AN INVESTIGATION ON THE EFFECTS OF RELEASE DATE IN A MULTI-PRODUCT ASSEMBLY SHOP

by 1264

BASUDEB SARKAR

B. Tech. (Hons. Electrical), Indian Institute of Technology
Kharagpur, India, 1965

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1969

Approved by:

L. E. Grosh
Major Professor

TABLE OF CONTENTS

CHAPTER I

1.0 INTRODUCTION

The job shop scheduling problem has been the subject of extensive research in recent years. Most of these studies are however concerned with scheduling a set of jobs arriving sequentially over time at a simple job shop. The assumption that all jobs are independent eliminates the possibility of combining the jobs at any stage of the process. So, only these jobs compete for the available resources i.e., the man or the man-machine systems. Though this model has many practical applications, there is yet another group of problems for which the above assumption is invalid.

Manufacture of any complex product, made of several component parts is an example of the other group. These products must be processed in a series of steps. In each of these steps various component parts, in integral amounts are combined to create a new part--an assembly. In every one of these steps there is a precedence relationship between the parts. The parts can be processed in series or in parallel as long as this precedence relationship is maintained. We will develop a technique to describe the component-assembly relationship, graphically as well as mathematically. Matrix algebra is used to represent these relationships and to calculate the quantity of each of these components.

This problem becomes slightly different when we consider the manufacture of a number of such products. In life the component parts of each product are not totally different. The fact is, these component parts are used in the end-items in an inconsistent fashion. For example, a firm which manufactures different types of electrical motors uses the same bolts, nuts etc. (detail parts) and the same magnetic coil casings (sub assemblies) in several of the models. Therefore, when we plan to schedule production for these types of parts, we might as well determine the quantities of each component parts required to meet the end-items' demand.

The problem of planning a production schedule has been dealt with in two major parts:

Part I: Requirement generation.

1. Part explosion
2. Demand and requirement

Part II. Scheduling

3. Lot size
4. Machine Scheduling.

The total scheduling problem is discussed in these introductory pages. The initial data is given with the customers' orders. These orders dictate the type and quantity of components to be produced. They also determine the time when these components have to be completed.

The processing of each component part involves a "set of distinct diverse processing facilities and a set of tasks, each requiring work to be done in sequence, at one or more of these facilities" (6). In general, the order in which a task passes from one facility to another is not the same for all tasks in the system. The objective of scheduling is to array the tasks over the facilities in time subject to occupancy and precedence constraints, so as to optimize some temporal characteristics of the task and/or the facilities (7). Clearly, the task may be job-lot quantities of piece-parts with sequences of machining operations to be performed.

Most research in the area of job shop scheduling has been accomplished using digital computer simulation as the method of analysis. Using simulation techniques, Conway (5) was successful in determining superiority of shortest operation rule for a simple job shop. Rowe (16) used the simulation technique to establish decision rules for job lot production scheduling. Many attempts have been made to develop a priority rule for the assembly job shop which would perform well in all situations.

This problem has been formulated mathematically using linear programming techniques. In many cases this approach becomes undesirable due to vast number of constraint equations generated.

More experiments need be done to study the behavior of the model with reference to the release date of orders. Manne (12) tested a model for finding economic lot sizes for a machine shop required to produce many different items so as to meet a rigid delivery schedule. In our experiment we will cluster the end-item orders upto a certain length of time before releasing it to the shop. Using one standard priority rule for the assembly job shop, we will study the effects of this release date on the overall performance of the system.

1.1 DESCRIPTION OF THE PROBLEM

An assembly job shop consists of a number of groups of processing facilities. Each facility group consists of a number of identical machines. Orders for end-items come from customers. These end-items, made up of groups of parts (sub-assembly and details) arrive at the shop at a certain exponential rate. The sub-assembly and detail parts consist of several individual jobs requiring the use of the available machine shop resources. The amount of processing time for any part is known with a mean and standard deviation. The total requirement of these parts are determined after adjusting against the available inventory. The cumulative requirements for the component parts are calculated until they are released to the shop. The parts are then scheduled to be manufactured as per the current operating procedure, and are dispatched to an assembly shop on completion. The existence of a predictable time between successive operations of each part gives one the ability to schedule the completion of parts at desired due dates with reasonable assurance. This ability is crucial in the production of parts for assembly lines; the shortage of one part could result in the shutdown of the line or an expensive out-of-stock cost. In order to maximize the on-time completions of end-items we attempted to integrate the scheduling, loading and dispatching functions in a manner which capitalizes on the natural movements of work through the shop.

We define the control period as the period at the end of which the orders are released to the shop. The manner in which the orders are released depends on the type of orders, their arrival rate and their due dates. If the orders are released as soon as they arrive at the shop we might face several problems. Firstly, since the shop makes requirements for subsequent orders, it will be difficult to follow this particular order to the end of processing. Secondly, since some of the component parts required by this order may be identical as that of the subsequent orders we will lose the benefits of making the identical parts in lots. Thirdly, shop utilization can not be at its best.

1.2 PURPOSE OF THE RESEARCH

The purpose of this research is to develop a model for exploring the part requirements for a multi-product assembly shop so that the common parts can be identified and stored in part files. The objective of this experiment is to evaluate the performance of various releasing dates in an assembly job shop.

CHAPTER II

2.0 PART EXPLOSION AND NETTING

Any complex product made of several components must be processed in a series of steps. Each of these component parts, in integral amount, are combined to create a new part--an assembly. In every one of these steps a certain number of identical articles, the lot or batch, is processed with the same production operations. We will develop a technique to describe the component-assembly relationship, graphically as well as mathematically. Matrix algebra is used to represent these relationships and to calculate the number of components which are integrated into the manufactured parts. In a typical industrial system, orders for the end-items arrive at the shop. When we receive such order, we will generate the direct requirements for the end-item. We then net the requirements with the available stock and determine the requirements for the indirect components. Arrival of every new order necessitates this part explosion and netting. Unless we determine the quantities of the component parts which are to be scheduled at the end of the release date our experiment to establish this release date will not be successful.

2.1 DEFINITION OF BASIC TERMS (8)

2.11 Operations

An operation is a procedure that uses the facility to alter the physical, chemical or location state of the part being manufactured.

2.12 Production facility

A production facility is a man or man-machine combination which performs the operation. It may be an individual worker equipped with the simplest hand tools (or no tools) or it may be an entire manufacturing or purchasing department in a production system. In our case it is the man-machine combination which performs an operation on a part.

2.13 Operation sheet

The operation sheet is a list of manufacturing operations that must be performed in the stated sequence on the specified facilities to convert a certain amount of raw materials, detail parts, and assemblies into a finished part. In our problem we will consider only the groups of facilities required by the part.

2.14 Part

Final assembly or end-item is a part not assembled to another part. This end-item is not a component for any other assembly in the system. Such a part represents the output of the manufacturing system and is shipped to the customer or to another department of the company outside our jurisdiction. An end-item therefore does not appear as a comsumed sub-part or component on any bill or material in the system.

A sub-assembly is manufactured from a number of parts (detail parts). This is a component of a main-assembly (end-item) and is directly consumed by it.

A detail part is a "discrete" component of any type of assembly and it does not require any part to assemble itself. These parts are directly made from the raw materials and are consumed by the sub-assemblies and end-items. They themselves, therefore do not consume any part which we have processed in the system.

In our problem each of the sub-assembly and detail parts are unique in a sense that the same part has the same operation sheet throughout the simulation. The operation sheet of each of the sub parts and detail parts are generated only once at the beginning of simulation and they are not altered until the end of an iteration has occurred. These operation sheets, however, may be different for another iteration but the same operation sheet should be followed until the end of that particular interation.

2.2 DIRECT AND INDIRECT CONSUMPTIONS

As we said earlier a certain number of sub-parts are consumed to generate an assembly. A part i is directly consumed by assembly j, if it is directly required in the assembly operation. For the "direct" consumption of part i in the assembly j, part i is a positive quantity in the bill of material of assembly j. Part i is indirectly consumed by assembly j, if part i is consumed by a directly consumed sub-assembly of part j. In our case only the end-items can have both direct and indirect consumption of detail parts.

2.3 BILL OF MATERIAL

The bill of material of a given end-item is a listing of the quantities of parts which are directly consumed per unit of the given end-item. If an end-item has a null bill of materials (that is, zero quantities of all parts are directly consumed in its production), it should be assumed that the end-item does not require any parts to be made.

For each entry in a bill of material there will be three items of information:

(1) Designation of the end-item

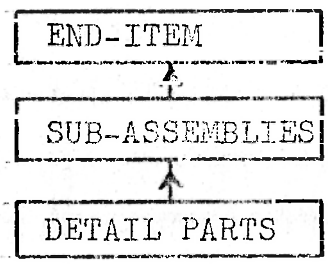
(2) Designation of the directly consumed part

(3) The quantity of the directly consumed part which is consumed per unit of the produced consuming part.

In our investigation both (2) and (3) are generated using pseudo-random numbers.

2.4 LEVEL ORDER

The hierarchy of parts (i.e., final assembly, sub-assembly and detail parts) leads to the expression "level." Parts requiring the same number of assembly steps to appear in the end-item, are parts of the same level. We assign the end-item the level number "3" and a two digit number "30" to denote its number. We denote the sub-assembly as level number "2". In our investigation we have assumed ten sub-assemblies, numbering 20, 21, . . . , 28, 29; likewise we denoted ten detail parts by numbering 10 to 19. This hierarchy of parts is shown in fig. 2.1.



Level Number	3
Part Number	30
Level Number	2
Part Number	20, 21, . . . , 28, 29
Level Number	1
Part Number	10, 11, . . . , 18, 19

Fig. 2.1

2.5 PART STRUCTURE DIAGRAM

The configuration of an assembly can be represented with a part structure diagram (Fig. 2.2).

