "
,FILE5$:FILE5$=FILE5$+".TIM"
4770 OPEN FILE5$ FOR INPUT AS #1
4780 INPUT #1,SPAGV,SPUAGV,SPCON,TTSLO,TTULSO
4790 CLOSE #1
4800 CLS:COLOR 11,0
4810 LOCATE 5,2:PRINT "SPEED AND TRAVEL TIME DETAILS OF THE  M H S "
4820 LOCATE 6,2:PRINT "---------------------------------------------"
4830 LOCATE 8,2:PRINT "1.  SPEED OF THE LOADING AGV -------------- ";SPAGV;" [mts/min]"
4840 LOCATE 9,2:PRINT "2.  SPEED OF THE UNLOADING AGV ------------ ";SPUAGV;" [mts/min]"
4850 LOCATE 10,2:PRINT "3.  SPEED OF THE TOOL CONVEYOR ------------ ";SPCON;" [mts/min]"
4860 LOCATE 11,2:PRINT "4.  TRAVEL TIME FOR THE LOADING CRANE ----- ";TTSLO;"[mins]
4870 LOCATE 12,2:PRINT "5.  TRAVEL TIME FOR THE UNLOADING CRANE --- ";TTULSO;"[mins]"
4880 COLOR 10,0:LOCATE 15,2:INPUT "ANY CHANGES TO BE MADE [ Y-Yes, N-No ] ? ",AN$
4890 IF AN$="Y" OR AN$="y" THEN GOTO 4930
4900 IF AN$="N" OR AN$="n" THEN GOTO 5030
4910 SOUND 500,10:AN$="  "
4920 LOCATE 15,2:PRINT STRING$(70,32):GOTO 4880
4930 LOCATE 16,2:INPUT "ENTER YOUR CHOICE [ Number before each line ] : ",ISP1
4940 IF ISP1 > 0 AND ISP1 < 6 THEN GOTO 4970
4950 SOUND 500,10:ISP1=0
4960 LOCATE 16,2:PRINT STRING$(70,32):GOTO 4930
4970 ON ISP1 GOTO 4980,4990,5000,5010,5020
4980 CLS:LOCATE 20,2:INPUT "NEW SPEED OF LOADING AGV [ mts/min ] = ",SPAGV:GOTO 4800
4990 CLS:LOCATE 20,2:INPUT "NEW SPEED OF THE UNLOADING AGV [ mts/min ] = ",SPUAGV:GOTO 4800
5000 CLS:LOCATE 20,2:INPUT "NEW SPEED OF THE TOOL CONVEYOR [ mts/min ] = ",SPCON:GOTO 4800
5010 CLS:LOCATE 20,2:INPUT "NEW TRAVEL TIME FOR THE LOADING CRANE [ mins ] = ",TTSLO:GOTO 4800
5020 CLS:LOCATE 20,2:INPUT "NEW TRAVEL TIME FOR THE UNLOADING CRANE [ mins ] = ",TTULSO:GOTO 4800
5030 OPEN FILE5$ FOR OUTPUT AS #1
5040 PRINT #1,USING "          ###.###";SPAGV;:PRINT #1,USING "     ###.###";SPUAGV;:PRINT #1,USI
NG "     ###.###";SPCON;
5050 PRINT #1,USING "                    ###.###";TTSLO;:PRINT #1,USING "     ###.###";TTULSO
5060 CLOSE #1
5070 GOTO 2440
5080 REM *****************************************************
5090 REM * SUBROUTINE TO APPEND DATA TO THE EXISTING FILES  *
5100 REM *****************************************************
5110 CLS:LOCATE 7,16:COLOR 10,0:PRINT "OPTIONS THAT ARE AVAILABLE : "
5120 LOCATE 10,21:PRINT "1.  ADD ANOTHER PART TYPE"
5130 LOCATE 12,21:PRINT "2.  ADD ANOTHER MACHINE "
5140 LOCATE 14,21:PRINT "3.  QUIT THIS SEGMENT"
5150 LOCATE 17,16:INPUT "ENTER YOUR CHOICE : ",CH04
5160 IF CH04 > 0 OR CH04 < 4 THEN GOTO 5190
5170 SOUND 500,10:CH04=0
5180 LOCATE 17,16:PRINT STRING$(70,32):GOTO 5150
5190 ON CH04 GOTO 5200,5980,350
5200 REM ----------------------------
```

```
5210 REM  ROUTINE TO APPEND PART TYPE
5220 REM -----------------------------
5230 CLS:LOCATE 5,2:COLOR 11,0:PRINT "FILE(S) THAT CONTAIN PART TYPE DETAILS : ";:FILES "C:*.PAR"
5240 COLOR 10,0:PRINT:PRINT TAB(2);:INPUT "ENTER NAME OF FILE THAT CONTAINS PART TYPE DETAILS : "
,FILE1$:FILE1$=FILE1$+".PAR"
5250 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN PRIORITY DETAILS : ";:FILES "C:*.PRI"
5260 COLOR 10,0:PRINT:PRINT TAB(2);:INPUT "ENTER NAME OF FILE THAT CONTAINS PRIORITY DETAILS : ",
FILE2$:FILE2$=FILE2$+".PRI"
5270 IREC=0:IPRIOR=0
5280 OPEN FILE1$ FOR INPUT AS #1
5290 IREC=IREC+1:KTOT=0:IF EOF(1) THEN GOTO 5390
5300 FOR INY=1 TO 220:PARR(IREC,INY)=0!:NEXT INY
5310     INPUT #1,PARR(IREC,1),PARR(IREC,2),PARR(IREC,3)
5320     FOR K=1 TO 10
5330         J11=4+(K-1)*21
5340         INPUT #1,PARR(IREC,J11),PARR(IREC,J11+1)
5350         FOR JA=J11+2 TO J11+10:INPUT #1,PARR(IREC,JA):NEXT JA
5360         FOR JAB=J11+11 TO J11+20:INPUT #1,PARR(IREC,JAB):NEXT JAB
5370     NEXT K
5380 GOTO 5290
5390 NTPT=IREC
5400 CLOSE #1
5410 FOR JHY=1 TO 25:JPRPT(JHY)=0:NEXT JHY
5420 OPEN FILE2$ FOR INPUT AS #2
5430      IPRIOR=IPRIOR+1:IF EOF(2) THEN GOTO 5460
5440      INPUT #2,IPE,JPRPT(IPRIOR)
5450      GOTO 5430
5460 CLOSE #2
5470 NPTY=IPRIOR
5480 OPEN FILE1$ FOR APPEND AS #1
5490 L1=1:M1=1
5500 FOR IND=1 TO 220:PARR(NTPT,IND)=0!:NEXT IND
5510 CLS:COLOR 10,0:LOCATE 7,2:PRINT "DATA FOR PART TYPE ";NTPT
5520 LOCATE 8,2:PRINT "----------------------"
5530 LOCATE 9,2:PRINT "BATCH SIZE           = ";:INPUT " ",NBS
5540 LOCATE 10,2:PRINT "NUMBER OF OPERATIONS = ";:INPUT " ",NOPS
5550 PARR(NTPT,1)=NTPT:PARR(NTPT,2)=NBS:PARR(NTPT,3)=NOPS
5560 FOR K=1 TO NOPS
5570     J1=4+(K-1)*21
5580     PRINT:PRINT TAB(2);"PROCESSING TIME FOR OPERATION ";K;"   = ";:INPUT " ",PARR(NTPT,J1)
5590     PRINT TAB(2);"FIXTURE TYPE FOR OPERATION ";K;"      = ";:INPUT " ",PARR(NTPT,J1+1)
5600     PRINT TAB(2);"POSSIBLE MACHINES FOR OPERATION ";K;" = ";:INPUT " ",NALM
5610     LJ=1
5620     FOR L=J1+2 TO J1+NALM+1
5630         PRINT TAB(20);"POSSIBILITY #";LJ;"  = ";:INPUT " ",PARR(NTPT,L)
5640         LJ=LJ+1
5650     NEXT L
5660     L1=L1+9
5670     PRINT TAB(2);"TOOLS REQUIRED FOR OPERATION ";K;"    = ";:INPUT " ",NTOOL
5680     LN=1
```

```
5690     FOR M=J1+11 TO J1+NTOOL+10
5700         PRINT TAB(20);"TOOL NUMBER ";LN;"   = ";:INPUT " " ,PARR(NTPT,M)
5710         LN=LN+1
5720     NEXT M
5730     M1=M1+10
5740 NEXT K
5750 PRINT #1,USING "## ";PARR(NTPT,1);:PRINT #1,USING "### ";PARR(NTPT,2);:PRINT #1,USING "## ";
PARR(NTPT,3);
5760     J3=4:KTOT=0
5770     KTOT=KTOT+1
5780     PRINT #1,USING "###.## ";PARR(NTPT,J3);
5790     PRINT #1,USING "## ";PARR(NTPT,J3+1);
5800     FOR JA=J3+2 TO J3+10:PRINT #1,USING "# ";PARR(NTPT,JA);:NEXT JA
5810     FOR JAB=J3+11 TO J3+20:PRINT #1,USING "## ";PARR(NTPT,JAB);:NEXT JAB
5820     J3=J3+21
5830     IF KTOT < 10 THEN GOTO 5770
5840     A$=" ":PRINT #1,USING "\ \";A$
5850 OPEN FILE2$ FOR APPEND AS #2
5860      PRINT:PRINT:PRINT TAB(2);"ANY PRIORITY FOR PART TYPE ";NPTY;" [ 1-Yes , 2-No ] ? ";:INP
UT " ",IPRO
5870      IF IPRO=1 THEN JPRPT(NPTY)=1
5880      PRINT #2,USING "          ###";NPTY;:PRINT #2,USING "     ##";JPRPT(NPTY)
5890 CLOSE #2
5900 PRINT:PRINT TAB(2);:INPUT "ANY MORE ADDITIONS [ Y-Yes, N-No ] ? ",AN$
5910 IF AN$="Y" OR AN$="y" THEN GOTO 5930
5920 IF AN$="N" OR AN$="n" THEN GOTO 5940
5930 NPTY=NPTY+1:NTPT=NTPT+1:GOTO 5490
5940 CLOSE #1:GOTO 5080
5950 SOUND 500,10:AN$="  "
5960 LOCATE ,,2:PRINT STRING$(70,32): GOTO 5900
5970 REM -----------------------------
5980 REM  ADDITION OF ANOTHER MACHINE
5990 REM -----------------------------
6000 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN THE FMS LAYOUT DETAILS : ";:FILES "C:*
.LAY"
6010 COLOR 10,0:PRINT:PRINT TAB(2);"ENTER NAME OF FILE THAT CONTAINS THE FMS LAYOUT DETAILS : ";:
INPUT " ",FILE4$:FILE4$=FILE4$+".LAY"
6020 ILA=0
6030 FOR MA=1 TO 9:DULOM(MA)=0!:DLOM(MA)=0!:DSTM(MA)=0!:ITRACK(MA)=0:NEXT MA
6040 OPEN FILE4$ FOR INPUT AS #1
6050 ILA=ILA+1:IF EOF(1) THEN GOTO 6080
6060 INPUT #1,ILT,ITRACK(ILA),DLOM(ILA),DULOM(ILA),DSTM(ILA)
6070 GOTO 6050
6080 CLOSE #1
6090 IF ILA < 10 THEN GOTO 6180
6100 COLOR 11,0
6110 LOCATE 10,16:PRINT "SORRY, BUT THE SIMULATION MODEL CAN ONLY ALLOW A "
6120 LOCATE 12,16:PRINT "MAXIMUM OF NINE MACHINES.  YOU ARE EXCEEDING THE "
6130 LOCATE 14,16:PRINT "LIMITS OF THE SIMULATION MODEL.  ":COLOR 30,0:LOCATE 22,14
6140 PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E "
```

```
6150 IF INKEY$ < > "" THEN GOTO 6150
6160 PRO$=INKEY$
6170 IF PRO$ < > " " THEN GOTO 6160
6180 OPEN FILE4$ FOR APPEND AS #1
6190 CLS:COLOR 11,0:LOCATE 5,2:PRINT "LAYOUT DETAILS FOR MACHINE ";ILA
6200 LOCATE 6,2:PRINT "----------------------------"
6210 COLOR 10,0:LOCATE 8,2:PRINT "THE LANE ON WHICH MACHINE ";ILA;" IS LOCATED            = ";:
INPUT " ",ITRACK(ILA)
6220 LOCATE 9,2:PRINT "DISTANCE BETWEEN MACHINE ";ILA;" AND THE LOADING STATION  = ";:INPUT " ",D
LOM(ILA)
6230 LOCATE 10,2:PRINT "DISTANCE BETWEEN MACHINE ";ILA;" AND UNLOADING STATION    = ";:INPUT " ",
DULOM(ILA)
6240 LOCATE 11,2:PRINT "DISTANCE BETWEEN MACHINE ";ILA;" AND THE TOOL STORAGE     = ";:INPUT " ",
DSTM(ILA)
6250     PRINT #1,USING "     ###";ILA;:PRINT #1,USING "     ##";ITRACK(ILA);:PRINT #1,USING "
      ####.##";DLOM(ILA);
6260     PRINT #1,USING "     ####.##";DULOM(ILA);:PRINT #1,USING "     ####.##";DSTM(ILA)
6270 LOCATE 13,2:INPUT "ANY MORE ADDITIONS [ Y-Yes, N-No ] ? ",AN$
6280 IF AN$="Y" OR AN$="y" THEN ILA=ILA+1:GOTO 6090 ELSE 6290
6290 IF AN$="N" OR AN$="n" THEN GOTO 6320
6300 SOUND 500,10:AN$=" "
6310 LOCATE 12,2:PRINT STRING$(60,32):GOTO 6270
6320 CLOSE #1:GOTO 5080
6330 REM ********************************************
6340 REM * SUBROUTINE TO PRINT OUT PART DETAILS  *
6350 REM ********************************************
6360 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN PART TYPE DETAILS : ";:FILES "C:*.PAR"
6370 PRINT:COLOR 10,0
6380 PRINT TAB(2);"ENTER NAME OF FILE CONTAINING PART TYPE DETAILS : ";:INPUT " ",FILE1$:FILE1$=F
ILE1$+".PAR"
6390 IREC=0
6400 OPEN FILE1$ FOR INPUT AS #1
6410 IREC=IREC+1:KTOT=0:IF EOF(1) THEN GOTO 6520
6420 FOR INR=1 TO 220:PARR(IREC,INR)=0!:NEXT INR
6430     INPUT #1,PARR(IREC,1),PARR(IREC,2),PARR(IREC,3)
6440     FOR K=1 TO 10
6450         J4=4+(K-1)*21
6460         INPUT #1,PARR(IREC,J4)
6470         INPUT #1,PARR(IREC,J4+1)
6480         FOR JA=J4+2 TO J4+10:INPUT #1,PARR(IREC,JA):NEXT JA
6490         FOR JAB=J4+11 TO J4+20:INPUT #1,PARR(IREC,JAB):NEXT JAB
6500     NEXT K
6510 GOTO 6410
6520 NTPT=IREC-1
6530 CLOSE #1:CLOSE #3
6540 CLS:COLOR 11,0
6550 SOUND 500,20:LOCATE 11,21:PRINT "PLEASE SWITCH ON THE LINE PRINTER"
6560 LOCATE 15,21:PRINT "POSITION THE PAPER IN THE PRINTER"
6570 LOCATE 22,14:COLOR 30,0:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
6580 IF INKEY$ < > "" THEN GOTO 6580
```

```
6590 PRO$=INKEY$
6600 IF PRO$ < > " " THEN GOTO 6590
6610 CLS:COLOR 11,0:LOCATE 13,20:PRINT "PRINTING OF PART TYPE DETAILS IN PROGRESS"
6620 LPRINT:LPRINT CHR$(27)CHR$(15);:WIDTH "LPT1:",132
6630 LPRINT TAB(11);"P A R T   T Y P E   D E T A I L S"
6640 LPRINT TAB(11);"┌─────┬───────┬─────┬──────────┬────┬──────────────────────────┬──────
─────────────────────┐"
6650 LPRINT TAB(11);"│ Part │ Batch │ Opn │   Proc.   │ FT │         Machine(s)         │
 Tools required       │"
6660 LPRINT TAB(11);"│ Type │ Size  │ No. │   Time    │    │                            │
                      │"
6670 LPRINT TAB(11);"├─────┼───────┼─────┼──────────┼────┼──────────────────────────┼──────
─────────────────────┤"
6680 FOR LN=1 TO NTPT
6690     IF PARR(LN,1)=0! THEN GOTO 7080
6700     LPRINT TAB(11);"│";TAB(13);LN;
6710     LABS=PARR(LN,2):LPRINT TAB(18);"│";TAB(21);LABS;
6720     LNOP=PARR(LN,3):J11=4:J21=1
6730     FOR MN2=1 TO LNOP
6740     LPRINT TAB(26);"│";TAB(28);MN2;
6750     LPRINT TAB(32);"│";:LPRINT USING "  ###.##   ";PARR(LN,J11);
6760     LPRINT TAB(44);"│";:IFTY=PARR(LN,J11+1):LPRINT TAB(46);IFTY;
6770     LPRINT TAB(49);"│ ";
6780     ICDE=0
6790     FOR JA=J11+2 TO J11+10:IALMAC=PARR(LN,JA)
6800         ICDE=ICDE+1
6810         IF IALMAC=0 THEN GOTO 6860
6820         IF ICDE > 1 THEN GOTO 6840
6830         LPRINT USING "#";IALMAC;:GOTO 6850
6840         LPRINT USING "or#";IALMAC;
6850     NEXT JA
6860     LPRINT TAB(79);"│ ";
6870     ICTA=0
6880     FOR JAB=J11+11 TO J11+20:IALTO=PARR(LN,JAB)
6890         ICTA=ICTA+1
6900         IF IALTO=0 THEN GOTO 6950
6910         IF ICTA > 1 THEN GOTO 6930
6920         LPRINT USING "##";IALTO;:GOTO 6940
6930         LPRINT USING ",##";IALTO;
6940     NEXT JAB
6950     LPRINT TAB(112);"│":J11=J11+21:J21=J21+1
6960     IF MN2=LNOP THEN GOTO 6980
6970     LPRINT TAB(11);"│      │     ";
6980     NEXT MN2
6990 IF LN=NTPT THEN GOTO 7010
7000 LPRINT TAB(11);"├─────┼───────┼─────┼──────────┼────┼──────────────────────────┼──────
─────────────────────┤"
7010 NEXT LN
7020 LPRINT TAB(11);"└─────┴───────┴─────┴──────────┴────┴──────────────────────────┴──────
─────────────────────┘"
```