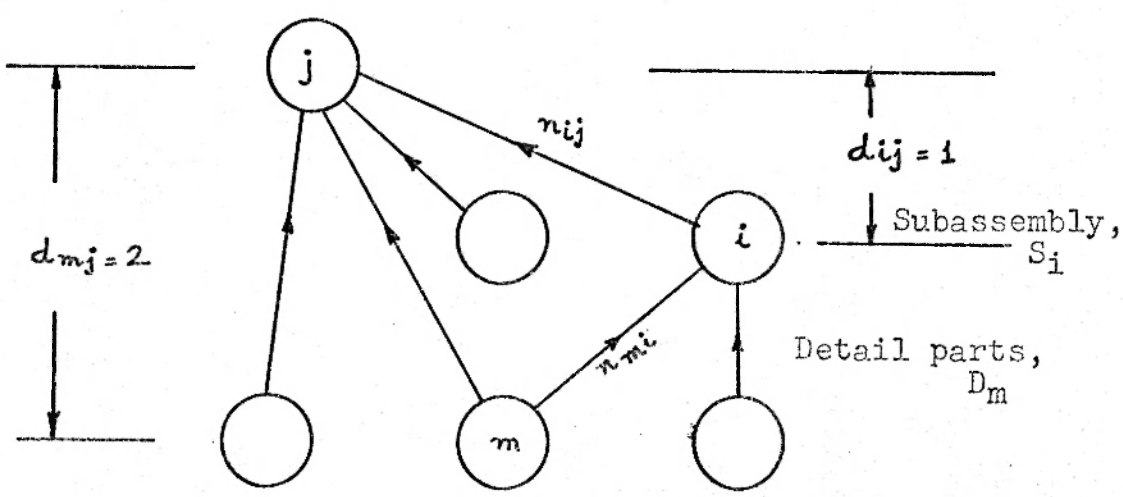


Fig. 2.2

Each part in the figure 2.2 is shown by a small circle or node, which includes the part number. The direct consumption of one part by another is represented by a line connecting the two circles. The arrow points to the consuming material only. The number i represents the number of the sub-assembly part and n_{ij}, the quantity of part i that is required to make one unit of end-item j. The number m represents the number of detail parts and n_{mi} represents the quantity of detail part m that go to make one unit of the sub-assembly i. Similarly, n_{mj} represents the quantity of detail part m that are required to make one unit of end-item, j. So, as per our definition both n_{ij} and n_{mi} are direct requirements of part m for the end-items j. Quantity n_{mi} is indirectly required by part j, but is directly required by part i. In our investigation, all the numbers i and m and their quantities, n_{mj}; n_{mi} and n_{ij} are generated for a given end-item j.

2.6 TOTAL REQUIREMENT MATRIX

2.61 Definition

In the total requirement matrix, T each element t_{ij} is defined as the total quantity of part i needed to make one unit of part j. Since the total requirement matrix is very sparse, we will develop a method to generate and store the requirement matrix in a small computer storage space.

2.62 Procedure

Step I: We will generate randomly the quantity of end-item. Let us assume this quantity is 10.

Step II: We will generate randomly the quantity of level 2 parts (sub-assembly) that go directly into one unit of the end-item. Let us assume this number is 3. Next we will generate the part number of these parts. Suppose we find these numbers are 21, 25 and 27. This means that, sub-assembly 21, 25 and 27 are required by the end-item. In the following step the quantities for each of the sub-assemblies are generated randomly. Suppose these generated requirements for part numbers 21, 25 and 27 are 10, 15 and 5 respectively. The total requirement of these three parts is obtained by multiplying each of them by the number of end-items. So the total required quantities for part #21 are 10 x 10 = 100; part #25 are 15 x 10 = 150; and for part #27 are 5 x 10 = 50. Suppose the available inventory for part #21 are 20 units; for part #25 are 50 units; and for part #27 are 10 units. Therefore, the demanded quantities for part #21 are 100-20 = 80 units; for part #25 are 150 - 50 = 100 units; and for part #27 are 50 - 10 = 40 units. In the matrix representation this is

$$\begin{matrix}\text{Part \#21}\\ \text{Part \#25}\\ \text{Part \#27}\end{matrix} \quad 10 \cdot \begin{bmatrix}10\\15\\5\end{bmatrix} - \begin{bmatrix}20\\50\\10\end{bmatrix} = \begin{bmatrix}100\\150\\50\end{bmatrix} - \begin{bmatrix}20\\50\\10\end{bmatrix} = \begin{bmatrix}80\\100\\40\end{bmatrix}$$

Assumption: there is no available inventory for end-items.

Each of these quantities are stored in a column vector in which the rows indicate the part number and the elements of the vector represent the quantities demanded for each part.

Step III: The demanded quantities for the sub-assembly parts are updated each time an order for an end-item arrives.

Step IV: The detail part requirements are calculated against the updated quantities of end-items and sub-assembly parts. To generate the part number and their individual quantities for one unit of subparts and end-items the pseudo-random number generator is used. Suppose we generate a requirement of 5 units of part #11 and 2 units of part #13 for making one unit of part #21. We require 2 units of part #13 to make one unit of part #25 and 1 unit of part #14 to make one unit of part #27. We also find that 10 units of part #13 are required to make one unit of the end-item. Therefore, in the matrix representation this becomes:

$$\begin{matrix} \text{Part \#11} \\ \text{Part \#12} \\ \text{Part \#13} \\ \text{Part \#14} \end{matrix} \quad \underset{(A)}{\begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 10 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \cdot \underset{(B)}{\begin{bmatrix} 80 \\ 100 \\ 40 \\ 10 \end{bmatrix}} = \underset{(C)}{\begin{bmatrix} 400 \\ 0 \\ 460 \\ 40 \end{bmatrix}}$$

Suppose the available inventory for these parts (part #11, 12, 13 and 14) are 100, 0, 260 and 100 units. Therefore, the demanded quantities for part #11 are 400 - 100 = 300; for part #12 is 0; for part #13 are 460 - 260 = 200; for part

#14 are 40 - 100 = -60 i.e., the demanded quantity for part #14 is 0 with an inventory of 60 units.

We find that the requirement matrix is very sparse i.e., there are only a few non-zero elements in it. To save computer memory we do not store the requirement (A), but store only the significant non-zero elements. As soon as an order for an end-item arrives, we repeat the procedure to update the quantities of each sub-assembly and detail parts. Thus we need only a (10 x 1) array for storing the detail parts and a (10 x 1) array for storing the sub-assembly parts.

CHAPTER III

BILL OF MATERIALS AND LOT SIZE

3.0 BILL OF MATERIALS

As discussed in Chapter II, the bill of materials shows the listings of the quantities of parts which are directly consumed per unit of the end-item. A simple product is made up of several component parts. But in many cases, the product consists of certain subparts, each or some of which are again an assemblage of several detail parts. In an ideal job shop situation the product (end-item) is made up of several jobs or subparts. When an order for an end-item arrives at the shop, we are concerned with making all of these subparts or jobs in a desired fashion. In this case we assume that these parts have usage in this product only and they would not be used in the next or subsequent products. So we manufacture exactly the required quantities of the parts. Therefore, in an ideal job shop situation there is very little or no inventory on hand. The manufacturer's attention is fully focused on completing these parts so that the job order can be maintained. The bill of material in this case will contain only the listings of the jobs and their precedence relationships.

There is another situation where the subsequent orders for end-items result in making the same subparts. In this case the subsequent end-item requires some parts which are used by the previous end-item. For example, one unit of end-item A requires 5 units of S1, 3 units of S2 and 5 units of D1 and 4 units of D2, where S1 and S2 are sub-assemblies 1 and 2, and D1 and D2 are detail parts 1 and 2. One unit of the next end-item B requires 10 units of S1 and 10 units of D1. This kind of problem is common in the case of firms manufacturing a varieties of end-items. Examples are, firms making different varieties of consumer goods, viz., TV sets, radios, electric fans, airconditioners and even automobiles.

In the above example, if we wait till the next order arrives, we can release to the shop an order of (5 + 10) = 15 units of S1, 3 units of S2, (5 + 10) = 15 units of D1 and 4 units of D2. We can therefore afford to delay the release date of orders to the shop subject to the condition that the scheduled delivery dates for the orders are maintained. In such situations, the part requirements should be reviewed with the arrival of a new order.

3.1 LOT SIZE

Normally a company produces many products of the same type but not identical. Many of the parts are used in a number of end-items, however, in an inconsistent fashion. The ordering fashion of the end-items is also of uncertain nature.

The production has to fulfill the requirements R for each part. The available production hours of each machine, the time it takes to manufacture each required part is known.

There are also costs for each set-up and for carrying an inventory. The problem is to determine the lot size for each individual part so that the production requirements are met and the combined costs of manufacturing and inventory are minimized.

3.2 MATHEMATICAL STATEMENT OF THE PROBLEM (18)

3.21 Requirement (or Market Demand Restrictions)

The manufacturing company produces I parts, i on the M available machines, m. We are asked to plan for the T future planning periods, t. The requirement r_{it} of each part i and each planning period t, are obtained from the end-item orders. We must determine the quantity x_{it} of part i that is to be manufactured in the t^{th} period and call it the lot size or order quantity X_{it}. The cumulative requirement $_{it}$ for each part i from k = 0 upto and including period t can be given by

$$\rho_{it} = \sum_{k=0}^{t} r_{ik}$$

and cumulative order quantity can be written as

$$X_{it} = \sum_{k=0}^{t} x_{ik}$$

The production plan must meet the given requirements. So we get the inequality:

$$X_{it} \geqslant \rho_{it}$$

This is shown in Fig. 3.1.

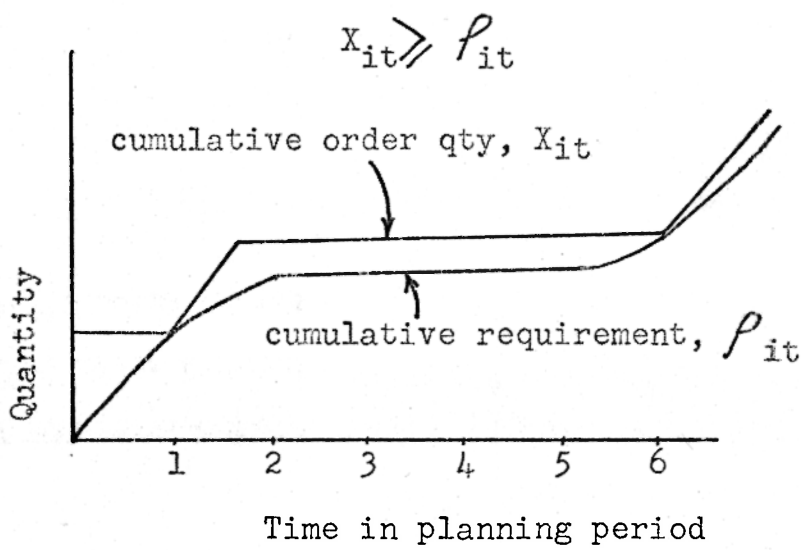


Fig. 3.1

3.22 Machine restrictions

The production plan must stay within the available machine capacity, say ϕ_{mt}. To satisfy this requirement we need to determine the machine time required by the production plan. If the production of a single part i on machine m needs τ_{im} hours, the load on machine m by part i is expressed by:

$$h_{mit} = \tau'_{im} \; x_{it}$$

For all the parts processed on machine m the total requirement time is given by

$$h_{mt} = \sum_{i=1}^{I} \tau_{im} x_{it}$$

This machine load must stay within the available machine capacity for that period. So we have the inequality,

$$h_{mt} \leqslant \phi_{mt}$$

3.23 Inventory cost

The production plan specifies that a cumulative quantity of x_{it} parts are manufactured in period, t whereas there are only $\hat{\rho}_{it}$ parts required. All these extra parts i.e., $(x_{it} - \rho_{it})$ parts produced in period t for a later usage have to be stored. This requires maintenance of an inventory.

If c_i : cost of each part, i

p : carrying rate in per cent

pc_i : the carrying cost of each part i for one period, t

C_I : the total inventory cost

then, we can state:

$$C_I = \sum_i \sum_t pc_i \; (x_{it} - \rho_{it}).$$

3.24 Set-up cost

If the set-up cost of manufacturing part, i on machine m is given by $\hat{c}_{im}$ and $U(x_{it})$ is the number of set-ups for part i during the period t then the total set-up cost can be written as:

$$C_s = \sum_i \sum_m \sum_t \hat{C}_{im} \cdot U(x_{it}).$$

3.25 Overtime cost

The production time outside the normal machine capacity $_{mt}$ is penalized by additional costs, the overtime costs. The overtime $(h_{mt} - \eta_{mt})$ on machine, m in period, t generates total overtime cost C_c which is given by

$$C_o = \sum_m \sum_t (h_{mt} - \eta_{mt}) \cdot \bar{C}_m \cdot \bar{p}$$

where, $\bar{C}_m$: the normal production costs per hour of machine m,

$\bar{p}$: the overtime costs expressed as a percentage of normal production costs per hour.

3.26 Total cost equation and method of obtaining solution

Adding up inventrcy, set-up and overtime costs for the production plan, we get

$$C_{total} = C_I + C_s + C_o$$

$$= p \sum_i \sum_t C_i (X_{it} - \hat{f}_{it}) + \sum_i \sum_m \sum_t C_{im} \cdot U(x_{it})$$

$$+ \sum_m \sum_t (h_{mt} - \eta_{mt}) \, \bar{p} \quad \bar{C}_m \,..$$

The problem is to acquire a set of x_{it} that will minimize the total cost equation subject to the inequalities:

$$X_{it} \geqslant \rho_{it} \quad \ldots . (1)$$

$$\text{and} \quad h_{mt} \lesseqgtr \phi_{mt} \quad \ldots . (2)$$

$$\text{with} \quad X_{it} = \sum_{k=0}^{t} x_{ik} \quad \ldots (3)$$

$$\text{and} \quad \rho_{it} = \sum_{k=0}^{t} r_{ik} \quad \ldots (4)$$

This optimization problem is a special case of linear programming problem. To realize the magnitude of the problem we consider the example of a situation where we are to make 20 different parts on 10 different groups of machine for a total planning interval of 10 weeks. So, the number of unknowns are 20 · 10 = 200 i.e., equation (3) has to be writtin for each part, i and each production period, t. This amounts to 200 equations. Similarly, equation (1) will give rise to 200 equations. Equation (2) represents the condition for staying within the maximal machine capacity. For 10 machines and 10 planning periods, this will give 100 inequalities. The cost equation contains the sum of three terms. The first term contains I.T = 20 10 = 200 terms, the second term contains I.M.T. = 20 10 · 10 = 2000 terms and the third therm contains M.T. = 10 · 10 = 100 terms. We thereby gain a total of 2300 terms.

Summing up, we have 200 unknowns, 500 inequalities and an objective function with 2300 terms. The handling of this size of problem is not worthwhile compared to the benefits derived from it.

In our research we propose to solve the lotting problem in the following manner.

3.3 LOTTING

The manufacturing company in our example produces a variety of parts. The requirement of each part fluctuates from one period to another. If we draw the requirement pattern of parts, it will be like Fig. 3.2.

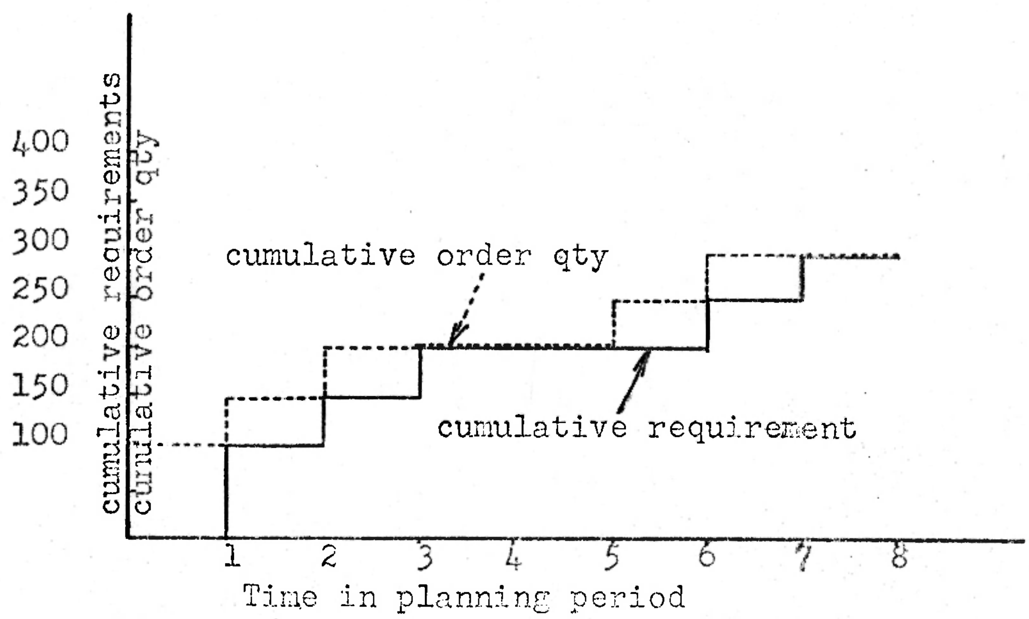


Fig. 3.2

To have a satisfactory scheme we must have a production plan that meets the requirement i.e., we have the required quantity of parts ready in hand before we use the part. We can have a production plan described in Fig. 3.2. This plan envisages production of the exact quantity at each stage of the planning period. If at a certain date some quantities of parts are required, we will determine the date when the production of this part should be started. For example, in Fig. 3.2 we need 50 units at the end of period 2. The production of these 50 units require 1 plan period. We will therefore start making these 50 units at the end of period 1. We also find in the above figure that we do not require this part at the end of 4th and 5th period. But we need 50 units of this part at the end of 6th period. Since the time to make these 50 units is only 1 plan period, we will start making these 50 units at the beginning of 6th period and not before.

The main advantage of this scheme is no in-process inventory; the parts are manufactured as and when they are required. So it has all the advantages of carrying no in-process inventory, including minimum record keeping for the parts.

This process may be ideal for a true job shop model. But in situations where the parts are used over and over again it has many potential disadvantages, viz.,

1) the cost of set-up becomes high,
2) the cost of overtime might have to be included when the production can not be met.

3) there is always a risk of running hand to mouth, specially in an assembly plant. Here any unforeseen cause might delay the production of any part during any period. This means that the next assembly operation can not be performed. Referring to our example in Fig. 3.2, if the machine breaks down during the period 6-7, the estimated 50 units of the part can not be ready at the end of the 6th planning period. As a result the assembly which uses this part will be delayed.

To overcome all these difficulties we shall adopt a scheme for scheduling the parts in lots. Under this proposed plan, we would accumulate the end-item orders for a certain length of time. This time period may be one week or one month or one year depending upon the manufacturing situation and the nature of orders. The effect of this release date on the performance of the shop and the overall system will be investigated during our experiment.

In establishing the release date we face the following problems. Suppose, the orders are released to the shop after two weeks. But is this the proper time to release the orders? We have the restriction of completing the order before its due date. It is necessary to set the due dates at a projected future period, say four weeks from the day the order is received. At the end of the release date, orders are released to the shop. Normally, the due date is more or less fixed for the orders. The sales department generally makes a commitment regarding the delivery of orders. First, we will assume this commitment is a justifiable one. Our efforts will be to improve the shop performance by varying the release date (control period).

If the control period is short then the situation is like a true job shop. If it is too large it is likened to a flow shop. In our investigation we will assume an intermediate situation. We know that the same parts are ordered. But their quantity and arrival time are uncertain. In our simulation experiment we will generate the quantity and arrival time for orders and try to establish a suitable control period. The proposed scheme is shown in Fig. 3.3.

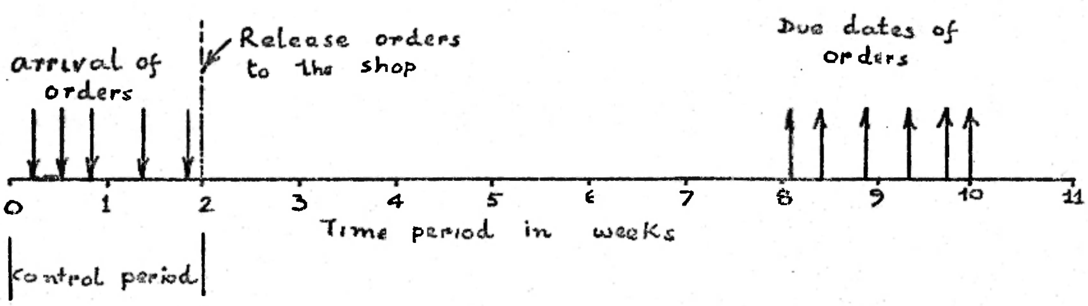


Fig. 3.3

CHAPTER IV

DESCRIPTION OF COMPUTER MODEL

4.1 DEFINITION OF BASIC TERMS

<u>The shop</u> is composed of a set of machines (resources or men) partioned into a set of machine groups or work centers. The size of the load on the shop will depend on the number of machine groups available as well as the number of machines in each machine group.

<u>The product</u> or end-item is an assembly of parts. These parts are to be processed in the shop as per certain precedence relationship.

<u>The processing time</u> is associated with each part and can be defined as the actual machine time required to complete the operation.

<u>The assembly shop</u> is a fictitious shop where all completed jobs await the completion of the last job of their respective products. When all of the jobs of a product are complete, the product is allowed to exit from the system.