```
7030 LOCATE 22,14:COLOR 30,0:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
7040 IF INKEY$ < > "" THEN GOTO 7040
7050 PRO$=INKEY$
7060 IF PRO$ < > " " THEN GOTO 7050
7070 FOR I=1 TO 5:LPRINT:NEXT I
7080 CLS:LOCATE 5,2:COLOR 11,0:PRINT "FILE(S) THAT CONTAIN PRIORITY DETAILS : ";:FILES "C:*.PRI"
7090 COLOR 10,0:PRINT:PRINT TAB(2);"ENTER FILE NAME CONTAINING PRIORITY DETAILS : ";:INPUT " ",FI
LE2$:FILE2$=FILE2$+".PRI"
7100 CLS:LOCATE 13,20:COLOR 11,0:PRINT "PRINTING OF PRIORITY DETAILS IN PROGRESS"
7110 IPRIOR=0:FOR JH=1 TO 25:JPRPT(JH)=0:NEXT JH
7120 OPEN FILE2$ FOR INPUT AS #1
7130      IPRIOR=IPRIOR+1:IF EOF(1) THEN GOTO 7160
7140      INPUT #1,IPT,JPRPT(IPRIOR)
7150      GOTO 7130
7160 CLOSE #1
7170 LPRINT TAB(11);"PRIORITY DETAILS"
7180 LPRINT TAB(11);"┌──────┬──────────┐"
7190 LPRINT TAB(11);"│ Part │ Priority │";TAB(41);"*** 1 INDICATES PRIORITY"
7200 LPRINT TAB(11);"│ Type │          │";TAB(41);"    0 INDICATES NO PRIORITY"
7210 LPRINT TAB(11);"├──────┼──────────┤"
7220 FOR JH=1 TO IPRIOR-1
7230     LPRINT TAB(11);"│";TAB(13);JH;TAB(18);"│   ";JPRPT(JH);TAB(29);"│"
7240 NEXT JH
7250 LPRINT TAB(11);"└──────┴──────────┘"
7260 LOCATE 22,14:COLOR 30,0:PRINT "P R E S S   S P A C E   B A R   T O   C O N T I N U E"
7270 IF INKEY$ < > "" THEN GOTO 7270
7280 PRO$=INKEY$
7290 IF PRO$ < > " " THEN GOTO 7280
7300 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN THE SYSTEM LAYOUT DETAILS : ";:FILES "
C:*.LAY"
7310 COLOR 10,0:PRINT:PRINT TAB(2);"ENTER NAME OF FILE THAT CONTAINS THE SYSTEM LAYOUT DETAILS :
";:INPUT " ",FILE4$:FILE4$=FILE4$+".LAY"
7320 IMM=0:FOR IFC=1 TO 9:DLOM(IFC)=0!:DULOM(IFC)=0!:DSTM(IFC)=0!:ITRACK(IFC)=0:NEXT IFC
7330 OPEN FILE4$ FOR INPUT AS #1
7340      IMM=IMM+1:IF EOF(1) THEN GOTO 7370
7350      INPUT #1,IMD,ITRACK(IMM),DLOM(IMM),DULOM(IMM),DSTM(IMM)
7360      GOTO 7340
7370 CLOSE #1
7380 LPRINT CHR$(12);
7390 CLS:COLOR 11,0:LOCATE 13,20:PRINT "PRINTING SYSTEM LAYOUT DETAILS IN PROGRESS"
7400 LPRINT TAB(11);"SYSTEM LAYOUT DETAILS "
7410 LPRINT TAB(11);"┌─────────┬──────┬──────────┬───────────┬──────────────┐"
7420 LPRINT TAB(11);"│ Machine │ Lane │ Loading  │ Unloading │ Central Tool │"
7430 LPRINT TAB(11);"│         │      │ Station  │  Station  │   Storage    │"
7440 LPRINT TAB(11);"├─────────┼──────┼──────────┼───────────┼──────────────┤"
7450 FOR IMF=1 TO IMM-1
7460 LPRINT TAB(11);"│";TAB(16);IMF;TAB(21);"│";TAB(24);ITRACK(IMF);TAB(28);"│";TAB(32);DLOM(IMF)
;TAB(40);"│";TAB(44);DULOM(IMF);TAB(52);"│";TAB(57);DSTM(IMF);TAB(67);"│"
7470 NEXT IMF
7480 LPRINT TAB(11);"└─────────┴──────┴──────────┴───────────┴──────────────┘"
```

```
7490 LOCATE 22,14:COLOR 30,0:PRINT "P R E S S  S P A C E  B A R  T O  C O N T I N U E"
7500 IF INKEY$ < > "" THEN GOTO 7500
7510 PRO$=INKEY$
7520 IF PRO$ < > " " THEN GOTO 7510
7530 CLS:COLOR 11,0:LOCATE 5,2:PRINT "FILE(S) THAT CONTAIN THE TIME AND SPEED DETAILS OF THE  MHS
  : ":FILES "C:*.TIM"
7540 COLOR 10,0:PRINT TAB(2);"ENTER THE NAME OF FILE THAT CONTAINS THE DETAILS OF  M H S  : ";:IN
PUT " ",FILE5$:FILE5$=FILE5$+".TIM"
7550 OPEN FILE5$ FOR INPUT AS #1
7560     INPUT #1,SPAGV,SPUAGV,SPCON,TTSLO,TTULSO
7570 CLOSE #1
7580 CLS:COLOR 11,0:LOCATE 13,20:PRINT "PRINTING OF  M H S DETAILS IN PROGRESS"
7590 FOR IP=1 TO 5:LPRINT:NEXT IP
7600 LPRINT TAB(11);"TIME AND SPEED DETAILS OF THE MATERIAL HANDLING SYSTEMS"
7610 LPRINT TAB(11);"-------------------------------------------------------"
7620 LPRINT
7630 LPRINT TAB(11);"SPEED OF THE LOADING AGV -------------- ";SPAGV;" [mts/min]"
7640 LPRINT TAB(11);"SPEED OF THE UNLOADING AGV ------------ ";SPUAGV;" [mts/min]"
7650 LPRINT TAB(11);"SPEED OF THE TOOL CONVEYOR ------------ ";SPCON;" [mts/min]"
7660 LPRINT TAB(11);"TRAVEL TIME FOR THE LOADING CRANE ----- ";TTSLO;"[mins]"
7670 LPRINT TAB(11);"TRAVEL TIME FOR THE UNLOADING CRANE --- ";TTULSO;"[mins]"
7680 GOTO 350
```

APPENDIX-B

COMPUTER PROGRAM LISTING OF SLAM DISCRETE EVENT PROGRAM

```
C --------------------------------------------------------------------
C
C         SIMULATION MODEL FOR THE FLEXIBLE MANUFACTURING SYSTEM
C
C                      DISCRETE EVENT SEGMENT-1
C
C --------------------------------------------------------------------
C
C -----------------------------------------------------
C     AUTHOR - THIRUVENGADAM RAVI
C              GRADUATE STUDENT
C              DEPARTMENT OF INDUSTRIAL ENGINEERING
C              UNIVERSITY OF WINDSOR
C -----------------------------------------------------
C
C ************
C METACOMMANDS
C ************
$STORAGE:2
$NOTSTRICT
$PAGESIZE:60
$LARGE
$LINESIZE:132
$NOFLOATCALLS
C
C ****************
C THE MAIN PROGRAM
C ****************
      PROGRAM MAIN
C ------------------------COMMON BLOCK----------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM6/NCRANE,NUCRANE,NULO,NUNLO,NAGV,NUAGV,NPAL,NTRAC
C ----------------------------------------------------------------------
      DIMENSION IGD(3),IALM(9),ITOOL(10)
      CHARACTER INIBUF*10,PARBUF(10)*5B,ANS*1,PRINT*2,FULNAME*12,
     &SELRULE*70,FILNAM*12
      LOGICAL QEX
C
C --- FILE # 8 REFERS TO THE LINE PRINTER ---
C
```

```
      OPEN(UNIT=8,FILE='PRN:')
      DO 29 ISA=1,24
         WRITE(*,35)
   29 CONTINUE
      WRITE(*,91)
   91 FORMAT(/,10X,'PLEASE TURN ON THE PRINTER ')
      PAUSE
C
C --- SETTING UP THE PRINTER FOR THE COMPRESSED MODE ---
C
         WRITE(8,14)CHAR(27),CHAR(15)
   14    FORMAT(2A1)
C
         WRITE(8,15)
   15    FORMAT(30X,'**************************************************')
         WRITE(8,16)
   16    FORMAT(30X,'*                                                *')
         WRITE(8,20)
   20    FORMAT(30X,'*   S I M U L A T I O N    O F    A N    F M S   *')
         WRITE(8,16)
         WRITE(8,15)
         WRITE(8,17)
   17    FORMAT(/)
C
C --- SELECTION OF THE REQUIRED RULE ---
C
   25 CONTINUE
      DO 30 ISA=1,24
         WRITE(*,35)
   35    FORMAT(/)
   30 CONTINUE
         WRITE(*,40)
   40    FORMAT(10X,'SCHEDULING RULES THAT ARE AVAILABLE : ')
         WRITE(*,45)
   45    FORMAT(//,15X,'1.  Random selection rule        [ RANDOM ]')
         WRITE(*,50)
   50    FORMAT(/,15X,'2.  Fewest operations remaining  [ FOPR ]')
         WRITE(*,55)
   55    FORMAT(/,15X,'3.  Most operations remaining    [ MOPR ]')
         WRITE(*,60)
   60    FORMAT(/,15X,'4.  Shortest processing time     [ SPT ]')
         WRITE(*,65)
   65    FORMAT(/,15X,'5.  Longest processing time      [ LPT ]')
         WRITE(*,70)
   70    FORMAT(///,10X,'ENTER THE SCHEDULING RULE YOU NEED : ',\)
         READ(*,'(BN,I2)')JRU
         IF(JRU.GT.5.OR.JRU.LT.0) THEN
              WRITE(*,80)CHAR(7)
   80         FORMAT(A1,25X,'WRONG CHOICE - PLEASE ENTER AGAIN ')
              PAUSE
              GO TO 25
         ELSE
         ENDIF
         PRINT='OF'
         WRITE(*,85)
   85    FORMAT(/,10X,'ENTER THE TOTAL NUMBER OF RUNS : ',\)
         READ(*,'(BN,I3)')ITRUN
   89    WRITE(*,86)
   86    FORMAT(/,10X,'DO YOU WANT TO TRACE THE SIMULATION [ Y OR N ] '\)
         WRITE(*,87)
   87    FORMAT('? ',\)
         READ(*,'(A1)')ANS
         IF(ANS.EQ.'Y'.OR.ANS.EQ.'n') THEN
              PRINT='ON'
         ELSEIF(ANS.EQ.'N'.OR.ANS.EQ.'n') THEN
              PRINT='OF'
         ELSE
              WRITE(*,88)CHAR(7)
   88         FORMAT(A1,25X,'WRONG CHOICE - PLEASE TRY AGAIN')
              GO TO 89
         ENDIF
C
C --- WRITING TITLE OF RULE SELECTED ---
C
         IF(JRU.EQ.1) THEN
              SELRULE='Random selection rule  [ RANDOM ]'
         ELSEIF(JRU.EQ.2) THEN
              SELRULE='Fewest operations remaining  [ FOPR ]'
         ELSEIF(JRU.EQ.3) THEN
              SELRULE='Most operations remaining  [ MOPR ]'
         ELSEIF(JRU.EQ.4) THEN
              SELRULE='Shortest processing time  [ SPT ]'
         ELSEIF(JRU.EQ.5) THEN
              SELRULE='Longest processing time  [ LPT ]'
         ELSE
         ENDIF
         WRITE(8,90)SELRULE
   90    FORMAT(/,10X,'Scheduling rule used : ',A70)
C
C --- INITIALISATION BLOCK ---
C
         IPRIOR=0
         IRUN=0
         IREC=0
         KTF=0
         NMREC=0
         IDEOP=10
         DO 105 ISC=1,25
              DO 110 ISD=1,30
                   DUEDAT(ISC,ISD)=0.0
```

```
  110          CONTINUE
  105    CONTINUE
C
C --- ACCESSING DATA FROM THE DATA FILES ---
C
      DO 115 JX=1,24
         WRITE(*,120)
  120    FORMAT(/)
  115 CONTINUE
C
C --- OPENING FILE 1 ---
C
  140 WRITE(*,125)
  125 FORMAT(/,1X,'ENTER NAME OF FILE THAT STORES THE PART DATA : ',\)
      READ(*,'(A)')FULNAME
      INQUIRE(FILE=FULNAME,EXIST=QEX)
      IF(QEX.EQV.(.FALSE.)) THEN
         WRITE(*,130)CHAR(7),CHAR(7),FULNAME
  130    FORMAT(2A1,'FILE ',A12,' DOES NOT EXIST IN HARD DISK - ',\)
         WRITE(*,135)
  135    FORMAT('TRY AGAIN')
         GO TO 140
      ELSE
         OPEN(UNIT=1,FILE=FULNAME,STATUS='OLD',ACCESS='SEQUENTIAL',
     &FORM='FORMATTED')
      ENDIF
C
C --- FILE 1 HAS BEEN OPEN ---
C
      DO 145 IRA=1,25
         DO 150 IRB=1,220
              PARR(IRA,IRB)=0.0
  150    CONTINUE
  145 CONTINUE
  155 IREC=IREC+1
      READ(1,160,END=165)INIBUF,(PARBUF(N),N=1,10)
  160 FORMAT(A10,10(A58))
         READ(INIBUF,170)(IGD(J),J=1,3)
  170    FORMAT(I2,1X,I3,1X,I2,1X)
         PARR(IREC,1)=IGD(1)
         PARR(IREC,2)=IGD(2)
         PARR(IREC,3)=IGD(3)
         DO 175 K=1,10
              J1=4+(K-1)*21
              READ(PARBUF(K),180)PRT,IFT,(IALM(J),J=1,9),(ITOOL(J),J=1,1
     &0)
  180         FORMAT(F6.2,1X,I2,1X,9(I1,1X),10(I2,1X))
              PARR(IREC,J1)=PRT
              PARR(IREC,J1+1)=IFT
              L1=1
              DO 185 LJ=J1+2,J1+10
                   PARR(IREC,LJ)=IALM(L1)
                   L1=L1+1
  185         CONTINUE
              L2=1
              DO 195 LT=J1+11,J1+20
                   PARR(IREC,LT)=ITOOL(L2)
                   L2=L2+1
  195         CONTINUE
  175    CONTINUE
         GO TO 155
  165 NTPT=IREC-1
      CLOSE(1,STATUS='KEEP')
C
C --- FILE 1 IS CLOSED -- ALL PART DETAILS ARE STORED IN THE ARRAY ----
C
C --- OPENING FILE 2 ---
C
  215 WRITE(*,200)
  200 FORMAT(/,1X,'ENTER NAME OF FILE THAT STORES PRIORITY DATA : ',\)
      READ(*,'(A)')FULNAME
      INQUIRE(FILE=FULNAME,EXIST=QEX)
      IF(QEX.EQV.(.FALSE.)) THEN
         WRITE(*,205)CHAR(7),CHAR(7),FULNAME
  205    FORMAT(2A1,'FILE ',A12,' DOES NOT EXIST ON HARD DISK - ',\)
         WRITE(*,210)
  210    FORMAT('TRY AGAIN')
         GO TO 215
      ELSE
         OPEN(UNIT=2,FILE=FULNAME,STATUS='OLD',ACCESS='SEQUENTIAL',
     &FORM='FORMATTED')
      ENDIF
      DO 220 ISE=1,25
         JPRPT(ISE)=0
  220 CONTINUE
  235 IPRIOR=IPRIOR+1
         READ(2,225,END=230)IPF,JPRPT(IPRIOR)
  225    FORMAT(10X,I3,5X,I2)
      GO TO 235
  230 CLOSE(2,STATUS='KEEP')
C
C --- FILE 2 IS CLOSED -- PRIORITY DETAILS FOR ALL PART TYPES ACCESSED ---
C
C --- OPENING FILE 3 ---
C
  295 WRITE(*,280)
  280 FORMAT(/,1X,'ENTER NAME OF FILE THAT STORES LAYOUT DATA   : '\)
      READ(*,'(A)')FULNAME
      INQUIRE(FILE=FULNAME,EXIST=QEX)
      IF(QEX.EQV.(.FALSE.)) THEN
```