3.2 THE MODEL

The model is chosen to represent an assembly job shop. This shop has 5 groups of machines or processing facilities. Each facility group consists of 2 identical machines. Each order for end-item arriving at the shop contains several subparts and detail parts. The requirement and demand for these parts are calculated as described in Chapter II. The demanded parts require the usage of available shop resources. The amount of processing time for a particular part is specified with a mean and standard deviation. Similarly the production facilities on which a part is processed is known. We have used a random generator to generate the machine group requirement. The parts are scheduled according to the current operating procedure and are dispatched to an assembly shop on completion. The existence of a predictable time between the successive completion of each part gives one the ability to schedule the completion of parts at desired due dates with reasonable assurance. This ability is crucial in the production of parts for an assembly shop; the shortage of one part could result in the shut-down of the assembly line or an expensive out-of-station cost. In order to maximize the number of on-time completion of parts, we attempted to integrate scheduling, loading and dispatching functions in a manner which capitalizes on the natural movement of work through the shop.

Assumptions:

1) We assume that operation time i.e., the actual time spent on a machine is small compared to the total time the part spends in the shop.

2). The movetime between machine centers plus waiting time is approximately the same for all operations. The result is: the total time a particular part spends in the shop is highly correlated with the number of operations to be performed and is nearly independant of the total machining time required for its completion. It is also highly dependant on the number of parts the product contains (16).

4.21 FORMULATION OF THE MODEL

The following rules are incorporated while forming the model.

4.22 LOADING RULE

A series of I end-item orders arrive randomly at an assembly shop. Each end-item requires some of the 10 sub-assembly and 10 detail parts in a random quantity. The quantities of each of these 10 sub-assemblies and 10 detail parts are updated each time a new order for end-item is received. We assume there may be some on-hand inventory for the sub-parts and detail parts. But there is no available inventory for the end-item.

At the end of the control period (which we intend to vary) we determine the netted quantities of accumulated parts to be made. Each part is scheduled according to an established scheduling rule (which we will describe next). From that schedule we know that a particular part is to be processed within a given time period. If we arrange all the jobs being scheduled in a matrix (Fig. 4.1), we have formed a prediction of the load on each machine group for each time period, i.e., the sum of all the estimated operation times for each time period, by the machine group.

Time in hours

80 100 120 140 160 180

Machine group \ Time period	I	II	III	IV	V
1	part #20	part #25			
2			part #26		
3					part #27
4					
5	part #24	part #24			

Fig. 4.1

4.23 SCHEDULING RULE

Each job is scheduled backward from its due date. The due dates for each level of parts are calculated by subtracting the total processing time of its next higher level parts from the due date of the end-item. Thus if the due date for end-item is at 180 hours, and the total processing time of level 3 parts is 20 hours, then the due date for level 2 parts is at 160 hours (i.e., 180 - 20).

4.24 Dispatching Rule

In the above operation of the job shop we must also dispatch from a queue of jobs waiting for being processed on the machine. Each end-item should be completed by the due date. The priority of the parts will reflect the completion of the end-items with the shortest possible delay.

For example, if all the parts waiting for a particular machine group must be completed at the end of the next time period, they can be processed in any order. If, however, a part is late and has an earlier due date than the rest of the parts in the queue, it is worked first so that it may "catch up" with its time period. The rule is "the job in the queue with the earliest due date for the operation will be worked first." In case of ties, choice should be made on the basis of shortest operation.

4.3 MEASURES OF PERFORMANCE

The performance of alternative control period for releasing the orders to the shop could be measured in a number of different ways. If one were to state costs for overtime, storage of jobs completed ahead of schedule, work-in-process inventory, lateness, administration of the system etc., it would be possible to have a single measure of performance--total cost. However, the relative values of these costs change from shop to shop. Therefore, the following measure of performance are used in this investigation to provide some basis conclusions of general applicability.

1. Deviations from completion due-date

   a. Products earliness

   b. Products lateness

2. Wrok-in-process inventory i.e., average number of parts in the shop

3. Shop utilization.

CHAPTER V

THE SIMULATION

5.0 INTRODUCTION

Ideally, verification of new concepts and methods should be conducted in actual machine shops; however, interference with daily routines and the inevitable disruptions would prove prohibitive. Aside from possible resentment and high cost, results would be unrealistic without a properly controlled experiment, as well as the loss of much valuable time in obtaining and analyzing data. Computer simulation, on the other hand, provides a suitable means for testing alternate rules without spending many months to evaluate each parameter. A better understanding of the production system is possible as a result of a rapid and unbiassed means for studying the details of factory operations.

An analysis of our proposed system shows it has a number of interacting parameters viz.,

1) the arrival of orders for end-items,
2) the part-structure of end-items, i.e., what quantities of parts are required to produce one unit of end-item,
3) the machine or processing facility requirement for each part,
4) the processing time for each part,
5) the due-dates for the end-items.

The effect of changing these parameters can be well studied by using simulation techniques.

In our formulated system we are interested in two kinds of objectives:

1) evaluation of alternative rules for operating the system,
2) study the system performance if the system parameters are changed.

While evaluating alternative rules for operating the system we would investigate with the following objectives:

1) to determine the best possible control period for releasing orders to the shop,
2) to improve the processing efficiency of the shop in terms of machine utilization and congestion in the shop,
3) to improve the on-time completion of end-items.

5.1 ORGANISING THE COMPUTER MODEL

This simulation model was programmed in FORTRAN IV utilizing GASP II (General Activity Simulation Program), a Fortran based simulation language developed by Pritsker and Kiviat (15). The simulation was conducted on IBM 360/50 computer.

5.2 THE SIMULATION MODEL

The next event type simulation was developed. The specific events are:

1) Arrival of an order for an end-item,

2) End of processing for a part .

The model was designed so that a special product departure from the system would be unnecessary. This was accomplished by means of a variable JBPRT(LEV) which stored the number of uncompleted parts in each level. When the last part in the level was completed JBPRT(LEV) is set equal to zero.

Entities of the system are parts and machine groups. As a permanent entity each machine group is considered separately and its current status is available at any time. Parts are classified as a temporary entity and as such must be associated with enough informations to maintain their integrity. This is accomplished by defining various part attributes which follow any particular part until its processing has been completed and statistics calculated.

To satisfy the above requirements we have used 7 files. The first file being an event file. The second file is the part file for storing all the informations about the parts. The remaining files are for each of the machine groups.

5.3 ENTITIES AND ATTRIBUTES (15)

GASP has been designed to facilitate "next event" types of simulation. In such simulations, simulated time progresses from one event to the other until (1) an end of simulation event occurs, or (2) a preplanned total simulation time is exceeded.

An event is defined as "an occurrence, taking place or possibility of taking place of a change in the state of a system." Events take place at specific points in time as determined by the system to be simulated. It is therefore necessary that the projected times of occurrence of future events should be stored in chronological order in a calendar. Such a calendar is called a file. This file, along with other relavent information about the event also stores the types of events occurring at any such projected time.

As discussed above, "next event" simulation involves the portrayal of a system through time by examining the system at each event instant. A system is described by the entities (items) contained in the system and attributes of the entities. The entities may be associated with files (sets). Events can cause changes in the values of the attributes of the entities and/or the membership of entities in files. Thus, the portrayal of the system through time can be thought of as the status of the files containing the entities and the values of the attributes through time. Analysis of simulation involve the calculation of statistical properties of the files and the values of the attributes.

As demanded by GASP II, file 1 is the event file. When an order for an end-item (an entity) arrives, the attributes of the end-item are filed in the event file. At the same time various other informations of the end-item (attributes) are filed in file no. 2. Similarly the attributes of the files for machines are changed when a part is processed or queued. To affect this, the necessary attributes for each of the files are:

File 1 Attribute 1: Scheduled time of next event

Attribute 2: Event code

Arrival of order for end-item -- 1
(Subroutine ARRVL)

Release order to the shop -- 2
(Subroutine REPORT)

Netting with inventory -- 3
(Subroutine STOCK)

End of processing -- 4
(Subroutine PROCES)

Attribute 3: Arrival time of end-item (part)

Attribute 4 = 0

Attribute 5 = 0

Attribute 6 = End-item order number

Attribute 7 = 0

Attribute 8 = 0

Attribute 9 = Due date

Attribute 10= 0

File 2 Attribute 4 = Processing time per part

Attribute 5 = Part number

Attribute 7 = Quantity per part

Attribute 8 = Machine requirement

Attribute 10 = Inventory per part

Other attributes of the file are same as File 1.

File 3 to 7
(inclusive)

Attribute 4 = Processing time per part

Attribute 5 = Part number

Attribute 6 = 0

Attribute 7 = Quantity per part

Attribute 9 = Due date

Other attributes of the files are same as File 1.

<u>Ranking</u> File 1 and 2 are ranked on attribute 1 on the basis of FIFO (First-in-first-out). Ranking of other files depend on the due dates (attribute 9).

5.4 COLLECTION OF STATISTICS

5.41 Statistics related to products

Various statistics are necessary for a complete analysis of the system performance. The following statistical data resulting from the simulation are collected.

1. Average product flow time through the system.
2. Average product flow time through the job shop.
3. Average product flow time through the assembly shop.
4. Average departures from initial due date estimate.

We explain and justify that the collection of the above statistics is sufficient for our analysis:

1. Average product flow time ($\overline{PFT}$) is the sum of the sum of the average flow time through the job shop and the assembly shop. i.e.,

$$\overline{PFT} = \overline{PJS} + \overline{PAS}$$

In fact, when we collect statistics for the average flow-time through the job shop ($\overline{PJS}$) and the assembly shop ($\overline{PAS}$), this statistics is superfluous. But it is collected for checking the above relationship.

2. Average product flow-time through the job shop ($\overline{PJS}$) gives the measure of the job shop performance.
3. Average product flow-time through the assembly shop ($\overline{PAS}$) gives the average time each product remains in the assembly shop waiting for its component parts.
4. Average departure from the initial estimate of due-dates: gives the average departure from the due-dates, i.e.,--

$$\sum_{i=1}^{I} (EXCT_i - DDATE_i)/I$$

where, $EXCT_i$: Expected completion time of an end-item, i

$DDATE_i$: Initial estimate of due date for i

i : End-item number i = 1, 2, ... I.

Expected completion time for a product, i is determined in the following fashion.

When an order for an end-item (product) arrives, an expected completion time is calculated for each component part of the product, viz.,

$$E(CTIME_{ij}) = TIMCA_r + PTIME_{ij} \cdot A$$

where

$E(CTIME_{ij})$: Expected completion time of part , j and product, i

$TFMCA_r$: Time first machine is available in machine group, r

$PTIME_{ij}$: Processing time for part, j of product, i

A : Factor to take care of the waiting time.

Thus the expected completion time for the product, i is given by

$$\sum_{j=1}^{J} E(CTIME_{ij}).$$

5.42 Statistics related to machines

The utilization of each machine group is obtained by using the GASP provided subroutine TMSTAT. This subroutine collects statistical data based on observations made over time. The relative frequency in which a variable has a given value can be considered as the proportion of time that the variable has that value (15). The expected value of the variable would then be:

$$\text{Expected value} = \sum_x \frac{(\text{time variable has value } x)}{\text{total simulation time}}$$

$$= \left(\frac{1}{\text{total simulation time}}\right) \sum_x (\text{time variable has value } x)\ x$$

Therefore, by continuously monitoring the number of busy machines, we can obtain a measure of utilization.

In general, shop utilization is obtained as (19):

$$U_{shop} = \frac{1}{M} \sum_{r=1}^{M} \frac{1}{L_r} \sum_{k=1}^{L_r} \frac{TBUZ_k}{TTIME}$$

where,

M = total number of machine groups

r = machine group number, $r = 1, 2, \ldots M$

L_r = number of machines in machine group, r

$TBUZ_k$ = total busy time for machine k of machine group, r

$TTIME$ = total time period considered.

The mean values for each of the machine groups are calculated and they are monitored under "TIME GENERATED DATA" i.e., this corresponds to

$$\sum_{k=1}^{L_r} \frac{TBUZ_k}{TTIME}$$

From this we calculate the shop utilization by adding all of the mean values and dividing by the total number of machines in the shop.

CHAPTER VI

6.0 EXPERIMENTS AND RESULTS

6.1 CONDITIONS FOR EXPERIMENTAL SET-UP

Every run consisted of 400 end-item orders with inter-arrival intervals obtained (in advance in the end-item order generation) from an exponential distribution. The mean, of this exponential distribution was set to yield a utilization of 75% in the shop (and therefore 75% on each machine).

In each case, the shop was initially empty and no orders were released till the release date. At the end of each release date a batch of end-item orders were released to the shop. This technique itself accelerates the initial loading of the shop. Previous experiments by Conway (5) showed that under this condition stability is achieved very rapidly.

A simulation experiment is undertaken with an eye to obtaining information pertaining to some parameters of the given structure. Since computer simulation is an experimental procedure, one of the most important points is that the simulated system has attained its steady-state condition. In our case, the initial 100 orders were taken as required to set up the shop condition. Statistics were collected after the first 100 orders were completed and it was continued till the completion of 400 orders. Conway (5) had found that the shop settled down at a time concurrent with the arrival of 400 jobs. In our investigation, the completion of 400 jobs was roughly equivalent to the arrival of 30 orders.

During each run a set of orders were generated using pseudo-random numbers. This same set was presented to the shop to compare alternative release dates. Three such sets of orders were generated for the three runs. Table I gives the job-table for the three sets.

Random routing was used throughout the investigation. The number of machine groups was set at 5, each group having 2 machines. Though a larger shop with a greater number of machines may exhibit full shop complexity, it was not used for the sake of computer time (excution time of experiments would vary approximately as the square of the number of machines (5)).

For each run, end-item orders were analyzed i.e., their requirements of sub-parts and detail parts were determined. The part-structure (requirement of subparts and detail parts for an end-item) of end-items were generated with different seeds for pseudo-random generator. Thus end-item orders were made different from one another in a sense since they did not require the same subparts and detail parts in the same fashion. But the subparts and detail parts were unique i.e.,they had the same operation sheet throughout. Therefore the routing and processing time for each of the parts were generated once, at the beginning of each run.

Finally, these conditions were completely stationary--the same conditions and procedures were used throughout a run. Three such runs were made to investigate the alternative order releasing procedure. The following five order releasing dates were candidates for investigation:

1. 20 hours
2. 40 hours
3. 60 hours
4. 80 hours
5. 120 hours

6.1.1

The present analysis was restricted by memory storage and time limitations on the IBM 360/50 computer. Originally it was planned to use a shop of 10 machine groups and we wanted to keep one file for each of the 10 subparts and 10 detail parts. This required an approximate calendar size of 10 · 3000; and it used approximately 30-35 minutes of computer time for each run of simulation. To facilitate ease of computation, the problem had to be trimmed. We had to group the subparts and detail parts together and use a shop of 5 machine groups. This required an approximate calendar of 10 · 2000. The computing time also was greatly reduced. Each run now required approximately 20 minutes of computer time, so that we could make 3 runs for one release date within an approximate time of 60 minutes.

6.2 MEASUREMENT OF PERFORMANCE

Basically there were four measures of performance of interest. Two of these--the utilization of facilities, and the amount of work-in-process inventory--are essentially attributes of the shop. The other two--the total time in the shop, and the lateness with respect to assigned due dates--are attributes of the job passing through the shop. These were measured in the following manner.

6.21 Utilization

The utilization of each machine was obtained by means of the GASP provided subroutine TMSTAT, which collects statistics on the relative frequency with which a variable has a given value. Therefore, by continuously monitoring the number of busy machines in each machine group, we obtained a measure of utilization. Throughout our investigation we considered a shop with 5 machine groups, each group having 2 machines. The statistics generated by each of these 5 machine groups are listed under the heading "TIME GENERATED DATA" (Table II). The mean value of each of these machine groups are listed under the column heading "MEAN" in the same table. Total utilization for the shop is obtained by adding these "MEAN" utilizations for each of the 5 groups and dividing by the total number of machines in the shop. Total shop utilization based on the average of three runs are tabulated in Table III.

6.22 Work-in-process Inventory

An estimate of the work-in-process inventory is obtained by measuring the average number of jobs waiting in queue. The queue contents also vary from one service period to another under the same release rule. An average of these queue contents gives a measure of the total number of jobs waiting to be serviced. This queue statistics were obtained by using subroutine PRINTQ (provided in GASP II). Data were obtained for,

1. Average number of jobs in the queue with respect to time,
2. Maximum number of jobs in the queue,
3. Status of the queue at the end of service period.

Average work-in-process inventory for each release dates are tabulated in Table IV. A typical queue printout is included in Appendix A.

6.23 Total time in the shop

Total time in the shop for an order is calculated as the difference of departure time and arrival time. This was obtained by monitoring each arrival and departure time of orders by using GASP provided subroutine COLECT.

6.24 Lateness/earliness

When an order arrives, its due-date is determined as multiple of its total processing time (suggested by Conway (6)). A multiple of 9 times the total processing time gave the due-date for an order. Lateness is defined as the difference between the actual completion time of an order and its estimated due-date. A negative lateness indicates the order was completed early. By continuously monitoring the difference between the due-date and completion date, an estimate of deviation from the due dates is obtained. This is tabulated in Table V.

Figure 6.1 gives the effect of varying release date on the on-time completion of orders. Figures 6.2 and 6.3 give the queue content and total shop utilization, respectively for a release date.

TABLE I

DESCRIPTION OF ORDER GENERATING ROUTINE

Set No.	Routing	Processing Time Distribution	Interarrival Intervals	No. of Orders Processed	Seed
1	Random	Erlang (2, 2) .5-10.0 (μ= 1.0)	Erlang (1, 1) 8.0-20.0 ($\mu \pm$ 10.0)	400	13127 29245
2	Random	Erlang (2, 2) .5-10.0 (μ= 1.0)	Erlang (1, 1) 8.0-20.0 (μ = 10.0)	400	21725 31761
3	Random	Erlang (2, 2) .5-10.0 (μ = 1.0)	Erlang (1, 1) 8.0-20.0 (μ= 10.0)	400	91271 49781

TABLE II

TIME GENERATED DATA

CODE	MEAN	STD.DEV.	MIN.	MAX.	TOTAL TIME
1	1.75	0.73	1531.42	2.00	1531.420
2	1.52	0.64	1531.42	2.00	1531.420
3	1.53	0.73	1531.42	2.00	1531.420
4	1.68	0.76	1531.42	2.00	1531.420
5	1.49	0.71	1531.42	2.00	1531.420

*** TOTAL UTILIZATION= 79.71

SERVICE PERIOD (HRS.)= 60.00 RUN # 3

TABLE III

SHOP UTILIZATION

Release Interval (Hrs)	No. of Runs	Av. Utilization %
20	3	71.2
40	3	78.5
60	3	79.8
80	3	86.9
120	3	67.5

TABLE IV

NO. OF JOBS IN THE SHOP AT THE END OF THE SERVICE PERIOD

Service Period (Hrs)	No. of Jobs in Queue Mean	Variance
20	55	18
40	40	38
60	20	19
80	15	3
120	9	15

TABLE V

ON-TIME COMPLETION OF ORDERS

Release Interval (Hrs)	No. of runs	Lateness per Order		Earliness of Order	
		Mean	Variance	Mean	Variance
20	3	3.16	1.5		
40	3			1.74	.47
60	3			3.02	.31
80	3			.01	.12
120	3	6.15	2.11		