```
           WRITE(*,285)CHAR(7),CHAR(7),FULNAME
  285      FORMAT(2A1,'FILE ',A12,' DOES NOT EXIST ON HARD DISK - ',\)
           WRITE(*,290)
  290      FORMAT('TRY AGAIN')
           GO TO 295
        ELSE
           OPEN(UNIT=3,FILE=FULNAME,STATUS='OLD',ACCESS='SEQUENTIAL',
      &FORM='FORMATTED')
        ENDIF
        DO 300 IZZ=1,9
           ITRACK(IZZ)=0
           DLOM(IZZ)=0.0
           DULOM(IZZ)=0.0
           DSTM(IZZ)=0.0
  300   CONTINUE
  315   NMREC=NMREC+1
           READ(3,305,END=310)NUM,ITRACK(NMREC),DLOM(NMREC),DULOM(NMREC),D
      &STM(NMREC)
  305      FORMAT(5X,I3,5X,I2,10X,F7.2,5X,F7.2,5X,F7.2)
        GO TO 315
  310   NMAC=NMREC-1
        CLOSE(3,STATUS='KEEP')
C
C --- FILE 3 CLOSED -- ALL SYSTEM LAYOUT DETAILS HAVE BEEN ACCESSED ---
C
C --- OPENING FILE 4 ---
C
  335   WRITE(*,320)
  320   FORMAT(/,1X,'ENTER NAME OF FILE THAT STORES MHS TIME DATA : ',\)
        READ(*,'(A)')FULNAME
        INQUIRE(FILE=FULNAME,EXIST=QEX)
        IF(QEX.EQV.(.FALSE.)) THEN
           WRITE(*,325)CHAR(7),CHAR(7),FULNAME
  325      FORMAT(2A1,'FILE ',A12,' DOES NOT EXIST ON HARD DISK - '),\)
           WRITE(*,330)
  330      FORMAT('TRY AGAIN')
           GO TO 335
        ELSE
           OPEN(UNIT=4,FILE=FULNAME,STATUS='OLD',ACCESS='SEQUENTIAL',
      &FORM='FORMATTED')
        ENDIF
           READ(4,340)SPAGV,SPUAGV,SPCON,TTSLO,TTULSO
  340      FORMAT(10X,3(F7.3,5X),15X,2(F7.3,5X))
        CLOSE(4,STATUS='KEEP')
C
  365   WRITE(*,345)
  345   FORMAT(//,1X,'DO YOU WANT A PRINT OUT OF THE INPUT DETAILS ',\)
        WRITE(*,350)
  350   FORMAT(' [ 1-Yes, 2-No ] ? ',\)
        READ(*,'(BN,I2)')IAN
      IF(IAN.EQ.1) THEN
         GO TO 355
      ELSEIF(IAN.EQ.2) THEN
         GO TO 615
      ELSE
         WRITE(*,360)CHAR(7)
  360    FORMAT(20X,'WRONG CHOICE - PLEASE TRY AGAIN ')
         GO TO 365
      ENDIF
C
C --- SEGMENT THAT PRINTS OUT THE INPUT DETAILS OF THE SIMULATION MODEL ---
C
  355 CONTINUE
      WRITE(8,95)
   95 FORMAT(/,10X,'Details of all the part types to be ',\)
      WRITE(8,100)
  100 FORMAT('manufactured in the planning period ',//)
      WRITE(8,356)
  356 FORMAT(T11,'P A R T   T Y P E   D E T A I L S ',/)
      WRITE(8,375)
  375 FORMAT(10X,'┌──────┬───────┬─────┬───────────┬────┬──────────',\)
      WRITE(8,380)
  380 FORMAT('───────────────────┬────────────────────────────────┐')
      WRITE(8,385)
  385 FORMAT(10X,'│ Part │ Batch │ Opn │   Proc.   │ FT │       Mac',\)
      WRITE(8,390)
  390 FORMAT('hine(s)            │         Tools required         │')
      WRITE(8,391)
  391 FORMAT(10X,'│ Type │  Size │ No. │   Time    │    │          ',\)
      WRITE(8,392)
  392 FORMAT('                   │                                │')
      WRITE(8,393)
  393 FORMAT(10X,'├──────┼───────┼─────┼───────────┼────┼──────────',\)
      WRITE(8,394)
  394 FORMAT('───────────────────┼────────────────────────────────┤')
      DO 370 LN=1,NTPT
         LABS=PARR(LN,2)
         LNOP=PARR(LN,3)
         J11=4
         WRITE(8,395)LN,LABS
  395    FORMAT(T11,'│',T13,I2,T18,'│',T20,I3,\)
         DO 400 MN2=1,LNOP
              IFTY=PARR(LN,J11+1)
              WRITE(8,405)MN2,PARR(LN,J11),IFTY
  405         FORMAT(T26,'│',T28,I2,T32,'│',2X,F6.2,T44,'│',1X,I2,\)
              WRITE(8,407)
  407         FORMAT(T49,'│  ',\)
              ICDE=0
              DO 410 JA=J11+2,J11+10
                   IALMAC=PARR(LN,JA)
                   ICDE=ICDE+1
                   IF(IALMAC.EQ.0) GO TO 420
                   IF(ICDE.GT.1) GO TO 416
                   WRITE(8,415)IALMAC
415                FORMAT(I1,\)
                   GO TO 410
416                WRITE(8,417)IALMAC
417                FORMAT('or',I1,\)
410           CONTINUE
420           WRITE(8,425)
425           FORMAT(T79,'│ ',\)
              ICTA=0
              DO 430 JAB=J11+11,J11+20
                   IALTO=PARR(LN,JAB)
                   ICTA=ICTA+1
                   IF(IALTO.EQ.0) GO TO 440
                   IF(ICTA.GT.1) GO TO 436
                   WRITE(8,435)IALTO
435                FORMAT(I2,\)
                   GO TO 430
436                WRITE(8,437)IALTO
437                FORMAT(',',I2,\)
430           CONTINUE
440           WRITE(8,445)
445           FORMAT(T112,'│')
              J11=J11+21
              IF(MN2.EQ.LNOP) GO TO 400
              WRITE(8,450)
450           FORMAT(T11,'│      │    ',\)
400      CONTINUE
      IF(LN.EQ.NTPT) GO TO 454
      WRITE(8,451)
451   FORMAT(10X,'├─────┼──────┼────┼─────────┼───┼──────────',\)
      WRITE(8,452)
452   FORMAT('─────────────┼──────────────────────────┤')
370   CONTINUE
454   WRITE(8,457)
457   FORMAT(10X,'└─────┴──────┴────┴─────────┴───┴──────────',\)
      WRITE(8,458)
458   FORMAT('─────────────┴──────────────────────────┘')
      WRITE(8,485)
485   FORMAT(/)
      WRITE(8,456)
456   FORMAT(T11,'PRIORITY DETAILS',/)
      WRITE(8,460)
460   FORMAT(T11,'┌─────┬──────────┐')
      WRITE(8,465)
465   FORMAT(T11,'│ Part │ Priority │',T41,'*** 1 - Indicates priority')
      WRITE(8,470)
470   FORMAT(T11,'│ Type │          │',T41,'    0 - No priority')
```

```
    WRITE(8,461)
461 FORMAT(T11,'├─────┼─────────┤')
    DO 475 IPRI=1,NTPT
       WRITE(8,480)IPRI,JPRPT(IPRI)
480    FORMAT(T11,'│',1X,I2,3X,'│',4X,I2,4X,'│')
475 CONTINUE
    WRITE(8,471)
471 FORMAT(T11,'└─────┴─────────┘')
    WRITE(8,455)CHAR(27),CHAR(12)
455 FORMAT(2A1)
    WRITE(8,535)
535 FORMAT(T11,'SYSTEM LAYOUT DETAILS [All distances are in mts]',/)
    WRITE(8,540)
540 FORMAT(T11,'┌────────┬─────┬────────┬──────────┬',\)
    WRITE(8,541)
541 FORMAT('─────────────┐')
    WRITE(8,545)
545 FORMAT(T11,'│ Machine │ Lane │ Loading │ Unloading │',\)
    WRITE(8,546)
546 FORMAT(' Central Tool │')
    WRITE(8,550)
550 FORMAT(T11,'│         │      │ Station │  Station  │',\)
    WRITE(8,551)
551 FORMAT('   Storage    │ ')
    WRITE(8,552)
552 FORMAT(T11,'├────────┼─────┼────────┼──────────┼',\)
    WRITE(8,553)
553 FORMAT('─────────────┤')
    DO 555 IMA=1,NMAC
       WRITE(8,560)IMA,ITRACK(IMA),DLOM(IMA),DULOM(IMA),DSTM(IMA)
560    FORMAT(T11,'│',T16,I1,T21,'│',T25,I1,T28,'│',T30,F6.2,T38,'│',
   &T41,F6.2,T50,'│',T54,F6.2,T65,'│')
555 CONTINUE
    WRITE(8,556)
556 FORMAT(T11,'└────────┴─────┴────────┴──────────┴',\)
    WRITE(8,557)
557 FORMAT('─────────────┘')
    WRITE(8,570)
570 FORMAT(////,T11,'TRAVEL DETAILS FOR THE VARIOUS MHS USED',/)
    WRITE(8,575)SPAGV
575 FORMAT(10X,'Speed of the loading AGV   = ',F7.3,' [mts/min]')
    WRITE(8,580)SPUAGV
580 FORMAT(10X,'Speed of the unloading AGV = ',F7.3,' [mts/min]')
    WRITE(8,585)SPCON
585 FORMAT(10X,'Speed of the tool conveyor = ',F7.3,' [mts/min]')
    WRITE(8,590)
590 FORMAT(/,10X,'Travel time for the loading crane from the Centr',\)
    WRITE(8,600)TTSLO
600 FORMAT('al Parts Storage to loading station         =',
   &F7.3,' [mins]')
```

```
      WRITE(8,605)
  605 FORMAT(10X,'Travel time for the unloading crane from the unloa',\)
      WRITE(8,610)TTULSO
  610 FORMAT('ding station to the Central Parts Storage =',F7.3
     &,' [mins]')
      WRITE(8,611)
  611 FORMAT(//)
C
C --- RESOURCE QUANTITIES INPUT FROM TERMINAL ---
C
  615 CONTINUE
      DO 616 IXQ=1,24
         WRITE(*,617)
  617    FORMAT(//)
  616 CONTINUE
      WRITE(*,618)
  618 FORMAT(1X,'DETAILS OF THE RESOURCES USED TO BE ENTERED')
      WRITE(*,619)
  619 FORMAT(1X,'-------------------------------------------')
      WRITE(*,620)
  620 FORMAT(/,1X,'ENTER THE NUMBER OF LOADING CRANES AVAILABLE---- ',\)
      READ(*,'(BN,I3)')NCRANE
      WRITE(*,625)
  625 FORMAT(/,1X,'ENTER THE NUMBER OF UNLOADING CRANE AVAILABLE--- ',\)
      READ(*,'(BN,I3)')NUCRANE
      WRITE(*,630)
  630 FORMAT(/,1X,'ENTER THE NUMBER OF LOADERS AVAILABLE----------- ',\)
      READ(*,'(BN,I3)')NULO
      WRITE(*,635)
  635 FORMAT(/,1X,'ENTER THE NUMBER OF UNLOADERS AVAILABLE--------- ',\)
      READ(*,'(BN,I3)')NUNLO
      WRITE(*,640)
  640 FORMAT(/,1X,'ENTER THE NUMBER OF LOADING AGVs AVAILABLE------ ',\)
      READ(*,'(BN,I3)')NAGV
      WRITE(*,645)
  645 FORMAT(/,1X,'ENTER THE NUMBER OF UNLOADING AGVs AVAILABLE---- ',\)
      READ(*,'(BN,I3)')NUAGV
      WRITE(*,650)
  650 FORMAT(/,1X,'ENTER THE NUMBER OF PALLETS AVAILABLE----------- ',\)
      READ(*,'(BN,I3)')NPAL
      WRITE(*,652)
  652 FORMAT(/,1X,'ENTER THE NUMBER OF LANES IN THE SYSTEM--------- ',\)
      READ(*,'(BN,I3)')NTRAC
C
C --- FIXTURE TYPE QUANTITIES INPUT FROM TERMINAL ---
C
      DO 661 IUD=1,24
         WRITE(*,617)
  661 CONTINUE
      DO 662 LNF=1,30
```

```
         NF(LNF)=0
         MNF(LNF)=0
  662 CONTINUE
      WRITE(*,663)
  663 FORMAT(1X,'ENTER THE NUMBER OF FIXTURE TYPES USED -- ',\)
      READ(*,'(BN,I3)')JFTY
      DO 664 KTF=1,JFTY
         WRITE(*,666)KTF
  666    FORMAT(/,1X,'QUANTITY OF FIXTURE TYPE',I3,' AVAILABLE --- ',\)
         READ(*,'(BN,I3)')NUFT
         NF(KTF)=NUFT
         MNF(KTF)=NUFT
  664 CONTINUE
C
      NCRDR=5
      NPRNT=0
      NTAPE=7
      DO 653 IAS=1,24
         WRITE(*,654)
  654    FORMAT(/)
  653 CONTINUE
      WRITE(*,656)SELRULE
  656 FORMAT(1X,'Scheduling rule requested is ',A55)
C
C --- DECIDING THE RIGHT TRANSLATED FILE ---
C
      IF(JRU.EQ.1) THEN
         FILNAM='RAND.TRA'
      ELSEIF(JRU.EQ.2) THEN
         FILNAM='FOPR.TRA'
      ELSEIF(JRU.EQ.3) THEN
         FILNAM='MOPR.TRA'
      ELSEIF(JRU.EQ.4) THEN
         FILNAM='SPT.TRA'
      ELSEIF(JRU.EQ.5) THEN
         FILNAM='LPT.TRA'
      ELSE
      ENDIF
      WRITE(*,657)
  657 FORMAT(//,1X,'FILE NAME OF THE TRANSLATED FILE THAT YOU HAVE ',\)
      WRITE(*,658)FILNAM
  658 FORMAT('TO USE : ',A12)
      PAUSE
      CALL SLAM
      END
C
C --- ALL THE RELEVANT DATA ACCESSED FROM THE DATA FILES ---
C
C
C **********************
```

```
C INITIALIZATION ROUTINE
C **********************
      SUBROUTINE INTLC
C ------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM6/NCRANE,NUCRANE,NULO,NUNLO,NAGV,NUAGV,NPAL,NTRAC
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C
C --- ALTERING THE MACHINES ---
C
      DO 665 IM=1,NMAC
         CALL ALTER(IM,1)
  665 CONTINUE
C
C --- ALTERING THE COMMON RESOURCES ---
C
      CALL ALTER(10,NCRANE)
      CALL ALTER(11,NUCRANE)
      CALL ALTER(12,NULO)
      CALL ALTER(13,NUNLO)
      CALL ALTER(14,NAGV)
      CALL ALTER(15,NUAGV)
      CALL ALTER(16,NPAL)
C
C --- ALTERING THE NUMBER OF FIXTURE TYPES ---
C
      DO 670 IFX=1,30
         NUFF=MNF(IFX)
         CALL ALTER(IFX+16,NUFF)
  670 CONTINUE
C
C --- ALTERING THE NUMBER OF TRACKS ---
C
      DO 675 ITR=1,NTRAC
         CALL ALTER(ITR+46,1)
         CALL ALTER(ITR+51,1)
  675 CONTINUE
C
      IRUN=IRUN+1
      IF(IRUN.GT.1) GO TO 715
C
C --- PRINTING OUT THE RESOURCE DETAILS ---
C
      WRITE(8,565)CHAR(27),CHAR(12)
  565 FORMAT(2A1)
      WRITE(8,679)
  679 FORMAT(/,10X,'RESOURCE DETAILS FOR THE  F M S ')
      WRITE(8,680)NNRSC(10)
  680 FORMAT(/,10X,'Number of loading cranes   =  ',I1)
      WRITE(8,685)NNRSC(11)
  685 FORMAT(10X,'Number of unloading cranes =  ',I1)
      WRITE(8,690)NNRSC(12)
  690 FORMAT(10X,'Number of human loaders    =  ',I1)
      WRITE(8,695)NNRSC(13)
  695 FORMAT(10X,'Number of human unloaders  =  ',I1)
      WRITE(8,700)NNRSC(14)
  700 FORMAT(10X,'Number of loading AGVs     =  ',I1)
      WRITE(8,705)NNRSC(15)
  705 FORMAT(10X,'Number of unloading AGVs   =  ',I1)
      WRITE(8,710)NNRSC(16)
  710 FORMAT(10X,'Number of pallets used     = ',I2)
      WRITE(8,490)
  490 FORMAT(T11,'FIXTURE TYPE DETAILS',/)
      WRITE(8,495)
  495 FORMAT(T11,'┌─────────┬──────────┐')
      WRITE(8,500)
  500 FORMAT(T11,'│ Fixture │ Quantity │')
      WRITE(8,505)
  505 FORMAT(T11,'│  Type   │          │')
      WRITE(8,510)
  510 FORMAT(T11,'├─────────┼──────────┤')
      DO 515 IFX=1,30
         IF(MNF(IFX).EQ.0) GO TO 525
         WRITE(8,520)IFX,MNF(IFX)
  520    FORMAT(T11,'│',3X,I2,4X,'│',4X,I2,4X,'│')
  515 CONTINUE
  525 WRITE(8,526)
  526 FORMAT(T11,'└─────────┴──────────┘')
      WRITE(8,530)
  530 FORMAT(////)
C
C --- FILING ALL THE PARTS IN THE CENTRAL STORAGE ---
C
  715 CONTINUE
      IPCO=0
      JCT1=21
      JCT2=17
      DO 720 JNT=1,NTPT
         JBS=PARR(JNT,2)
```

```
        DO 725 JK=1,JBS
             DO 730 JA=1,100
                  ATRIB(JA)=0.0
  730        CONTINUE
             IPCO=IPCO+1
             ATRIB(1)=JPRPT(JNT)
             ATRIB(2)=IPCO
             ATRIB(3)=JNT
             ATRIB(4)=PARR(JNT,3)
             ATRIB(5)=1.
             JAN=ATRIB(5)
             JCT3=JAN*JCT1-JCT2
             ATRIB(6)=PARR(JNT,JCT3)
             ATRIB(7)=PARR(JNT,JCT3+1)
             DO 735 JAT=22,40
                  ATRIB(JAT)=PARR(JNT,JCT3+2)
                  JCT3=JCT3+1
  735        CONTINUE
             IF(JRU.EQ.1) THEN
                  PUTRAND=DRAND(1)
                  ATRIB(9)=PUTRAND
             ELSE
             ENDIF
             CALL FILEM(20,ATRIB)
             JCT3=0
  725   CONTINUE
  720 CONTINUE
      ITOPA=NNQ(20)
      IF(IRUN.GT.1) GO TO 740
      WRITE(8,745)
  745 FORMAT(/,10X,'Total number of parts in the Central Parts '\)
      WRITE(8,750)ITOPA
  750 FORMAT('Storage at the start of the simulation = ',I4,/)
      WRITE(8,751)
  751 FORMAT(///)
C
C --- INDICATION ON SCREEN ---
C
  740 CONTINUE
      DO 742 ISF=1,24
         WRITE(*,744)
  744    FORMAT(/)
  742 CONTINUE
      WRITE(*,755)
  755 FORMAT(5X,'S I M U L A T I O N   I N   P R O G R E S S')
      WRITE(*,760)
  760 FORMAT(5X,'*********************************************'/)
      WRITE(*,770)IRUN
  770 FORMAT(6X,'RUN NUMBER ',I3,\)
      IF(PRINT.EQ.'OF') GO TO 771
      WRITE(8,773)CHAR(27),CHAR(12)
 773  FORMAT(2A1)
      WRITE(8,772)IRUN
 772  FORMAT(/,10X,'TRACE OF RUN ',I2,' OF THE SIMULATION',/)
C
C --- INITIALIZATION OF OTHER VARIABLES IN THE PROGRAM ---
C
 771  CONTINUE
      DO 775 KK=1,100
         XX(KK)=0.0
 775  CONTINUE
      DO 780 IH=1,9
         IDQU(IH)=0
         ITCS(IH)=0
         DO 785 KLM=1,100
              KTM(IH,KLM)=0
 785      CONTINUE
 780  CONTINUE
      DO 790 ISG=1,25
         JFOUT(ISG)=0
         NPARTY(ISG)=0
 790  CONTINUE
      ISTCO=0
      ISRFOL=0
      JPCRA=0
      MPCOUNT=0
      IENT=0
      IPAS=0
      JCMAQ=0
      LCOUNT=0
      MAQ=0
      FINTIM=99999.00
      RETURN
      END
C
C ********************************************
C ROUTINE FOR SELECTING THE APPROPIATE EVENT
C ********************************************
      SUBROUTINE EVENT(I)
C --------------------------COMMON BLOCK----------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,
     &DLOM(9),DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITAK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C -----------------------------------------------------------------------
      CHARACTER PRINT*2
C
C --- ROUTINES THAT ARE EXTERNAL TO THIS SEGMENT ---
C
      EXTERNAL PARTIN,RELAGVL,TOOLCAPT,FREEBUF,RELFIXT,REENTER,
     &REATRIB,PLOTFACT
      GO TO (1,2,3,4,5,6,7,8),I
    1 CALL PARTIN
         RETURN
    2 CALL RELAGVL
         RETURN
    3 CALL TOOLCAPT
         RETURN
    4 CALL FREEBUF
         RETURN
    5 CALL RELFIXT
         RETURN
    6 CALL REENTER
         RETURN
    7 CALL REATRIB
         RETURN
    8 CALL PLOTFACT
         RETURN
      END
C
C ***********************************************      ***********
C ROUTINE FOR THE ALLOCATION AT THE VARIOUS NODES  --  ALLOCATE(I)
C ***********************************************      ***********
      SUBROUTINE ALLOC(ICODE,IFLAG)
C -------------------------COMMON BLOCK ---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C -----------------------------------------------------------------------
      CHARACTER PRINT*2
      IFLAG=0
      GO TO (805,810,815),ICODE
C
C --- ALLOCATION OF LOADING CRANE, LOADING AGV, PALLET, LOADER &
C                                                 FIXTURE TYPE ----
```

```
C
  805 IFIX=ATRIB(7)
      IF(NNRSC(10).GE.1.AND.NNRSC(12).GE.1.AND.NNRSC(14).GE.1.AND.NNRSC(
     &16).GE.1.AND.NNRSC(IFIX+16).GE.1) THEN
         CALL SEIZE(10,1)
         CALL SEIZE(12,1)
         CALL SEIZE(14,1)
         CALL SEIZE(16,1)
         CALL SEIZE(IFIX+16,1)
         IF(PRINT.EQ.'OF') GO TO 806
         IPZ1=ATRIB(2)
         WRITE(8,807)TNOW,IPZ1
  807    FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' SEIZES THE LOADI',\)
         WRITE(8,808)
  808    FORMAT('NG CRANE, LOADING AGV, LOADER, PALLET AND FIXTURE')
  806    IFLAG=-1
         RETURN
      ELSE
         RETURN
      ENDIF
C
C --- ALLOCATION OF THE EXACT LOADING TRACK ---
C
  810 KTRA=ATRIB(10)
      IF(NRUSE(KTRA+46).EQ.0) THEN
         CALL SEIZE(KTRA+46,1)
         IFLAG=-1
         RETURN
      ELSE
         RETURN
      ENDIF

C
C --- ALLOCATION OF UNLOADING AGV, UNLOADER AND UNLOADING CRANE ---
C
  815 LTRA=ATRIB(10)
      IF(NNRSC(11).GE.1.AND.NNRSC(13).GE.1.AND.NNRSC(15).GE.1.AND.NRUSE(
     &LTRA+51).EQ.0) THEN
         CALL SEIZE(11,1)
         CALL SEIZE(13,1)
         CALL SEIZE(15,1)
         CALL SEIZE(LTRA+51,1)
         IF(PRINT.EQ.'OF') GO TO 816
         IPZ2=ATRIB(2)
         WRITE(8,817)TNOW,IPZ2
  817    FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' SEIZES THE UNLOA',\)
         WRITE(8,818)
  818    FORMAT('DING AGV, UNLOADER, AND UNLOADING CRANE')
  816    IFLAG=-1
         RETURN
      ELSE
         RETURN
      ENDIF
      END
C
C **********************
C USER FUNCTION ROUTINES
C **********************
      FUNCTION USERF(I)
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
      GO TO (855,860,865,870,875,880,885),I
C
C --- TRANSPORT TIME FROM CENTRAL PARTS STORAGE TO LOADING STATION ---
C
  855 IF(JPCRA.EQ.0) THEN
         USERF=TTSLO
         JPCRA=1
      ELSEIF(JPCRA.EQ.1) THEN
         USERF=2*TTSLO
      ELSE
      ENDIF
      IF(PRINT.EQ.'OF') RETURN
      IPZ3=ATRIB(2)
      WRITE(8,857)TNOW,IPZ3
  857 FORMAT(10X,'TIME = ',F10.4,' PART * ',I3,' TRANSPORTED FROM TH',\)
      WRITE(8,858)USERF
  858 FORMAT('E  C P S  TO THE LOADING STATION - TRAVEL TIME = ',F6.3)
      RETURN
C
C --- TRANSPORT TIME FROM LOADING STATION TO THE EXACT MACHINE ---
C
  860 MARO=ATRIB(8)
      TTLOM=DLOM(MARO)/SPAGV
      USERF=TTLOM
      IF(PRINT.EQ.'OF') RETURN
      IPZ4=ATRIB(2)
      MTRAC=ATRIB(10)
      WRITE(8,861)TNOW,IPZ4
  861 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' TRANSPORTED FROM LO',\)
      WRITE(8,862)MARO,USERF
  862 FORMAT('ADING STATION TO MACHINE ',I1,' - TRAVEL TIME = ',F6.3)
      WRITE(8,863)IPZ4
  863 FORMAT(20X,'LOADING AGV CARRYING PART # ',I3,' MOVING ON ',\)
      WRITE(8,864)MTRAC
  864 FORMAT('LOADING TRACK ',I2)
      RETURN
C
C --- TRAVEL TIME FOR EMPTY CART FROM MACHINE TO THE LOADING STATION ---
C
  865 MACHO=ATRIB(8)
      TTMLO=DLOM(MACHO)/SPAGV
      USERF=TTMLO
      IF(PRINT.EQ.'OF') RETURN
      IPZ5=ATRIB(2)
      MMTRAC=ATRIB(10)
      WRITE(8,866)TNOW,IPZ5
  866 FORMAT(10X,'TIME = ',F10.3,' PART # ',I3,' RELEASES CART AT MA',\)
      WRITE(8,867)MACHO,USERF
  867 FORMAT('CHINE ',I1,' - AGV TRAVEL TIME TO  C P S = ',F6.3)
      WRITE(8,868)MMTRAC
  868 FORMAT(20X,'EMPTY LOADING AGV MOVING ON LOADING TRACK ',I2)
      RETURN
C
C --- TRAVEL TIME FROM TOOL STORAGE TO MACHINE ---
C
  870 MACGO=ATRIB(8)
      IF(ITCS(MACGO).EQ.0) THEN
         USERF=0.0
      ELSE
         USERF=DSTM(MACGO)/SPCON
      ENDIF
      ITCS(MACGO)=0
      IF(PRINT.EQ.'OF') RETURN
      IPZ6=ATRIB(2)
      WRITE(8,871)TNOW,IPZ6
  871 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' WAITS FOR TOOLS FR',\)
      WRITE(8,872)MACGO,USERF
  872 FORMAT('OM THE TOOL CRIB AT MACHINE ',I1,' - WAITING TIME = ',
     &F6.3)
      RETURN
C
C --- TRANSPORT TIME FROM THE MACHINE TO THE UNLOADING STATION ---
C
  875 MAFO=ATRIB(8)
      TTMAUL=DULOM(MAFO)/SPUAGV
      USERF=2*TTMAUL
      IF(PRINT.EQ.'OF') RETURN
      IPZ7=ATRIB(2)
      MNTRAC=ATRIB(10)
      WRITE(8,876)TNOW,IPZ7
  876 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' TRANSPORTED FROM MA',\)
      WRITE(8,877)MAFO,USERF
  877 FORMAT('CHINE ',I1,' TO UNLOADING STATION - TRAVEL TIME = ',F6.3)
      WRITE(8,878)IPZ7
  878 FORMAT(20X,'UNLOADING AGV CARRYING PART # ',I3,' MOVING ON ',\)
      WRITE(8,879)MNTRAC
  879 FORMAT('UNLOADING TRACK ',I2)
      RETURN
C
C --- TRAVEL TIME FOR UNLOADING CRANE FROM PARTS STORAGE TO UNLOADING
C     STATION -----
C
  880 USERF=TTULSO
      IF(PRINT.EQ.'OF') RETURN
      IPZ8=ATRIB(2)
      WRITE(8,881)TNOW,IPZ8
  881 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' WAITS FOR THE UNLOA',\)
      WRITE(8,882)USERF
  882 FORMAT('DING CRANE AT THE UNLOADING STATION - WAITING TIME = ',
     &F6.3)
      RETURN
C
C --- TRAVEL TIME FOR UNLOADING CRANE TO THE CENTRAL PARTS STORAGE ---
C
  885 USERF=TTULSO
      IF(PRINT.EQ.'OF')RETURN
      IPZ9=ATRIB(2)
      WRITE(8,886)TNOW,IPZ9
  886 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' TRANSPORTED FROM TH',\)
      WRITE(8,887)USERF
  887 FORMAT('E UNLOADING STATION TO THE  C P S - TRAVEL TIME = ',F6.3)
      RETURN
      END


C -----------------------------------------------------------------------
C
C           SIMULATION OF FLEXIBLE MANUFACTURING SYSTEM
C
C                  DISCRETE EVENT SEGMENT - 2
C
C -----------------------------------------------------------------------
C
C #############
C META COMMANDS
C #############
C
```

```
$STORAGE:2
$NOTSTRICT
$PAGESIZE:60
$LARGE
$LINESIZE:132
$NOFLOATCALLS
C
C *****************************************************     *******
C ROUTINE TO SELECT AND INPUT THE PARTS INTO THE SYSTEM  -- EVENT 1
C *****************************************************     *******
      SUBROUTINE PARTIN
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
C
C --- BLOCK THAT DECIDES WHERE THE ENTITY CAME FROM ---
C
      IF(II.EQ.1) THEN
         IPAS=0
         IENT=0
         NIQ=NMAC
 1005    CALL SELPART
         IF(JCMAQ.GT.NNQ(20)) GO TO 1010
         IF(IENT.LT.NIQ) GO TO 1005
 1010    XX(1)=1
         XX(4)=1
         IF(IPAS.EQ.0) XX(1)=2
         XX(2)=IPAS
         RETURN
      ELSEIF(II.EQ.2) THEN
         IF(JRU.EQ.1) THEN
              PURANG=DRAND(1)
              ATRIB(9)=PURANG
         ELSE
         ENDIF
         CALL FILEM(20,ATRIB)
         IF(NNQ(20).GE.1.AND.NNQ(22).EQ.0) THEN
              IDQ=0
```

```
               DO 1015 ICE=1,NMAC
                    IF(IDQU(ICE).GT.0) GO TO 1015
                         IDQ=IDQ+1
  1015         CONTINUE
               IPAS=0
               IENT=0
               NIQ=IDQ
  1020         CALL SELPART
               IF(JCMAQ.GT.NNQ(20)) GO TO 1025
               IF(IENT.LT.NIQ) GO TO 1020
  1025         XX(1)=1
               XX(2)=IPAS
               IF(IPAS.EQ.0) XX(1)=2
               XX(4)=1
               RETURN
          ELSE
               XX(1)=2
               XX(4)=1
               RETURN
          ENDIF
       ELSEIF(II.EQ.3) THEN
          IENT=0
          IPAS=0
          NIQ=IDQ
  1030    CALL SELPART
          IF(JCMAQ.GT.NNQ(20)) GO TO 1035
          IF(IENT.LT.NIQ) GO TO 1030
  1035    XX(2)=IPAS
          XX(1)=1
          XX(4)=2
          IF(IPAS.EQ.0) XX(1)=2
          RETURN
      ELSE
               XX(1)=2
               XX(4)=1
          RETURN
      ENDIF
      END
C
C --- ROUTINE TO SELECT PART AND SEND IT INTO THE SYSTEM ---
C
      SUBROUTINE SELPART
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
```

```
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
      IENT=IENT+1
      JCMAQ=0
      IF(NNQ(20).EQ.0) RETURN
      DO 1040 KPB=1,NMAC
         IDLEQ(KPB)=0
 1040 CONTINUE
      KPA=0
      DO 1045 KP=1,NMAC
         IF(IDQU(KP).GT.0) GO TO 1045
              KPA=KPA+1
              IDLEQ(KPA)=KP
 1045 CONTINUE
      ITER=0
 1050 NST=0
      ITER=ITER+1
 1055 CONTINUE
      DO 1060 NRANK=NST+1,NNQ(20)
         CALL COPY(NRANK,20,ATRIB)
         IF(ITER.EQ.2) GO TO 1065
         IF(ATRIB(1).EQ.0) GO TO 1060
 1065    IOPT=ATRIB(5)
         IRES=ATRIB(7)
         IF(NF(IRES).EQ.0) GO TO 1060
         NF(IRES)=NF(IRES)-1
         GO TO 1070
 1060 CONTINUE
      IF(ITER.LT.2) GO TO 1050
      RETURN
 1070 IQA=0
      MAQ=0
      DO 1075 IAB=1,9
         IALTQ(IAB)=0
         MACAQ(IAB)=0
 1075 CONTINUE
      DO 1080 IAL=22,30
         IF(ATRIB(IAL).EQ.0.0) GO TO 1085
              IQA=IQA+1
              IALTQ(IQA)=ATRIB(IAL)
 1080 CONTINUE
 1085 CONTINUE
      DO 1090 IQAC=1,NIQ
         DO 1095 IAT=1,IQA
              IF(IALTQ(IAT).NE.IDLEQ(IQAC)) GO TO 1095
```

```
                   MAQ=MAQ+1
                   MACAQ(MAQ)=IDLEQ(IQAC)
 1095    CONTINUE
 1090 CONTINUE
      IF(MAQ.EQ.0) THEN
         JCMAQ=JCMAQ+1
         NF(IRES)=NF(IRES)+1
         NST=NRANK
         GO TO 1055
      ELSEIF(MAQ.EQ.1) THEN
         ICOM=MACAQ(1)
         IDQU(ICOM)=1
      ELSE
         CALL SELRAND
      ENDIF
      CALL RMOVE(NRANK,20,ATRIB)
      ATRIB(8)=ICOM
      ATRIB(IDEOP+IOPT)=TNOW
      ATRIB(10)=ITRACK(ICOM)
      IF(PRINT.EQ.'OF') GO TO 1093
      IPZ10=ATRIB(2)
      WRITE(8,1091)TNOW,IPZ10
 1091 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' HAS BEEN SELECTED T',\)
      WRITE(8,1092)ICOM,ITRACK(ICOM)
 1092 FORMAT('O BE PROCESSED IN MACHINE ',I1,' ON TRACK ',I1,' FOR ',\)
      WRITE(8,1094)IOPT
 1094 FORMAT('OPERATION # ',I2)
 1093 IPAS=IPAS+1
      CALL FILEM(21,ATRIB)
      RETURN
      END
C
C --- ROUTINE FOR SELECTING THE QUEUE AT RANDOM ---
C
      SUBROUTINE SELRAND
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ---------------------------------------------------------------------
      DIMENSION A(9)
      CHARACTER PRINT*2
```