On-time Completion of Orders

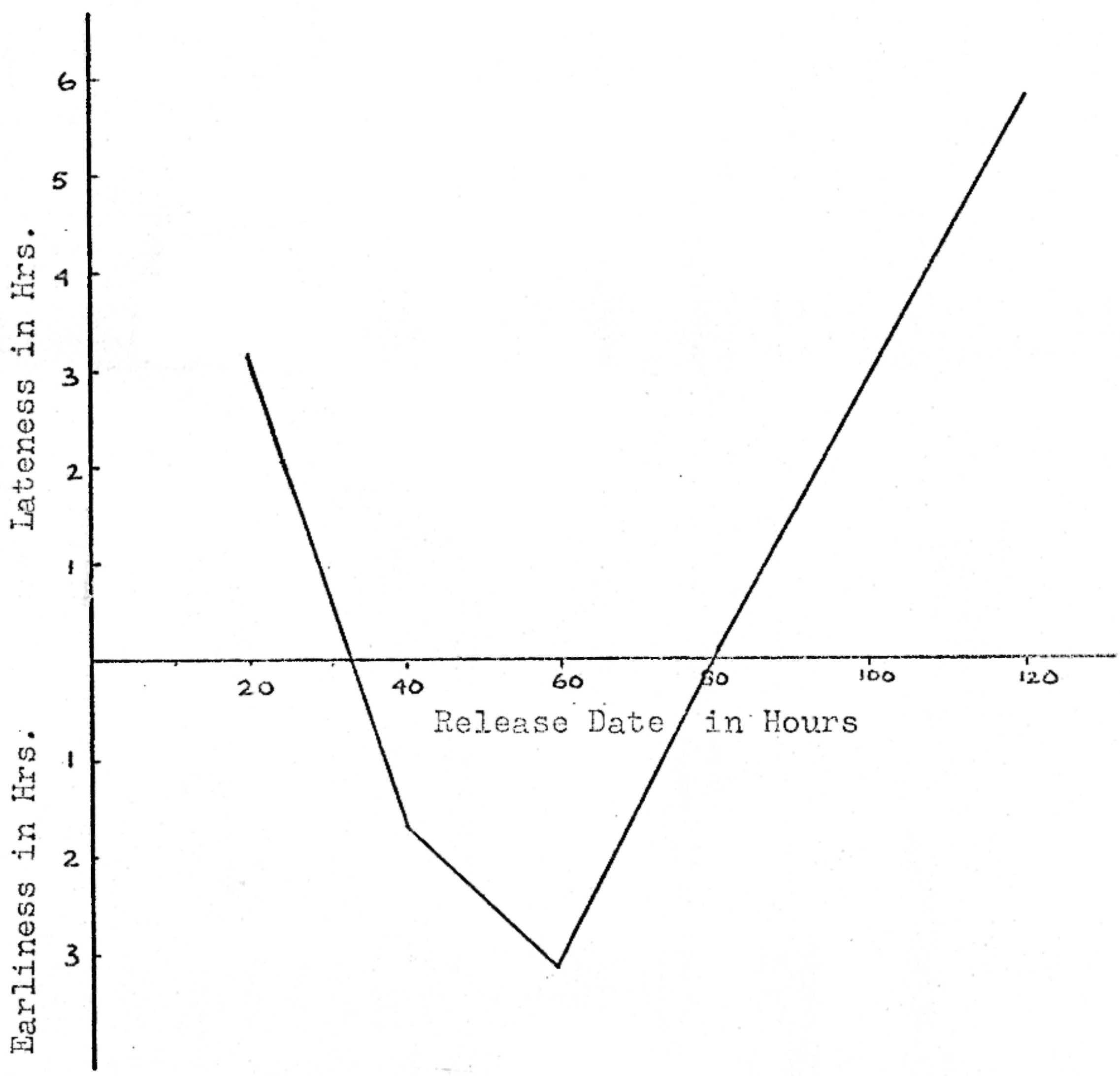
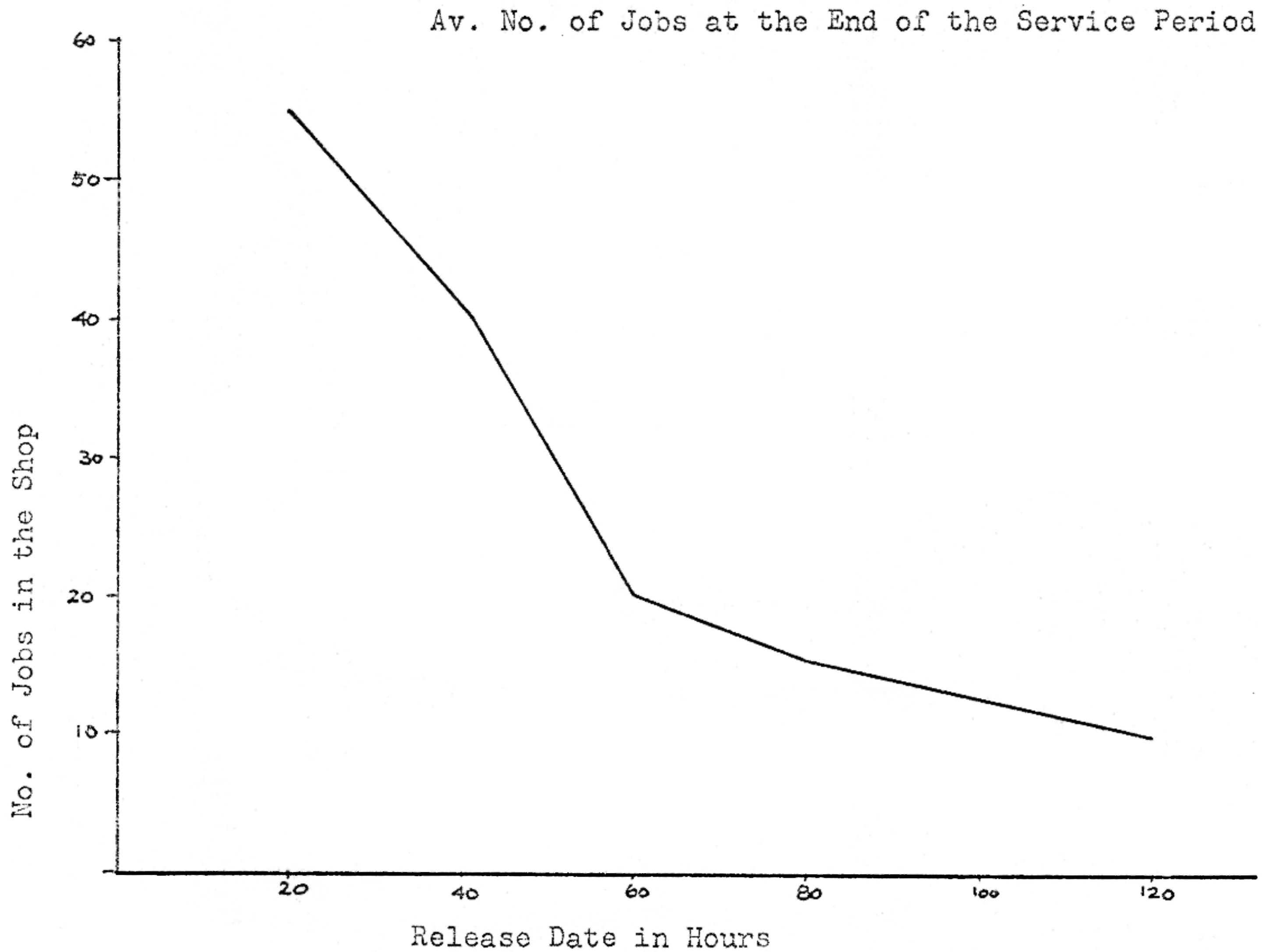


Fig. 6.1



Av. No. of Jobs at the End of the Service Period
60
50
40
30
20
10
No. of Jobs in the Shop
20
40
60
80
100
120
Release Date in Hours

Fig. 6.2

Shop Utilization

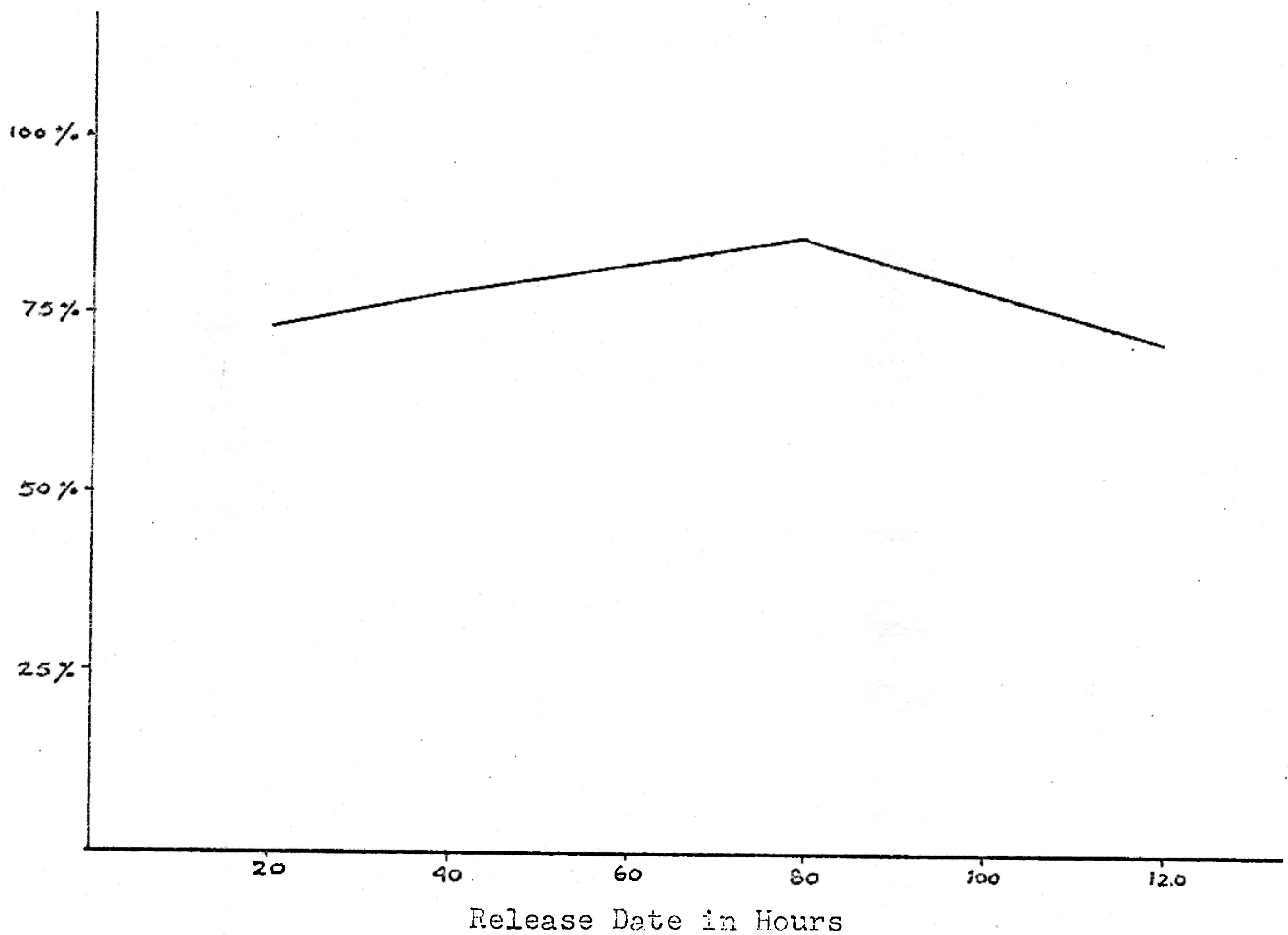


Fig. 6.3

CHAPTER VII

7.0 ANALYSIS AND DISCUSSION OF RESULTS

There were two principal areas of investigation. The first was the study of the on-time completion of orders. If orders are completed early, stockage of finished orders till the scheduled delivery date costs money. On the otherhand, late delivery is harmful to the goodwill of the company, often it results in making special arrangements for shipment. It is therefore desirable that orders be completed on time. The second major area of investigation was the minimization of the work-in-process inventory.