```
C ----------------------------------------------------------------------
      DO 1100 IN=1,NMAC
         A(IN)=0.0
 1100 CONTINUE
      DO 1110 IQ=1,MAQ
         A(IQ)=DLOM(MACAQ(IQ))+DULOM(MACAQ(IQ))
         IF(IQ.GT.1) GO TO 1115
         GMIN=A(IQ)
         MANO=MACAQ(IQ)
         GO TO 1110
 1115    IF(A(IQ).LT.GMIN) THEN
              GMIN=A(IQ)
              MANO=MACAQ(IQ)
         ELSE
         ENDIF
 1110 CONTINUE
      ICOM=MANO
      IDQU(ICOM)=1
      RETURN
      END
C
C ******************************    *******
C ROUTINE FOR RELEASING THE CART -- EVENT 2
C ******************************    *******
      SUBROUTINE RELAGVL
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
C
C --- RELEASING THE CART ---
C
      KUTRA=ATRIB(10)
      CALL FREE(KUTRA+46,1)
      CALL FREE(14,1)
      IF(PRINT.EQ.'OF') RETURN
      WRITE(8,1116)TNOW
 1116 FORMAT(10X,'TIME = ',F10.4,' LOADING AGV IS RELEASED AT THE ',\)
      WRITE(8,1117)
 1117 FORMAT('LOADING STATION')
      RETURN
      END
C
C *********************************************************      *******
C ROUTINE FOR CAPTURING THE TOOLS AS PART ENTERS THE SYSTEM  --  EVENT 3
C *********************************************************      *******
      SUBROUTINE TOOLCAPT
C -------------------------COMMON BLOCK--------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ---------------------------------------------------------------------
      CHARACTER PRINT*2
C ---------------------------------------------------------------------
      MACH=ATRIB(8)
      ITCS(MACH)=0
      DO 1120 JT=31,40
         IF(ATRIB(JT).EQ.0.0) GO TO 1125
         ITONE=ATRIB(JT)
         DO 1130 JTA=1,100
              IF(KTM(MACH,JTA).EQ.ITONE) GO TO 1120
 1130    CONTINUE
         ITCS(MACH)=ITCS(MACH)+1
 1120 CONTINUE
 1125 CONTINUE
      IF(PRINT.EQ.'OF') RETURN
      IPW=ATRIB(2)
      WRITE(8,1126)IPW,ITCS(MACH),MACH
 1126 FORMAT(20X,'PART # ',I3,' NEEDS ',I2,' TOOLS AT MACHINE ',I1)
      RETURN
      END
C
C ************************************      *******
C ROUTINE FOR FREEING THE INPUT BUFFER  --  EVENT 4
C ************************************      *******
      SUBROUTINE FREEBUF
C -------------------------COMMON BLOCK--------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
      TENFB=TNOW
      IFRQ=ATRIB(8)
      IDQU(IFRQ)=0
      IDQ=0
      DO 1135 IQQ=1,NMAC
         IF(IDQU(IQQ).GT.0) GO TO 1135
             IDQ=IDQ+1
 1135 CONTINUE
      CALL OPEN(2)
      IF(PRINT.EQ.'OF') RETURN
      IPW2=ATRIB(2)
      WRITE(8,1131)TNOW,IPW2
 1131 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' LEAVING THE INPUT ',\)
      WRITE(8,1132)IFRQ
 1132 FORMAT('BUFFER OF MACHINE ',I1)
      RETURN
      END
C
C
C *************************************      *******
C ROUTINE TO RELEASE MULTIPLE RESOURCES  --  EVENT 5
C *************************************      *******
      SUBROUTINE RELFIXT
C -------------------------COMMON BLOCK---------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
C
C --- RELEASING THE FIXTURE ---
```

```
C
      JFIX=ATRIB(7)
      KWTRA=ATRIB(10)
      NF(JFIX)=NF(JFIX)+1
      CALL FREE(KWTRA+51,1)
      CALL FREE(JFIX+16,1)
      IF(PRINT.EQ.'OF') RETURN
      IPW1=ATRIB(2)
      WRITE(8,1136)TNOW,IPW1
 1136 FORMAT(10X,'TIME = ',F10.4,' PART * ',I3,' RELEASES FIXTURE ')
      RETURN
      END
C
C **********************************************************     *******
C ROUTINE TO DECIDE WHETHER THE PART SHOULD REENTER OR NOT  --  EVENT 6
C **********************************************************     *******
      SUBROUTINE REENTER
C ------------------------COMMON BLOCK-------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFDL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C -------------------------------------------------------------------
      CHARACTER PRINT*2
      EXTERNAL FINSTAT
C -------------------------------------------------------------------
      JCOPT=ATRIB(5)
      ATRIB(4)=ATRIB(4)-1.
      ATRIB(IDEOP+JCOPT)=TNOW-ATRIB(IDEOP+JCOPT)
      ATRIB(5)=ATRIB(5)+1.
      IF(ATRIB(4).EQ.0.0) GO TO 1140
      DO 1145 IRE=22,40
          ATRIB(IRE)=0.0
 1145 CONTINUE
C
C --- SEGMENT FOR RE-ENTERING THE SYSTEM ---
C
      JRT=ATRIB(3)
      JCR1=21
      JCR2=17
      JAR=ATRIB(5)
      JCR3=JAR*JCR1-JCR2
      ATRIB(6)=PARR(JRT,JCR3)
      ATRIB(7)=PARR(JRT,JCR3+1)
      DO 1150 JART=22,40
           ATRIB(JART)=PARR(JRT,JCR3+2)
           JCR3=JCR3+1
 1150 CONTINUE
      II=2
      RETURN
C
C --- SEGMENT FOR COLLECTING DATA AS PART GOES OUT OF THE SYSTEM ---
C
 1140 II=4
      MPCOUNT=MPCOUNT+1
      IPTN=ATRIB(3)
      IPW3=ATRIB(2)
      IF(PRINT.EQ.'OF') GO TO 1144
      WRITE(8,1141)TNOW,IPW3
 1141 FORMAT(10X,'TIME = ',F10.4,' PART # ',I3,' BELONGING TO PART ',\)
      WRITE(8,1142)IPTN
 1142 FORMAT('TYPE ',I2,' HAS FINISHED ALL ITS OPERATIONS ')
      WRITE(8,1143)MPCOUNT
 1143 FORMAT(10X,'NUMBER OF PARTS THAT HAVE LEFT THE SYSTEM = ',I3)
 1144 NPARTY(IPTN)=NPARTY(IPTN)+1
      IF(NPARTY(IPTN).EQ.PARR(IPTN,2)) THEN
         DUEDAT(IPTN,IRUN)=TNOW
         LCOUNT=LCOUNT+1
         JFOUT(IPTN)=LCOUNT
      ELSE
      ENDIF
      IF(TNOW.EQ.TENFB) THEN
         GO TO 1155
      ELSE
         IDQ=0
         IF(NNQ(20).GE.1.AND.NNQ(22).EQ.0) THEN
              DO 1160 IX=1,NMAC
                   IF(IDQU(IX).GT.0) GO TO 1160
                   IDQ=IDQ+1
 1160         CONTINUE
              CALL OPEN(2)
         ELSE
         ENDIF
      ENDIF
 1155 CONTINUE
      IF(MPCOUNT.EQ.ITOPA) THEN
         FINTIM=TNOW
         CALL FINSTAT
         RETURN
      ELSE
         RETURN
      ENDIF
      END
```

```
C
C **********************************************      *******
C ROUTINE FOR READJUSTMENT OF THE ATRIBUTE ARRAY  --  EVENT 7
C **********************************************      *******
      SUBROUTINE REATRIB
C ------------------------COMMON BLOCK----------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
      DO 1165 IA=1,100
         ATRIB(IA)=0.0
 1165 CONTINUE
      RETURN
      END
C
C *****************************   *******
C ROUTINE FOR PLOTTING THE DATA   EVENT 8
C *****************************   *******
      SUBROUTINE PLOTFACT
C ------------------------COMMON BLOCK----------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARO,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM7/ICRANE,IUCRANE,IULO,IUNLO,IAGV,IUAGV,IPAL
C ----------------------------------------------------------------------
      CHARACTER PRINT*2
C ----------------------------------------------------------------------
      IF(TNOW.GE.FINTIM) THEN
         XX(9)=2
         RETURN
      ELSE
C        IF(IRUN.GT.1) GO TO 1199
```

```
C        WRITE(*,1200)TNOW,XX(20),NNQ(20)
C1200    FORMAT(/,1X,F10.4,5X,F3.0,5X,I3)
         XX(9)=1
         RETURN
         ENDIF
      END


C
C *************
C META COMMANDS
C *************
C
$STORAGE:2
$NOTSTRICT
$PAGESIZE:60
$LARGE
$LINESIZE:132
$NOFLOATCALLS
C
C ********************************************************
C ROUTINE TO COLLECT THE DATA AND PRINT IT OUT AT THE END
C ********************************************************
      SUBROUTINE FINSTAT
C -------------------------COMMON BLOCK--------------------------------
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
     &NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,
     &TNOW,XX(100)
      COMMON/UCOM1/ICOM,IDEOP,IDQ,IENT,IPAS,IPCO,ISRFOL,ISTCO,ITOPA,
     &JRU,JCMAQ,MAQ,NIQ,NMAC,NRANK,NTPT,IALTQ(9),IDLEQ(9),IDQU(9)
      COMMON/UCOM2/JPCRA,PRINT,SPAGV,SPCON,SPUAGV,TTSLO,TTULSO,DLOM(9),
     &DULOM(9),DSTM(9),NF(30),PARR(25,220),MNF(30),ITRACK(9)
      COMMON/UCOM3/ITCS(9),MACAQ(9),NAGVL(9),NPAGVL(5),KTM(9,100)
      COMMON/UCOM4/FINTIM,IRUN,ITRUN,MARQ,MPCOUNT,TENFB,JPRPT(25)
      COMMON/UCOM5/LCOUNT,NPARTY(25),DUEDAT(25,30),JFOUT(25)
      COMMON/UCOM6/NCRANE,NUCRANE,NULO,NUNLO,NAGV,NUAGV,NPAL,NTRAC
C --------------------------------------------------------------------
      DIMENSION SPAN(30),UTIMAC(9,30),UTIRES(7,30),UTIFIX(30,30),
     &SCHRA(30),JDF(25,30),SUMAC(9),AVUMAC(9),SDIF(9),VARMAC(9),
     &KRES(7)
      CHARACTER PRINT*2
C --------------------------------------------------------------------
C
C --- INITIALISATION OF ALL ARRAYS IN THE FIRST RUN ---
C
      IF(IRUN.GT.1) GO TO 3005
      DO 3010 IDCARX=1,9
         DO 3015 IDCARY=1,30
              UTIMAC(IDCARX,IDCARY)=0.0
```

```
             SPAN(IDCARY)=0.0
             SCHRA(IDCARY)=0.0
 3015    CONTINUE
 3010 CONTINUE
        DO 3020 IDCARY=1,30
             DO 3025 IDCARX=1,25
                 JDF(IDCARX,IDCARY)=0
 3025        CONTINUE
         DO 3030 IDCARX=1,7
             UTIRES(IDCARX,IDCARY)=0.0
 3030    CONTINUE
         DO 3035 IDCARX=1,30
             UTIFIX(IDCARX,IDCARY)=0.0
 3035    CONTINUE
 3020 CONTINUE
C
C --- ASSIGNING NUMBER OF RESOURCES TO ARRAY ---
C
      KRES(1)=NCRANE
      KRES(2)=NUCRANE
      KRES(3)=NULO
      KRES(4)=NUNLO
      KRES(5)=NAGV
      KRES(6)=NUAGV
      KRES(7)=NPAL
C
C --- ACTUAL COLLECTION OF DATA FOR EACH RUN IN THIS SEGMENT ---
C
 3005 SPAN(IRUN)=TNOW
      SCHRA(IRUN)=SPAN(IRUN)/ITOPA
C
C --- MACHINE UTILIZATION DATA COLLECTION ---
C
      DO 3040 IDCMAC=1,NMAC
         UTIMAC(IDCMAC,IRUN)=TTAVG(IDCMAC)
 3040 CONTINUE
C
C --- RESOURCE UTILIZATION DATA COLLECTION ---
C
      DO 3045 IDCRES=1,7
         UTIRES(IDCRES,IRUN)=RRAVG(IDCRES+9)/KRES(IDCRES)
 3045 CONTINUE
C
C --- FIXTURE UTILIZATION DATA COLLECTION ---
C
      DO 3050 IDCFIX=1,30
         IF(MNF(IDCFIX).EQ.0) GO TO 3055
         UTIFIX(IDCFIX,IRUN)=RRAVG(IDCFIX+16)/MNF(IDCFIX)
 3050 CONTINUE
C
```

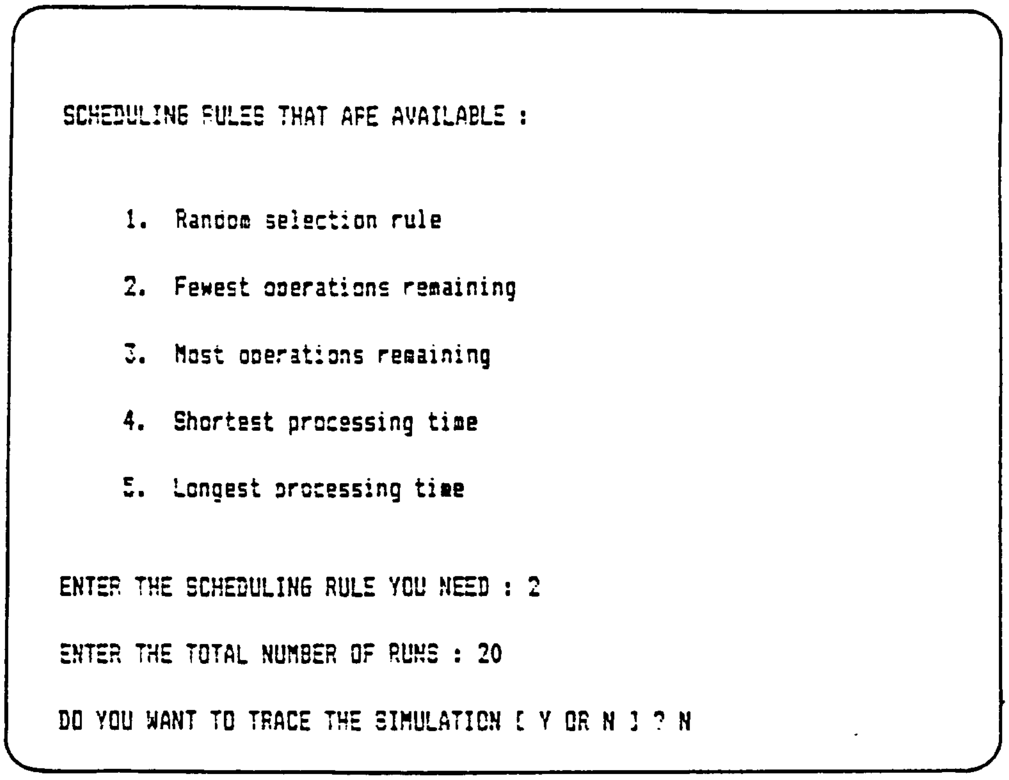
```
C --- DUE DATE COLLECTION ---
C
 3055 CONTINUE
      DO 3060 J6=1,NTPT
         JDF(J6,IRUN)=JFOUT(J6)
 3060 CONTINUE
       IF(IRUN.LT.ITRUN) RETURN
C
C --- AVERAGES FOR CALCULATED IN THIS SEGMENT ---
C
      WRITE(*,3065)
 3065 FORMAT(/,10X,'PLEASE WAIT - PRINTING ON LINE PRINTER'\)
      WRITE(8,3070)
 3070 FORMAT(//)
      WRITE(8,3071)CHAR(27),CHAR(12)
 3071 FORMAT(2A1)
      WRITE(8,3075)
 3075 FORMAT(10X,'******************************************')
      WRITE(8,3080)ITRUN
 3080 FORMAT(11X,'PRINTOUT AFTER ',I2,'  RUNS OF THE SIMULATION ')
      WRITE(8,3075)
      SUM1=0.0
      SUM2=0.0
      WRITE(8,3085)
 3085 FORMAT(//,T11,3X,'Run #     Makespan (mins)      Avg.Time / Part')
      WRITE(8,3090)
 3090 FORMAT(T11,3X,'---------------------------------------------')
      DO 3095 JRUN=1,ITRUN
         WRITE(8,3100)JRUN,SPAN(JRUN),SCHRA(JRUN)
 3100    FORMAT(13X,I3,6X.F11.4,15X,F7.3)
         SUM1=SUM1+SPAN(JRUN)
         SUM2=SUM2+SCHRA(JRUN)
 3095 CONTINUE
      AVSPAN=SUM1/ITRUN
      AVSCHRA=SUM2/ITRUN
      IF(ITRUN.EQ.1) GO TO 3105
      SQDIF1=0.0
      SQDIF2=0.0
      DO 3110 JRUN=1,ITRUN
         SQDIF1=SQDIF1+(ABS(SPAN(JRUN)-AVSPAN))**2
         SQDIF2=SQDIF2+(ABS(SCHRA(JRUN)-AVSCHRA))**2
 3110 CONTINUE
      VARSPAN=SQDIF1/(ITRUN-1)
      VARSCHRA=SQDIF2/(ITRUN-1)
 3105 WRITE(8,3090)
      WRITE(8,3115)AVSPAN,AVSCHRA
 3115 FORMAT(14X,'Avg ',4X,F11.4,15X,F7.3)
      IF(ITRUN.EQ.1) GO TO 3120
      WRITE(8,3125)VARSPAN,VARSCHRA
 3125 FORMAT(14X,'Var ',4X,F11.4,15X,F7.3)
```

```
C
C --- MACHINE UTILIZATIONS ---
C
 3120 CONTINUE
      WRITE(8,3122)
 3122 FORMAT(//)
      WRITE(8,3071)CHAR(27),CHAR(12)
      DO 3130 JINT=1,9
         SUMAC(JINT)=0.0
         AVUMAC(JINT)=0.0
         SDIF(JINT)=0.0
         VARMAC(JINT)=0.0
 3130 CONTINUE
      WRITE(8,3135)
 3135 FORMAT(///,10X,'UTILIZATION OF MACHINES :')
      WRITE(8,3140)
 3140 FORMAT(/,T11,3X,'Run # ',\)
      DO 3145 JM=1,NMAC
         WRITE(8,3150)JM
 3150    FORMAT('   Mach',I1,' ',\)
 3145 CONTINUE
      WRITE(8,3155)
 3155 FORMAT(/,T11,3X,'------',\)
      DO 3160 JM=1,NMAC
         WRITE(8,3165)
 3165    FORMAT('---------',\)
 3160 CONTINUE
      DO 3166 KR=1,ITRUN
         WRITE(8,3170)KR
 3170    FORMAT(/,13X,I3,2X,\)
         DO 3175 JM=1,NMAC
              WRITE(8,3180)UTIMAC(JM,KR)
 3180         FORMAT('  ',F7.5,\)
 3175    CONTINUE
 3166 CONTINUE
      DO 3185 KR=1,ITRUN
         DO 3190 JM=1,NMAC
              SUMAC(JM)=SUMAC(JM)+UTIMAC(JM,KR)
 3190    CONTINUE
 3185 CONTINUE
      DO 3195 JM=1,NMAC
         AVUMAC(JM)=SUMAC(JM)/ITRUN
 3195 CONTINUE
      IF(ITRUN.EQ.1) GO TO 3250
      DO 3200 KR=1,ITRUN
         DO 3205 JM=1,NMAC
              SDIF(JM)=SDIF(JM)+(ABS(AVUMAC(JM)-UTIMAC(JM,KR)))**2
 3205    CONTINUE
 3200 CONTINUE
      DO 3210 JM=1,NMAC
```

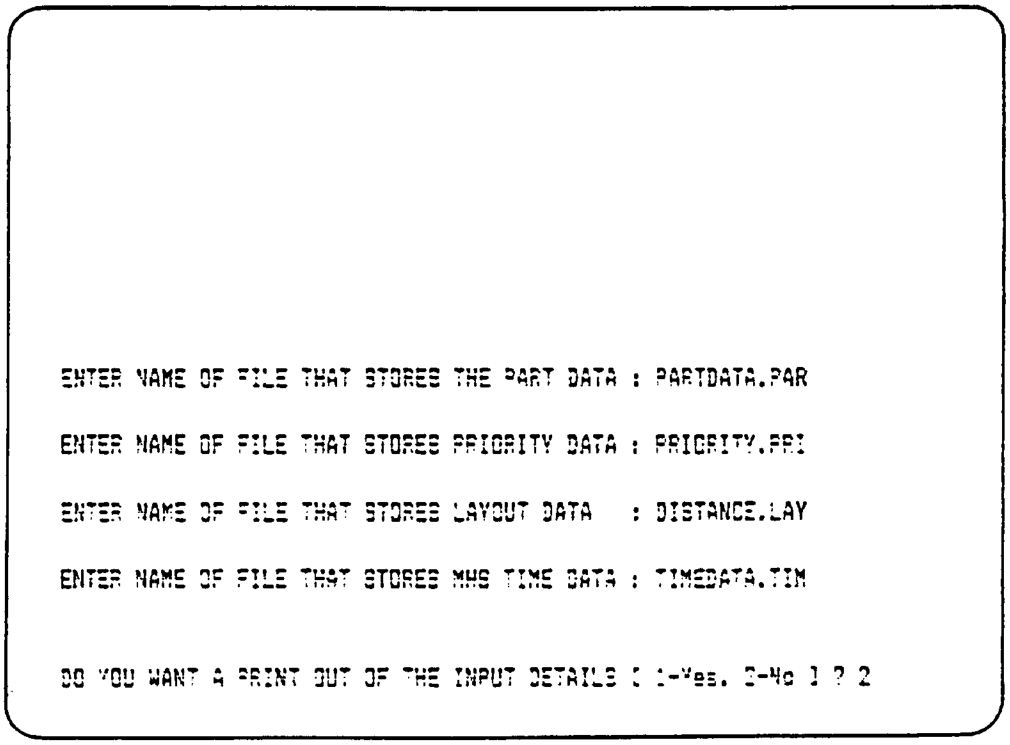
```
        VARMAC(JM)=SDIF(JM)/(ITRUN-1)
 3210 CONTINUE
 3250 WRITE(8,3155)
      DO 3215 JM=1,NMAC
         WRITE(8,3165)
 3215 CONTINUE
      WRITE(8,3220)
 3220 FORMAT(/,12X,'Average ',\)
      DO 3225 JM=1,NMAC
         WRITE(8,3230)AVUMAC(JM)
 3230    FORMAT(F7.5,'  ',\)
 3225 CONTINUE
      IF(ITRUN.EQ.1) GO TO 3255
      WRITE(8,3235)
 3235 FORMAT(/,12X,'Variance',\)
      DO 3240 JM=1,NMAC
         WRITE(8,3245)VARMAC(JM)
 3245    FORMAT(F7.5,'  ',\)
 3240 CONTINUE
C
C --- RESOURCE UTILIZATIONS ---
C
 3255 CONTINUE
      WRITE(8,3257)
 3257 FORMAT(///)
      WRITE(8,3256)CHAR(27),CHAR(12)
 3256 FORMAT(2A1)
      DO 3260 JINT=1,9
         SUMAC(JINT)=0.0
         AVUMAC(JINT)=0.0
         SDIF(JINT)=0.0
         VARMAC(JINT)=0.0
 3260 CONTINUE
      WRITE(8,3265)
 3265 FORMAT(///,10X,'UTILIZATION OF THE RESOURCES :')
      WRITE(8,3270)
 3270 FORMAT(/,11X,'Run #      Crane   Uncrane    Load      Uload ',\)
      WRITE(8,3275)
 3275 FORMAT('    AGV-L    AGV-UL    Pallet',\)
      WRITE(8,3280)
 3280 FORMAT(/,11X,'---------------------------------------------',\)
      WRITE(8,3285)
 3285 FORMAT('--------------------------------',\)
      DO 3290 KR=1,ITRUN
         WRITE(8,3295)KR
 3295    FORMAT(/,10X,I5,'   ',\)
         DO 3300 LR=1,7
              WRITE(8,3305)UTIRES(LR,KR)
 3305         FORMAT('   ',F7.5,\)
 3300    CONTINUE
 3290 CONTINUE
      WRITE(8,3280)
      WRITE(8,3285)
      DO 3310 KR=1,ITRUN
         DO 3315 JM=1,7
              SUMAC(JM)=SUMAC(JM)+UTIRES(JM,KR)
 3315    CONTINUE
 3310 CONTINUE
      DO 3320 JM=1,7
         AVUMAC(JM)=SUMAC(JM)/ITRUN
 3320 CONTINUE
      IF(ITRUN.EQ.1) GO TO 3360
      DO 3325 KR=1,ITRUN
         DO 3330 JM=1,NMAC
              SDIF(JM)=SDIF(JM)+(ABS(AVUMAC(JM)-UTIRES(JM,KR)))**2
 3330    CONTINUE
 3325 CONTINUE
      DO 3335 JM=1,7
         VARMAC(JM)=SDIF(JM)/(ITRUN-1)
 3335 CONTINUE
 3360 WRITE(8,3220)
      DO 3340 JM=1,7
         WRITE(8,3345)AVUMAC(JM)
 3345    FORMAT(' ',F7.5,'  ',\)
 3340 CONTINUE
      IF(ITRUN.EQ.1) GO TO 3365
      WRITE(8,3235)
      DO 3350 JM=1,7
         WRITE(8,3355)VARMAC(JM)
 3355    FORMAT(' ',F7.5,'  ',\)
 3350 CONTINUE
C
C --- FIXTURE UTILIZATIONS ---
C
 3365 CONTINUE
      WRITE(8,3370)
 3370 FORMAT(///,10X,'UTILIZATION OF FIXTURES : ',/)
      DO 3375 JF=1,30
         IF(MNF(JF).EQ.0) GO TO 3380
         SUMF=0.0
         FDIF=0.0
         DO 3385 JR=1,ITRUN
              SUMF=SUMF+UTIFIX(JF,JR)
 3385    CONTINUE
         AVFIX=SUMF/ITRUN
         IF(ITRUN.EQ.1) GO TO 3400
         DO 3390 JR=1,ITRUN
              FDIF=FDIF+(ABS(AVFIX-UTIFIX(JF,JR)))**2
 3390    CONTINUE
         VARFIX=FDIF/(ITRUN-1)
 3400    WRITE(8,3395)JF,ITRUN,AVFIX,VARFIX
 3395    FORMAT(12X,'Average utilization of fixture # ',I2,'  after ',
    &I2,'  runs  = ',F7.5,'  and the variance  =  ',F7.5)
 3375 CONTINUE
C
C  --- DUE DATES FOR THE PART TYPES ---
C
 3380 CONTINUE
      WRITE(8,3381)
 3381 FORMAT(//)
      WRITE(8,3071)CHAR(27),CHAR(12)
      WRITE(8,3405)
 3405 FORMAT(///,11X,'DUE DATES FOR THE PART TYPES :',/)
      WRITE(8,3410)
 3410 FORMAT(13X,'Run # ',\)
      DO 3415 IPM=1,NTPT
         WRITE(8,3420)IPM
 3420    FORMAT('     Type',I2,'    ',\)
 3415 CONTINUE
      WRITE(8,3425)
 3425 FORMAT(/,13X,'-------',\)
      DO 3430 IPM=1,NTPT
         WRITE(8,3435)
 3435    FORMAT('---------------',\)
 3430 CONTINUE
      DO 3440 JR=1,ITRUN
         WRITE(8,3445)JR
 3445    FORMAT(/,13X,I3,'   ',\)
         DO 3450 IPM=1,NTPT
              WRITE(8,3455)DUEDAT(IPM,JR),JDF(IPM,JR)
 3455         FORMAT(F10.3,' - ',I2,\)
 3450    CONTINUE
 3440 CONTINUE
      WRITE(8,3475)
 3475 FORMAT(//,10X,'==============================================',\)
      WRITE(8,3480)
 3480 FORMAT('=======================================================')
      WRITE(8,3491)
 3491 FORMAT(//)
      WRITE(8,3492)CHAR(27),CHAR(12)
 3492 FORMAT(2A1)
      DO 3490 JUTE=1,24
         WRITE(*,3495)CHAR(7)
 3495    FORMAT(A1,/)
 3490 CONTINUE
      WRITE(*,3465)
 3465 FORMAT(1X,'ALL THE RUNS YOU REQUESTED ARE OVER ')
      PAUSE 'PRESS  CTRL-BREAK  KEY TO STOP THE PROGRAM'
      RETURN
      END
```

APPENDIX-C

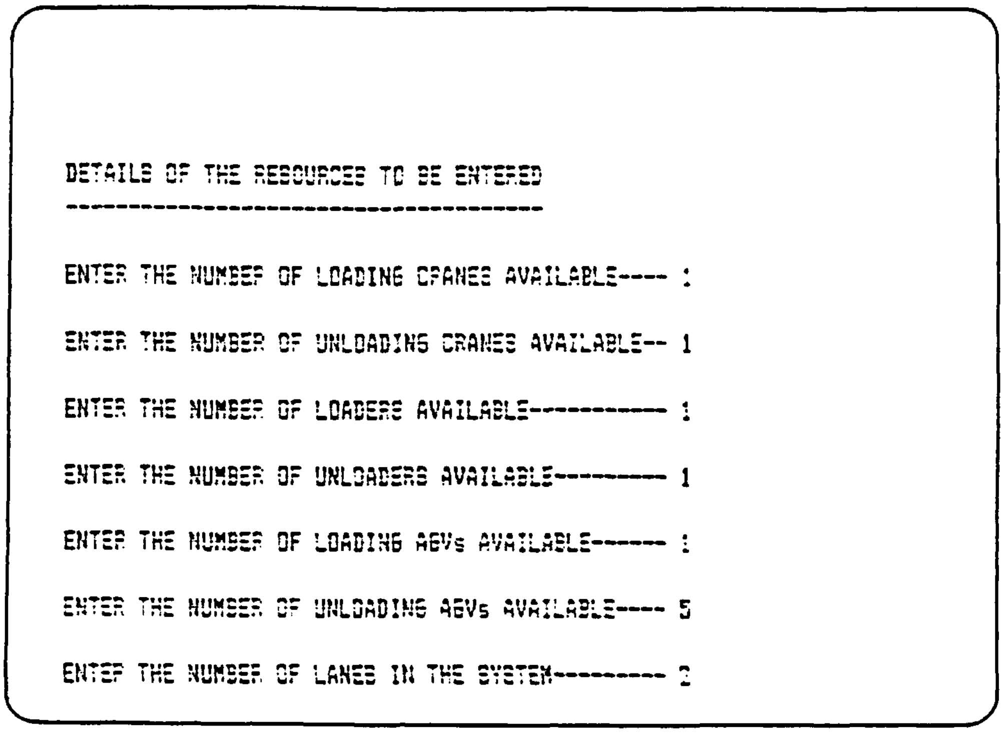
SAMPLE SCREEN DISPLAYS OF THE SIMULATION MODEL



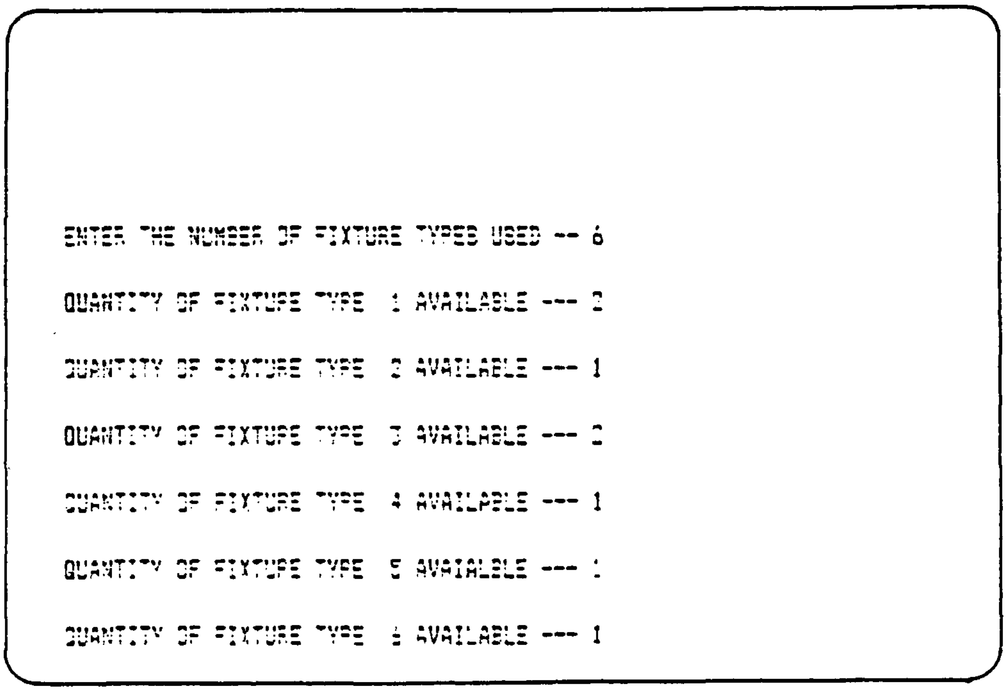
```
SCHEDULING RULES THAT ARE AVAILABLE :


     1.  Random selection rule

     2.  Fewest operations remaining

     3.  Most operations remaining

     4.  Shortest processing time

     5.  Longest processing time


ENTER THE SCHEDULING RULE YOU NEED : 2

ENTER THE TOTAL NUMBER OF RUNS : 20

DO YOU WANT TO TRACE THE SIMULATION [ Y OR N ] ? N
```



```
ENTER NAME OF FILE THAT STORES THE PART DATA : PARTDATA.PAR

ENTER NAME OF FILE THAT STORES PRIORITY DATA : PRIORITY.PRI

ENTER NAME OF FILE THAT STORES LAYOUT DATA   : DISTANCE.LAY

ENTER NAME OF FILE THAT STORES MHS TIME DATA : TIMEDATA.TIM


DO YOU WANT A PRINT OUT OF THE INPUT DETAILS [ 1-Yes, 2-No ] ? 2
```



```
DETAILS OF THE RESOURCES TO BE ENTERED
--------------------------------------

ENTER THE NUMBER OF LOADING CRANES AVAILABLE---- 1

ENTER THE NUMBER OF UNLOADING CRANES AVAILABLE-- 1

ENTER THE NUMBER OF LOADERS AVAILABLE---------- 1

ENTER THE NUMBER OF UNLOADERS AVAILABLE--------- 1

ENTER THE NUMBER OF LOADING AGVs AVAILABLE------ 1

ENTER THE NUMBER OF UNLOADING AGVs AVAILABLE---- 5

ENTER THE NUMBER OF LANES IN THE SYSTEM--------- 2
```



```
ENTER THE NUMBER OF FIXTURE TYPES USED -- 6

QUANTITY OF FIXTURE TYPE  1 AVAILABLE --- 2

QUANTITY OF FIXTURE TYPE  2 AVAILABLE --- 1

QUANTITY OF FIXTURE TYPE  3 AVAILABLE --- 2

QUANTITY OF FIXTURE TYPE  4 AVAILABLE --- 1

QUANTITY OF FIXTURE TYPE  5 AVAIALBLE --- 1

QUANTITY OF FIXTURE TYPE  6 AVAILABLE --- 1
```

APPENDIX-D

COMPUTER PROGRAM LISTING OF SLAM NETWORK PROGRAM

```
;-------------------------------------------------------------------
;
;      N E T W O R K   P O R T I O N   O F   T H E   M O D E L
;
;            Fewest Operations Remaining [ FOPR ] rule used
;
;-------------------------------------------------------------------
;-------------------------------------------------------------------
;
;   Physical limitations of the simulation model :
;
;        (i)  Maximum number of machines---------9
;       (ii)  Maximum number of parts that could be in the central parts
;             storage at any point in time-------250
;      (iii)  Maximum number of fixture types----30
;       (iv)  Maximum number of tool types-------100
;        (v)  Maximum number of part types-------25
;
;   For each part type
;        (i)  Maximum number of alternate machines available for an
;             operation  -  9 (But this could be less depending on
;                              the actual number of machines used in the
;                              system)
;       (ii)  Maximum number of operations  -  10
;      (iii)  Maximum number of tools per operation  -  10
;
;   Resource details - [ See resource blocks ]
;        (i)  Machines are treated as resources (Resource 1 to 9)
;       (ii)  The MHS are treated as resources (Resource 10 to 16)
;      (iii)  Fixture types are treated as resources (Resource 17 to 46)
;       (iv)  Tracks on which the machines are located are treated as
;                                             resources
;                             Loading tracks   - Resource 47 to 51
;                             Unloading tracks - Resource 52 to 56
;
;
;             The initial quantites of these resources are made zero as
;   can be seen in the resource block.  The initialisation routine
;   alters the value of these resources as the program is run.
;
;-------------------------------------------------------------------
;-----------------
;CONTROL STATEMENTS
;-----------------
GEN,RAVI THIRUVENGADAM,THESIS,06/10/86,30,N,N,,N,NO;
LIMITS,26,40,250;
PRIORITY/20,LVF(4);
SEEDS,43521343(1);
TIMST,XX(11),MACH1 UTIL;
```