7.1 THE INVESTIGATION OF ON-TIME COMPLETION OF ORDERS

The job attributes of principal interest were the total amount of time an order spent in the shop and the lateness or earliness with reference to the due-date. We hypothesised in Chapter III that, under a certain methods of assigning due-dates, the length of order releasing date is important. If this release interval is short, there will be an accumulation of unfinished orders in the shop. Under a fixed processing capacity of the shop, the orders would be completed late. Again, when orders were released very late, they were completed late because of shorter available time between the release date and the scheduled due-date. Thus there seems to be an optimum release date. This was well observed in our investigation. This release date was found to be 80 hours. The average lateness per order was -.01 hour i.e., an average order was completed .01 hour earlier. For a release date of 20 hours the average lateness was found to be 3.16 hours and for a release date of 120 hours this figure was 6.15 hours.

7.2 MINIMIZATION OF WORK-IN-PROCESS INVENTORY

Under the same experimental set-up, the work-in-process inventory is a measure of congestion in the shop. If the order releasing period was short, a fresh batch of orders was released to the shop before the last batch was fully processed. The small size of orders provided little chance of combining the lower level parts. As the release interval is increased, lower level parts for end-items are accumulated before they are released to the shop. As a result fewer parts remained in an unfinished stage in the shop.

Of the many ways of measuring inventory attention was focused on the number of jobs rather than some measure of the work content. This was done for theoretical rather than practical reason. In a manufacturing shop reduction of the number of jobs in process is desirable, but reduction of the dollar value of inventory is often more important, and in most cases this is more highly correlated with work content, than simple job count. In our investigation for an assembly shop we were making a number of end-items from many similar sub-assembly and detail parts. Here the appropriate objective was minimization of the number of end-item orders awaiting completion. In this case the number of jobs in the shop was a direct and important measure of inventory.

CHAPTER VIII

8.0 SUMMARY AND CONCLUSION

The objective of this research was to investigate the effects of varying release dates of orders on the overall system performance. This effect was studied under one priority dispatching rule. The rule was "among a number of jobs waiting in queue, process the job which has the earliest due date," and in case of ties "chose the job with the shortest processing time." We arrived at a release date of 80 hours. The system performance was found optimum at this date. The validity of this result is restricted to the assumption of arrival rate and an exponential pattern of orders. The processing time also was assumed exponential. The interaction of these arrival and processing rates coupled with the number of jobs per order gave a level of shop utilization. The results of this study are good for this level of utilization (75%). For some other level of utilization this release date may vary. A general inference may be that, if the shop is operating at a higher level of utilization, it may be found proper to release orders early. However, the behavior of release date with reference to shop utilization remains to be seen.

A second important criterion was the choice of a priority dispatching rule. The results of this study may be different for different priority rules. Though the earliest due date and shortest operation rule has performed well in many situations (6), its superiority over other rules is not unchallanged. Moreover, this rule has no lood-ahead feature. This, coupled with a short planning horizon made the results of the present investigation applicable to a restricted time period. This results may be applicable to a small company with short planning horizons or to maintenance problems.

To make the results of this study applicable to a wider planning period a forecast could be made by extrapolating the past data on net orders received, using the following method described by Brown (24).

$$\hat{S}(t+\tau) = \bar{S}(t) + \bar{b}(t)\left[1/\alpha + \tau - 1\right],\quad 0 < \alpha < 1,$$

where $\bar{S}(t)$ and $\bar{b}(t)$ are estimates at time t of the average of net orders received and the trend in these orders, respectively. $\bar{S}(t)$ and $\bar{b}(t)$ are obtained from the following relations:

$$\bar{S}(t) = \alpha S(t) + (1-\alpha)\,\bar{S}(t-1), \quad \text{and}$$
$$b(t) = \alpha\left[S(t) - \bar{S}(t-1)\right] + (1-\alpha)\,\bar{b}(t-1)$$

where $S(t)$ is the number of net orders actually received in period t. The value of α will depend upon the forecasting period. A value of $\alpha = 0.1$ has found to give minimum forecasting error, when forecasting one month into the future (3). Once $\hat{S}(t+\tau)$, the forecast of net orders received at a time units in the future is obtained, an appropriate release dates can be determined.

FUTURE RESEARCH

Based on the experience gained from this investigation, possibilities for future research include:

1. Study the effects of various interacting parameters, viz., the arrival and processing pattern and rate in determining release date.
2. Determining the optimal quantity of each part to be released and study the variation in these quantities from the optimal number of end-item orders.
3. Performance of GASP II language in solving large scale simulation problems does not seem satisfactory in referance to computer time. It is proposed to use other simulation language viz., GPSS and SIMSCRIPT and compare the language performance with respect to,
   (1) computer time, and
   (2) analysis of system performance.
4. Study the system with queue order to determine the best due date not set by the sales department.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his Major Professor, Dr. L. E. Grosh, for his guidance, helpful assistance and counsel during the preparation of this thesis.

REFERENCES

1. S. S. Ackerman, "Evenflow, a scheduling method for reducing lateness in job shops," Management Technology, vol. 3, No. 1, May 1963.

2. C. T. Baker, "Fabrication and assembly operations. Part I: The outlines of a control system." IBM Systems Journal, Vol. 4, No. 2, 1965, 86-93.

3. A. B. Calica, "Fabrication and assembly operations. Part II: Long range planning techniques," IBM Systems Journal, vol. 4, No. 2, 1965, 94-104.

4. F. L. Church, "Requirements generation, explosions and bills of material," IBM Systems Journal, September-December, 1963, 268-287.

5. R. W. Conway, W. L. Maxwell, "Priority dispatching and work-in-process inventory in a job shop," Journal of Industrial Engineering, Vol. XVL, No. 2, 1965.

6. R. W. Conway, W. L. Maxwell, L. W. Miller, "Theory of Scheduling," Addison Wesley Publishing Co., Reading, Massachussetts, 1967.

7. B. P. Dzielinski, C. T. Baker, and A. S. Manne, "Simulation tests of lot size programming," Management Science, vol. 9, No. 2, Jan. 1963, 229-257.

8. S. E. Elmaghraby, "An algebra for the analysis of generalized activity networks," Management Sciences, vol. 10, No. 3, April, 1964.

9. B. Giffler, "Mathematical solution of explosion and scheduling problems," IBM Research Report -RC-128.

10. B. Giffler, "Mathematical solution of production planning and scheduling problems, IBM Technical Report, 90-01028-026, 2nd printing, November, 1961.

11. I. J. Lieberman, "A mathematical model for integrated business system." Management Science, Vol. 2, No. 8, July 1956, 327-336.

12. P. G. Loewner, "Fabrication and assembly operations. Part III: Matrix methods for processing configuration data," IBM Systems Journal, vol. 4, No. 2, 1965.

13. A. S. Manne, "Programming of economic lot sizes," Management Science, vol. 4, No. 2, January, 1958, 115-135.

14. J. H. Mize and J. G. Cox, Essentials of Simulation, Prentice-Hall, Inc. Englewood, Cliffs, New Jersey, 1968.

15. A. A. B. Pritsker and Philip J. Kivial, "GASP II: A Fortran based simulation language, Dept. of Industrial Engineering, Arizona State University, September 1967.

16. Alan J. Rowe, "Sequential decision rules on production scheduling, Ph.D thesis, University of Los Angeles, California, 1958.

17. M. K. Starr and D. W. Miller, Inventory Control: Theory and Practice, Prentice Hall, Inc., 1962.

18. Alan W. Steinberg, "Some notes on the similarity of three management science models and their analysis by connectivity matrix techniques," Management Science, January 1963, vol. 9, No. 2.

19. William J. Thompson, The effect of priority dispatching rules on an assembly job-shop, M.S. report, Arizona State University, January 1968.

20. A.Vazsonyi, "Operations research in production control--a progress repot," Journal of the O.R. Society of America, vol. 4, 1956.

21. A. Vazsonyi, "The use of mathematics in production and inventory control," Management Science, Vol. 1, No. 1, October 1954, 70-85.