```
TIMST,XX(12),MACH2 UTIL;
TIMST,XX(13),MACH3 UTIL;
TIMST,XX(14),MACH4 UTIL;
TIMST,XX(15),MACH5 UTIL;
TIMST,XX(16),MACH6 UTIL;
TIMST,XX(17),MACH7 UTIL;
TIMST,XX(18),MACH8 UTIL;
TIMST,XX(19),MACH9 UTIL;
;--------------------------------
;NETWORK STATEMENTS i.e; THE NODES
;--------------------------------
NETWORK;
;---------------
;RESOURCE BLOCKS
;---------------
;
;MACHINES AS RESOURCES
      RESOURCE/MACH1(0),1;
      RESOURCE/MACH2(0),2;
      RESOURCE/MACH3(0),3;
      RESOURCE/MACH4(0),4;
      RESOURCE/MACH5(0),5;
      RESOURCE/MACH6(0),6;
      RESOURCE/MACH7(0),7;
      RESOURCE/MACH8(0),8;
      RESOURCE/MACH9(0),9;
;
;COMMON RESOURCES
      RESOURCE/CRANE(0),22;
      RESOURCE/UNCRANE(0),23;
      RESOURCE/LOADER(0),22;
      RESOURCE/UNLOADER(0),23;
      RESOURCE/AGVL(0),22;
      RESOURCE/UAGV(0),23;
      RESOURCE/PALLET(0),22;
;
;FIXTURES AS RESOURCES
      RESOURCE/FIXT1(0),22;
      RESOURCE/FIXT2(0),22;
      RESOURCE/FIXT3(0),22;
      RESOURCE/FIXT4(0),22;
      RESOURCE/FIXT5(0),22;
      RESOURCE/FIXT6(0),22;
      RESOURCE/FIXT7(0),22;
      RESOURCE/FIXT8(0),22;
      RESOURCE/FIXT9(0),22;
      RESOURCE/FIXT10(0),22;
      RESOURCE/FIXT11(0),22;
      RESOURCE/FIXT12(0),22;
      RESOURCE/FIXT13(0),22;
      RESOURCE/FIXT14(0),22;
      RESOURCE/FIXT15(0),22;
      RESOURCE/FIXT16(0),22;
      RESOURCE/FIXT17(0),22;
      RESOURCE/FIXT18(0),22;
      RESOURCE/FIXT19(0),22;
      RESOURCE/FIXT20(0),22;
      RESOURCE/FIXT21(0),22;
      RESOURCE/FIXT22(0),22;
      RESOURCE/FIXT23(0),22;
      RESOURCE/FIXT24(0),22;
      RESOURCE/FIXT25(0),22;
      RESOURCE/FIXT26(0),22;
      RESOURCE/FIXT27(0),22;
      RESOURCE/FIXT28(0),22;
      RESOURCE/FIXT29(0),22;
      RESOURCE/FIXT30(0),22;
;
;TRACKS AS RESOURCES
      RESOURCE/TRAC1(0),25;
      RESOURCE/TRAC2(0),25;
      RESOURCE/TRAC3(0),25;
      RESOURCE/TRAC4(0),25;
      RESOURCE/TRAC5(0),25;
      RESOURCE/TRAC6(0),23;
      RESOURCE/TRAC7(0),23;
      RESOURCE/TRAC8(0),23;
      RESOURCE/TRAC9(0),23;
      RESOURCE/TRAC10(0),23;
;
;-----------
;GATE BLOCKS
;-----------
      GATE/PILEUP,CLOSE,21;
      GATE/INTOP,CLOSE,24;
;
;-----------------------------------------------------
;NODE DETAILS OF THE ENTIRE FLEXIBLE MANUFACTURING SYSTEM
;-----------------------------------------------------
;
;   ***********************************************************
;   *                      SEGMENT - 1                        *
;   *                      -----------                        *
;   *  CREATING THE ENTIRE RANGE OF PARTS AND STORING THEM IN *
;   * THE CENTRAL PARTS STORAGE AT THE START OF THE SIMULATION *
;   ***********************************************************
;
CRE1  CREATE;
ASS1  ASSIGN,II=1,XX(3)=0;
EVE1  EVENT,1,1;                        Initial part input into system
```

```
      ACT,,XX(1).EQ.1,OPE1;
      ACT,,XX(1).EQ.2,DON;
      ACT;
ACCU  AWAIT(21),PILEUP;
ASS2  ASSIGN,XX(3)=XX(3)+1;
      ACT,,XX(3).EQ.XX(2),CLO1;
      ACT,,XX(3).LT.XX(2),MULT;
CLO1  CLOSE,PILEUP;
ASS3  ASSIGN,XX(3)=0;
MULT  AWAIT(22),ALLOC(1);              Capturing multiple resources
;
;   **************************************************************
;   *                       SEGMENT - 2                          *
;   *                       -----------                          *
;   * THE ACTUAL ENTRY OF PARTS INTO THE SYSTEM AT THE LOADING  *
;   *                         STATION                            *
;   **************************************************************
;
      ACT,USERF(1);                    Travel time to loading station
      GOON;
      ACT,UNFRM(2.0,3.0,1);            Loading time at loading station
FCRA  FREE,CRANE/1;
FLOA  FREE,LOADER/1,1;
AW25  AWAIT(25),ALLOC(2);              Waiting for track to clear
AS20  ASSIGN,XX(20)=XX(20)+1;          Increment total number in FMS by 1
      ACT,USERF(2);                    Travel time to the machine
SEPA  GOON,2;
      ACT,,,EVE3;
      ACT,USERF(3),,EVE2;              Travel time for AGV tol loading stn.
EVE2  EVENT,2;                         Release loading AGV, clearing track
      TERM;
EVE3  EVENT,3;                         Capturing the tools
;
;   **************************************************************
;   *                       SEGMENT - 3                          *
;   *                       -----------                          *
;   *  ROUTING OF THE PART TO THE APPROPRIATE MACHINE FOR THE    *
;   *                  OPERATION TO BE DONE                      *
;   **************************************************************
;
GOO1  GOON,1;
      ACT,,ATRIB(8).EQ.1,BUF1;
      ACT,,ATRIB(8).EQ.2,BUF2;
      ACT,,ATRIB(8).EQ.3,BUF3;
      ACT,,ATRIB(8).EQ.4,BUF4;
      ACT,,ATRIB(8).EQ.5,BUF5;
      ACT,,ATRIB(8).EQ.6,BUF6;
      ACT,,ATRIB(8).EQ.7,BUF7;
      ACT,,ATRIB(8).EQ.8,BUF8;
      ACT,,ATRIB(8).EQ.9,BUF9;
```

```
BUF1  AWAIT(1/1),MACH1/1;
      ACT,,,EVE4;
BUF2  AWAIT(2/1),MACH2/1;
      ACT,,,EVE4;
BUF3  AWAIT(3/1),MACH3/1;
      ACT,,,EVE4;
BUF4  AWAIT(4/1),MACH4/1;
      ACT,,,EVE4;
BUF5  AWAIT(5/1),MACH5/1;
      ACT,,,EVE4;
BUF6  AWAIT(6/1),MACH6/1;
      ACT,,,EVE4;
BUF7  AWAIT(7/1),MACH7/1;
      ACT,,,EVE4;
BUF8  AWAIT(8/1),MACH8/1;
      ACT,,,EVE4;
BUF9  AWAIT(9/1),MACH9/1;
      ACT,,,EVE4;
EVE4  EVENT,4;                          Freeing the input buffer
      ACT,USERF(4);                     Waiting time for tools from tool crib
      GOON,1;
      ACT,,ATRIB(8).EQ.1,GON1;
      ACT,,ATRIB(8).EQ.2,GON2;
      ACT,,ATRIB(8).EQ.3,GON3;
      ACT,,ATRIB(8).EQ.4,GON4;
      ACT,,ATRIB(8).EQ.5,GON5;
      ACT,,ATRIB(8).EQ.6,GON6;
      ACT,,ATRIB(8).EQ.7,GON7;
      ACT,,ATRIB(8).EQ.8,GON8;
      ACT,,ATRIB(8).EQ.9,GON9;
;
GON1  ASSIGN,XX(11)=1;
      ACT/1,ATRIB(6);                   Processing time on M/C 1
      ASSIGN,XX(11)=0;
      ACT,,,GOGO;
;
GON2  ASSIGN,XX(12)=1;
      ACT/2,ATRIB(6);                   Processing time on M/C 2
      ASSIGN,XX(12)=0;
      ACT,,,GOGO;
;
GON3  ASSIGN,XX(13)=1;
      ACT/3,ATRIB(6);                   Processing time on M/C 3
      ASSIGN,XX(13)=0;
      ACT,,,GOGO;
;
GON4  ASSIGN,XX(14)=1;
      ACT/4,ATRIB(6);                   Processing time on M/C 4
      ASSIGN,XX(14)=0;
      ACT,,,GOGO;
```

```
;
GON5  ASSIGN,XX(15)=1;
      ACT/5,ATRIB(6);                     Processing time on M/C 5
      ASSIGN,XX(15)=0;
      ACT,,,GOGO;
;
GON6  ASSIGN,XX(16)=1;
      ACT/6,ATRIB(6);                     Processing time on M/C 6
      ASSIGN,XX(16)=0;
      ACT,,,GOGO;
;
GON7  ASSIGN,XX(17)=1;
      ACT/7,ATRIB(6);                     Processing time on M/C 7
      ASSIGN,XX(17)=0;
      ACT,,,GOGO;
;
GON8  ASSIGN,XX(18)=1;
      ACT/8,ATRIB(6);                     Processing time on M/C 8
      ASSIGN,XX(18)=0;
      ACT,,,GOGO;
;
GON9  ASSIGN,XX(19)=1;
      ACT/9,ATRIB(6);                     Processing time on M/C 9
      ASSIGN,XX(19)=0;
      ACT,,,GOGO;
;
GOGO  GOON;
FREM  FREE,ATRIB(8)/1,1;                  Release the M/C after processing
;
      ACT,,ATRIB(8).EQ.1,QU10;
      ACT,,ATRIB(8).EQ.2,QU11;
      ACT,,ATRIB(8).EQ.3,QU12;
      ACT,,ATRIB(8).EQ.4,QU13;
      ACT,,ATRIB(8).EQ.5,QU14;
      ACT,,ATRIB(8).EQ.6,QU15;
      ACT,,ATRIB(8).EQ.7,QU16;
      ACT,,ATRIB(8).EQ.8,QU17;
      ACT,,ATRIB(8).EQ.9,QU18;
QU10  QUEUE(10),,1,BLOCK,SEL;
QU11  QUEUE(11),,1,BLOCK,SEL;
QU12  QUEUE(12),,1,BLOCK,SEL;
QU13  QUEUE(13),,1,BLOCK,SEL;
QU14  QUEUE(14),,1,BLOCK,SEL;
QU15  QUEUE(15),,1,BLOCK,SEL;
QU16  QUEUE(16),,1,BLOCK,SEL;
QU17  QUEUE(17),,1,BLOCK,SEL;
QU18  QUEUE(18),,1,BLOCK,SEL;
;
SEL   SELECT,LWF,,,QU10,QU11,QU12,QU13,QU14,QU15,QU16,QU17,QU18;
      ACT;
```

```
;   *************************************************************
;   *                      SEGMENT - 4                          *
;   *                      -----------                          *
;   *    TRANSPORTATION OF THE PART BACK TO THE UNLOADING       *
;   *      STATION FROM THE OUPUT BUFFER OF THE MACHINE         *
;   *************************************************************

AW23  AWAIT(23/1),ALLOC(3),BLOCK;
      ACT,USERF(5);                       Travel time to unloading station
AS26  ASSIGN,XX(20)=XX(20)-1;             Decrease total number in FMS by 1
      ACT,USERF(6);                       Travel time for unloading crane
      GOON;
      ACT,UNFRM(1.0,2.0,1);               Unloading time unloading station
FPAL  FREE,PALLET/1;                      Release pallet
FULO  FREE,UNLOADER/1;                    Release unloader
FRUG  FREE,UAGV/1;                        Release unloading AGV
EVE5  EVENT,5;                            Release fixture type, clearing track
      ACT,USERF(7);                       Travel time to parts storage
FUCR  FREE,UNCRANE/1;
EVE6  EVENT,6,1;                          Reentry of part decided
      ACT,,II.EQ.2,EVE1;
      ACT,,II.EQ.4;
      TERM;
;
;   *************************************************************
;   *                      SEGMENT - 5                          *
;   *                      -----------                          *
;   *                                                           *
;   *   INTERMEDIATE ENTRY OF PART INTO THE SYSTEM AS SOON AS   *
;   *       THE INPUT BUFFER OF A MACHINE BECOMES FREE          *
;   *************************************************************
;
CRE2  CREATE;
BAIN  AWAIT(24),INTOP;
CLO2  CLOSE,INTOP;
ASS4  ASSIGN,II=3;
      ACT,,II.EQ.3,EVE1;
EVE7  EVENT,7;
      ACT,,,BAIN;
;----------------------------------------------
      CREATE;
EVE8  EVENT,8,1;
      ACT,10.0,XX(9).EQ.1,EVE8;
      ACT,,XX(9).EQ.2,TER;
;----------------------------------------------
OPE1  OPEN,PILEUP,1;
      ACT,,XX(4).EQ.1,TER;
      ACT,,XX(4).EQ.2,EVE7;
DON   GOON,1;
      ACT,,XX(4).EQ.1,TER;
      ACT,,XX(4).EQ.2,EVE7;
TER   TERM;
      END;
SIMULATE;
SEEDS,-43521343(1);
SIMULATE;
SEEDS,79732799(1);
SIMULATE;
SEEDS,-79732799(1);
SIMULATE;
SEEDS,34349213(1);
SIMULATE;
SEEDS,-34349213(1);
SIMULATE;
SEEDS,56231987(1);
SIMULATE;
SEEDS,-56231987(1);
SIMULATE;
SEEDS,2356971(1);
SIMULATE;
SEEDS,-2356971(1);
SIMULATE;
SEEDS,9028303(1);
SIMULATE;
SEEDS,-9028303(1);
SIMULATE;
SEEDS,53458341(1);
SIMUATE;
SEEDS,-53458341(1);
SIMULATE;
SEEDS,23478311(1);
SIMULATE;
SEEDS,-23478311(1);
SIMULATE;
SEEDS,87678623(1);
SIMULATE;
SEEDS,-87678623(1);
SIMULATE;
SEEDS,73451239(1);
SIMULATE;
SEEDS,-73451239(1);
SIMULATE;
SEEDS,3459211235(1);
SIMULATE;
SEEDS,-3459211235(1);
SIMULATE;
SEEDS,62467831(1);
SIMULATE;
SEEDS,-62467831(1);
SIMULATE;
```

```
SEEDS,987313123(1);
SIMULATE;
SEEDS,-987313123(1);
SIMULATE;
SEEDS,798324125(1);
SIMULATE;
SEEDS,-798324125(1);
SIMULATE;
SEEDS,32453241(1);
SIMULATE;
SEEDS,-32453241(1);
FIN;
```

APPENDIX-E

SAMPLE TRACE OF THE SIMULATION

```
TRACE OF RUN  1 OF THE SIMULATION

TIME =      .0000 PART #   2 HAS BEEN SELECTED TO BE PROCESSED IN MACHINE 2 ON TRACK 2 FOR OPERATION #  1
TIME =      .0000 PART #   1 HAS BEEN SELECTED TO BE PROCESSED IN MACHINE 1 ON TRACK 1 FOR OPERATION #  1
TIME =      .0000 PART #   2 SEIZES THE LOADING CRANE, LOADING AGV, LOADER, PALLET AND FIXTURE
TIME =      .0000 PART #   2 TRANSPORTED FROM THE  C P S  TO THE LOADING STATION - TRAVEL TIME =  2.000
TIME =     4.0000 PART #   1 SEIZES THE LOADING CRANE, LOADING AGV, LOADER, PALLET AND FIXTURE
TIME =     4.0000 PART #   1 TRANSPORTED FROM THE  C P S  TO THE LOADING STATION - TRAVEL TIME =  4.000
TIME =     4.0000 PART #   2 TRANSPORTED FROM LOADING STATION TO MACHINE 2 - TRAVEL TIME =   .600
           LOADING AGV CARRYING PART #   2 MOVING ON LOADING TRACK  2
TIME =      4.600 PART #   2 RELEASES CART AT MACHINE 2 - AGV TRAVEL TIME TO  C P S =   .600
           EMPTY LOADING AGV MOVING ON LOADING TRACK  2
           PART #   2 NEEDS  3 TOOLS AT MACHINE 2
TIME =     4.6000 PART #   2 LEAVING THE INPUT BUFFER OF MACHINE 2
TIME =     4.6000 PART #   2 WAITS FOR TOOLS FROM THE TOOL CRIB AT MACHINE 2 - WAITING TIME =  1.000
TIME =     5.2000 LOADING AGV IS RELEASED AT THE LOADING STATION
TIME =    10.0000 PART #   1 TRANSPORTED FROM LOADING STATION TO MACHINE 1 - TRAVEL TIME =   .500
           LOADING AGV CARRYING PART #   1 MOVING ON LOADING TRACK  1
TIME =     10.500 PART #   1 RELEASES CART AT MACHINE 1 - AGV TRAVEL TIME TO  C P S =   .500
           EMPTY LOADING AGV MOVING ON LOADING TRACK  1
           PART #   1 NEEDS  3 TOOLS AT MACHINE 1
TIME =    10.5000 PART #   1 LEAVING THE INPUT BUFFER OF MACHINE 1
TIME =    10.5000 PART #   1 WAITS FOR TOOLS FROM THE TOOL CRIB AT MACHINE 1 - WAITING TIME =  1.000
TIME =    11.0000 LOADING AGV IS RELEASED AT THE LOADING STATION
TIME =    19.6000 PART #   2 SEIZES THE UNLOADING AGV, UNLOADER, AND UNLOADING CRANE
TIME =    19.6000 PART #   2 TRANSPORTED FROM MACHINE 2 TO UNLOADING STATION - TRAVEL TIME =  1.000
           UNLOADING AGV CARRYING PART #   2 MOVING ON UNLOADING TRACK  2
TIME =    20.6000 PART #   2 WAITS FOR THE UNLOADING CRANE AT THE UNLOADING STATION - WAITING TIME =  3.000
TIME =    23.6000 PART #   2 RELEASES FIXTURE
TIME =    23.6000 PART #   2 TRANSPORTED FROM THE UNLOADING STATION TO THE  C P S - TRAVEL TIME =  2.000
TIME =    25.6000 PART #   1 SEIZES THE UNLOADING AGV, UNLOADER, AND UNLOADING CRANE
TIME =    25.6000 PART #   1 TRANSPORTED FROM MACHINE 1 TO UNLOADING STATION - TRAVEL TIME =  1.500
           UNLOADING AGV CARRYING PART #   1 MOVING ON UNLOADING TRACK  1
TIME =    25.6000 PART #   2 BELONGING TO PART TYPE  2 HAS FINISHED ALL ITS OPERATIONS
NUMBER OF PARTS THAT HAVE LEFT THE SYSTEM =   1
TIME =    27.1000 PART #   1 WAITS FOR THE UNLOADING CRANE AT THE UNLOADING STATION - WAITING TIME =  3.000
TIME =    30.1000 PART #   1 RELEASES FIXTURE
TIME =    30.1000 PART #   1 TRANSPORTED FROM THE UNLOADING STATION TO THE  C P S - TRAVEL TIME =  2.000
TIME =    32.1000 PART #   1 BELONGING TO PART TYPE  1 HAS FINISHED ALL ITS OPERATIONS
NUMBER OF PARTS THAT HAVE LEFT THE SYSTEM =   2
```

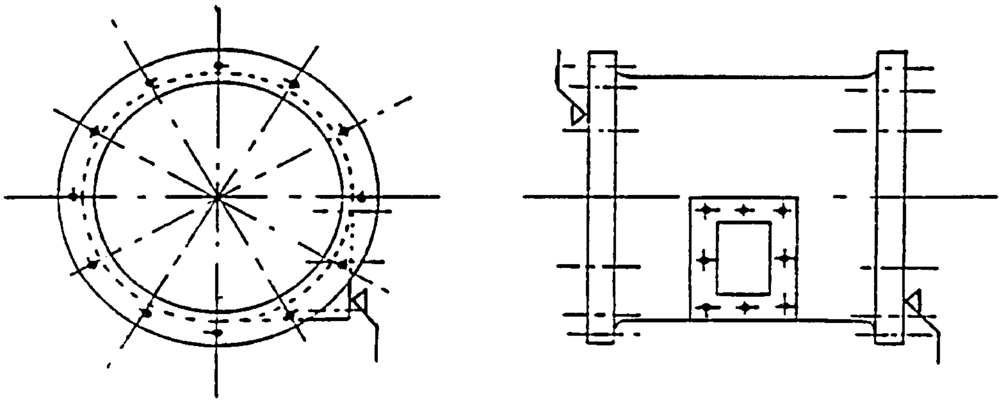
APPENDIX-F

PART SKETCHES AND PROCESS SHEETS

OPERATION SHEET

PART TYPE 1

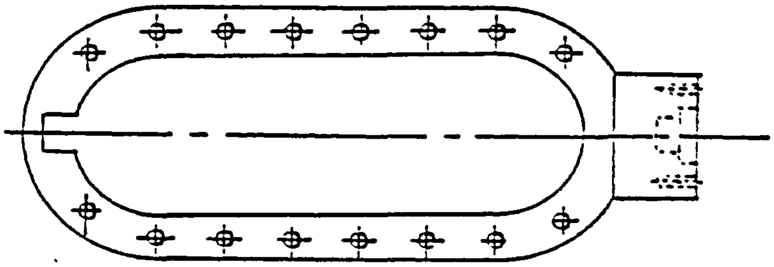
Opn	Operation Description	Tool	Time (mins)	Machines
01	Rough mill top face Finish mill top face Drill 10 dia. holes (12) Ream 10 dia. holes (12)	M1 M2 D1 R1	24	1or2or5
02	Rough mill bottom face Finish mill botom face Drill 10 dia. holes (12) Ream 10 dia. hole (12)	M1 M2 D1 R1	24	1or2or5
03	Rough mill side face Finish mill side face Drill 10 dia. holes (8) Ream 10 dia. holes (8)	M3 M4 D1 R1	18	3or4

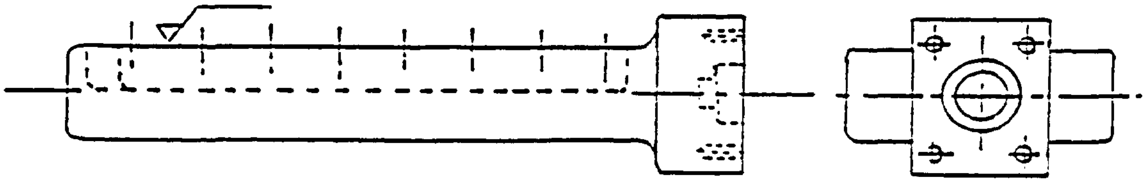


OPERATION SHEET

PART TYPE 2

Opn	Operation Description	Tool	Time (mins)	Machines
01	Rough mill top face	M5	31	1or2or5
	Finish mill top face	M6		
	Drill 8 dia. holes (16)	D2		
	Ream 8 dia. holes (16)	R2		
	Countersink 8 dia. holes (16)	CS1		

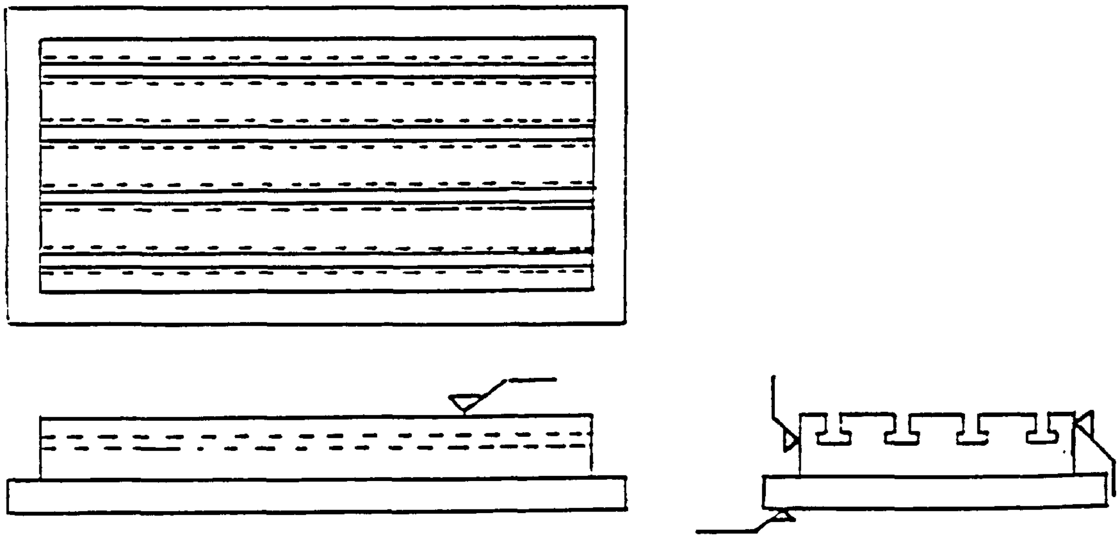




OPERATION SHEET

PART TYPE 3

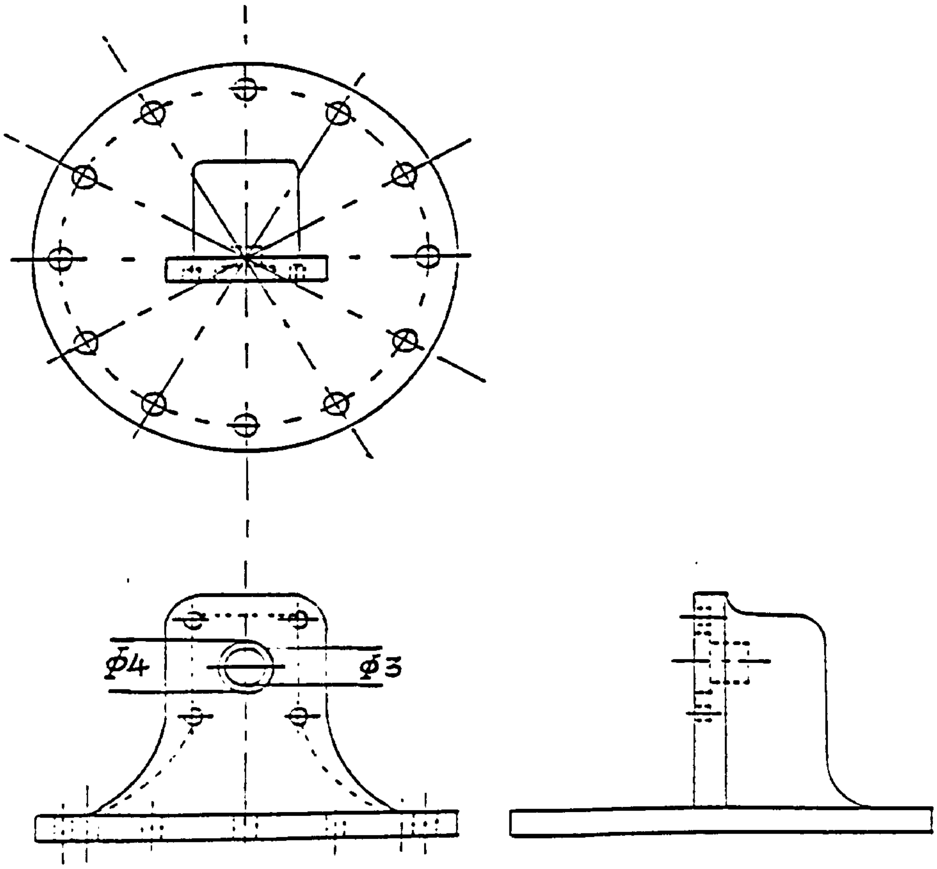
Opn	Operation Description	Tool	Time (mins)	Machines
01	Rough mill bottom face	M7	6	1or2or5
	Finish mill bottom face	M8		
02	Rough mill top face	M7	23	3or4
	Finish mill top face	M8		
	Rough mill all around	M5		
	Finish mill all around	M6		
	Mill slots (4)	M9		
	Mill bottom of slots	M10		



OPERATION SHEET

PART TYPE 4

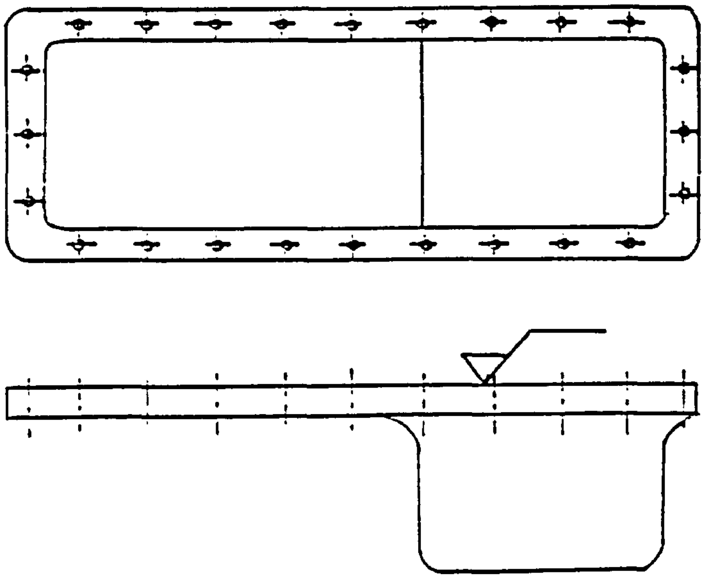
Opn	Operation Description	Tool	Time (mins)	Machines
01	Drill 10 dia. holes (12)	D1	9	1or2or5
	Enlarge holes to 20 dia. (12)	D3		
	Ream 20 dia. holes (12)	R3		
02	Rough mill end face	M3	17	3or4
	Finish mill end face	M4		
	Rough bore dia. $\emptyset_3$	B5		
	Finish bore dia. $\emptyset_3$	B6		
	Rough bore dia. $\emptyset_4$	B3		
	Finish bore dia. $\emptyset_4$	B4		
	Drill 12 dia. holes (4)	D4		
	Ream 12 dia. holes (4)	R4		
	Tap 12 dia. holes (4)	T1		



OPERATION SHEET

PART TYPE 5

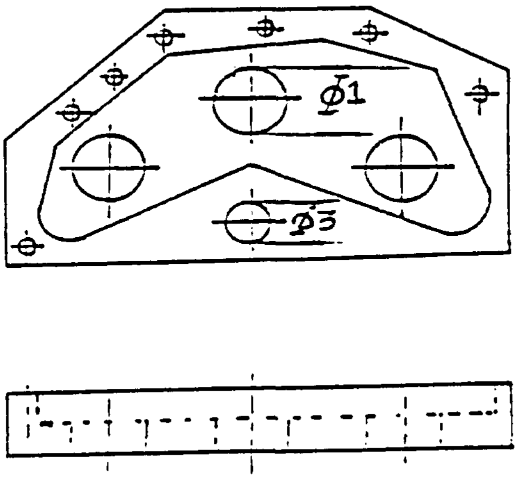
Opn	Operation Description	Tool	Time (mins)	Machines
01	Rough mill top face	M5	36	1or2or5
	Finish mill top face	M6		
	Drill 5 dia. holes (24)	D5		
	Ream 5 dia. holes (24)	R5		
	Tap 5 dia. holes (24)	T2		



OPERATION SHEET

PART TYPE 6

Opn	Operation Description	Tool	Time (mins)	Machines
01	Rough mill bottom face	M7	8	1or2or5
	Finish mill bottom face	M8		
02	Rough mill top face	M7	14	1or2or5
	Finish mill top face	M8		
	Drill 10 dia. holes (7)	D1		
	Enlarge holes to 20 dia. (7)	D3		
	Ream 20 dia. holes (7)	R3		
03	Rough mill inner profile	M9	21	4
	Finish mill inner profile	M11		
	Rough bore 3 holes of dia. ϕ_1	B1		
	Finish bore 3 holes of	B2		
	Rough bore 1 hole of dia. ϕ_3	B5		
	Finish bore 1 hole of dia. ϕ_3	B6		



APPENDIX-G

FIXTURE TYPE AND CUTTING TOOLS INFORMATION

FIXTURE TYPE DETAILS

Part type	Opn	Fixture
1	01	1
	02	1
	03	1
2	01	2
3	01	3
	02	3
4	01	4
	02	4
5	01	3
6	01	5
	02	5
	03	6

CUTTING TOOLS INFORMATION

Tool #	Tool code	Tool description
1	B1	Boring tool - I
2	B2	Boring tool - II
3	B3	Boring tool - III
4	B4	Boring tool - IV
5	B5	Boring tool - V
6	B6	Boring tool - VI
7	CS1	Counter sinking tool - I
8	D1	Drill 10 mm dia.
9	D2	Drill 8 mm dia.
10	D3	Drill 20 mm dia.
11	D4	Drill 12 mm dia.
12	D5	Drill 5 mm dia.
13	M1	Milling cutter - I
14	M2	Milling cutter - II
15	M3	Milling cutter - III
16	M4	Milling cutter - IV
17	M5	End milling cutter - I
18	M6	End milling cutter - II
19	M7	Milling cutter - V
20	M8	Milling cutter - VI
21	M9	End milling cutter - III
22	M10	End milling cutter - IV
23	M11	End milling cutter - V
24	R1	Ream 10 mm dia.
25	R2	Ream 8 mm dia.
26	R3	Ream 20 mm dia.
27	R4	Ream 12 mm dia.
28	R5	Ream 5 mm dia.
29	T1	Tap 12 mm dia.
30	T2	Tap 5 mm dia.

VITA AUCTORIS

1960	Born in Madras, India on the 28th of June.
1976	Completed high school education from Don Bosco Matriculation School, Madras, India, securing first division.
1977	Completed pre-university course from Loyola College, Madras, India, securing first division.
1982	Graduated from University of Madras, India with a Bachelor of Engineering (Honors) Degree in Production Engineering, securing first division with distinction.
1987	Currently a candidate for the M. A. Sc. Degree in Industrial Engineering at the University of Windsor, Windsor, Canada.