22. A. Vazsonyi, "Economic lot size formulas in manufacturing," Operations Research, vol. 5, No. 1, February, 1957.

23. A. Vazsonyi, "Scientific programming in business and industry, John Wiley and Sons, 1958.

24. R. G. Brown, Statistical forecasting for Inventory Control, McGraw-Hill Book Co., New York, 1959.

COMPUTER FLOW-DIAGRAM

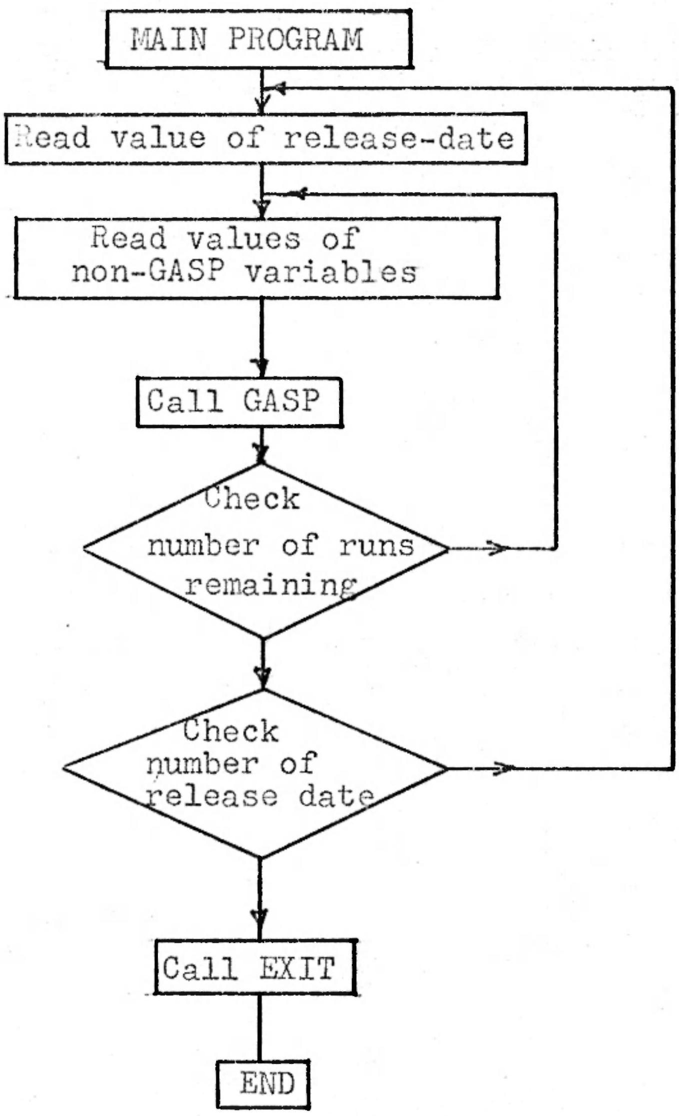


Fig. A-1 MAIN PROGRAM

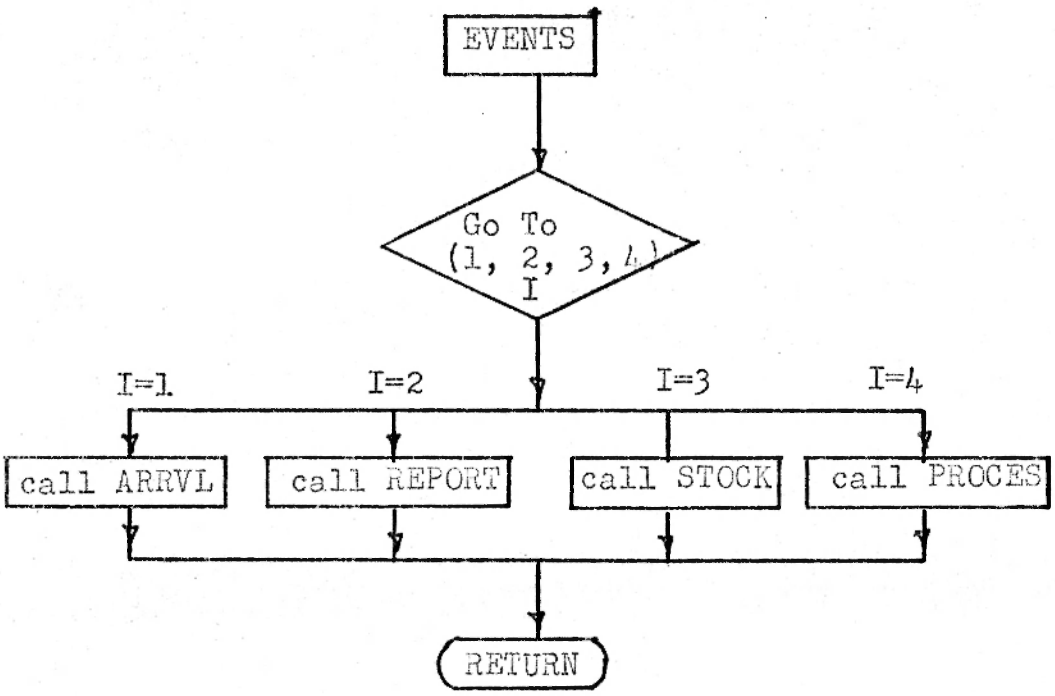


Fig. A-2 Subroutine EVENTS

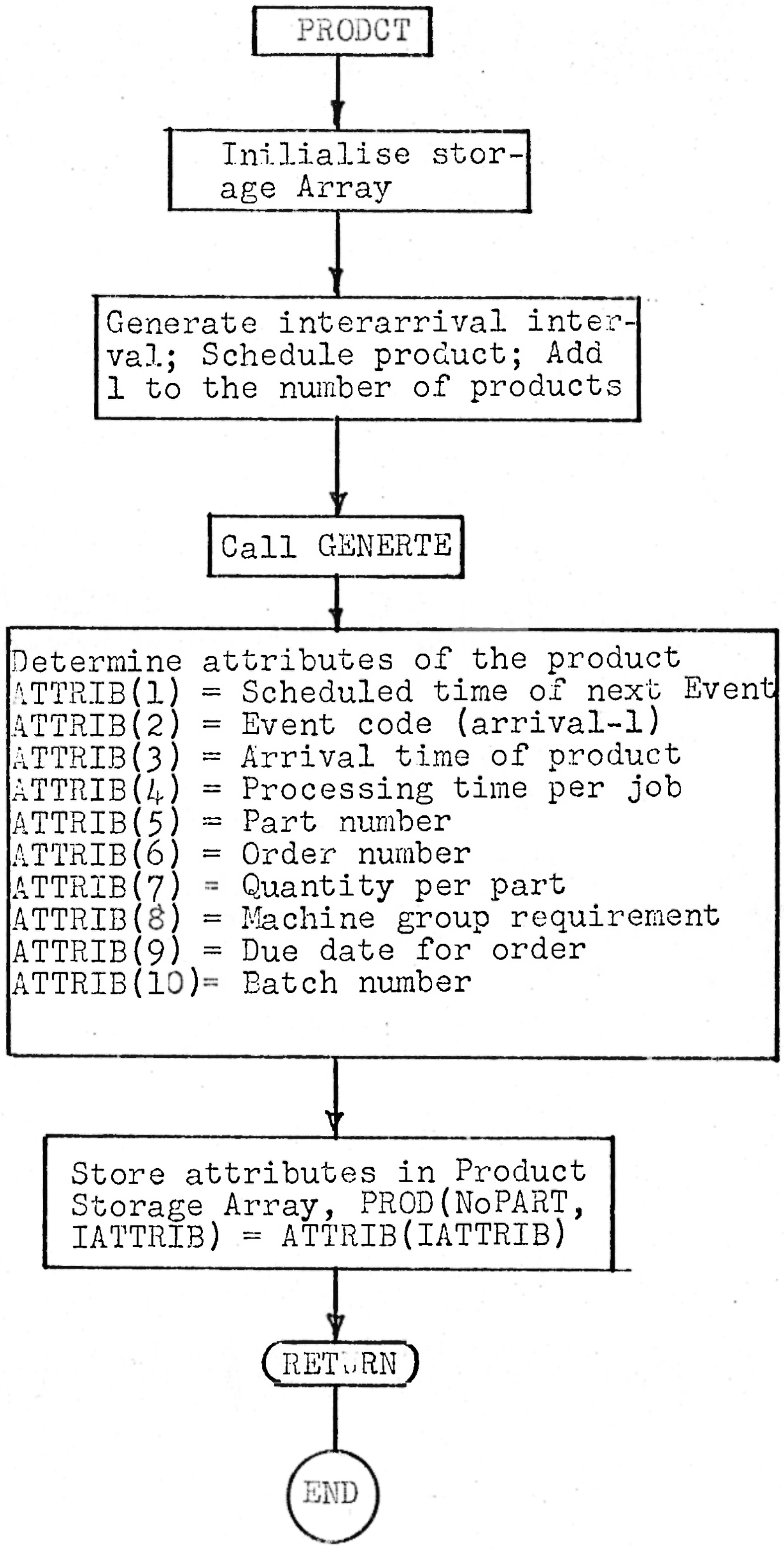


Fig. A-3 Subroutine PRODCT

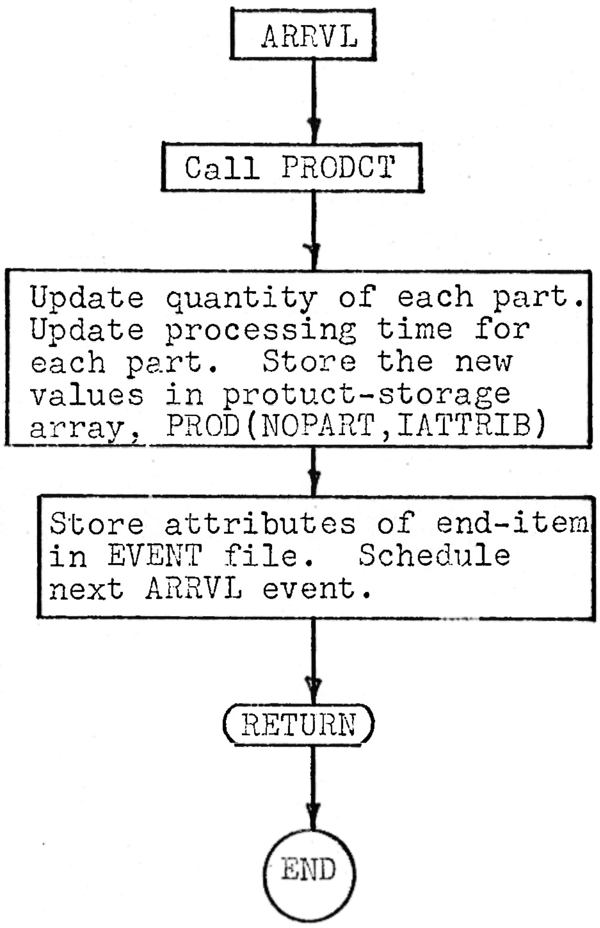


Fig. A-4 Subroutine ARRVL

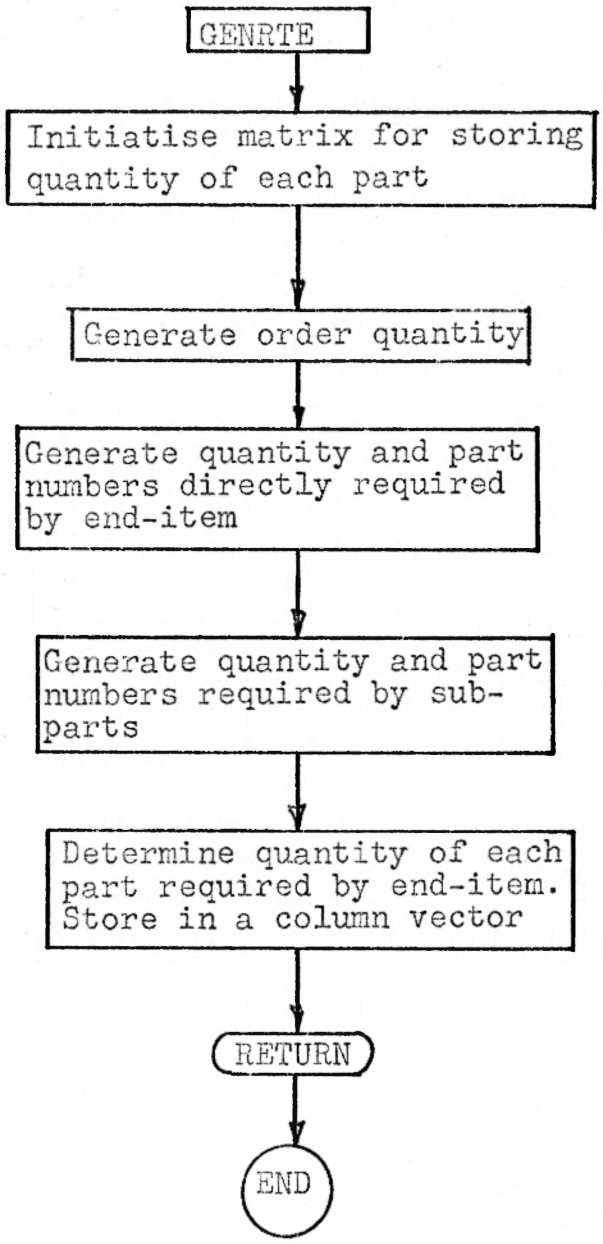


Fig. A-5 Subroutine GENRTE

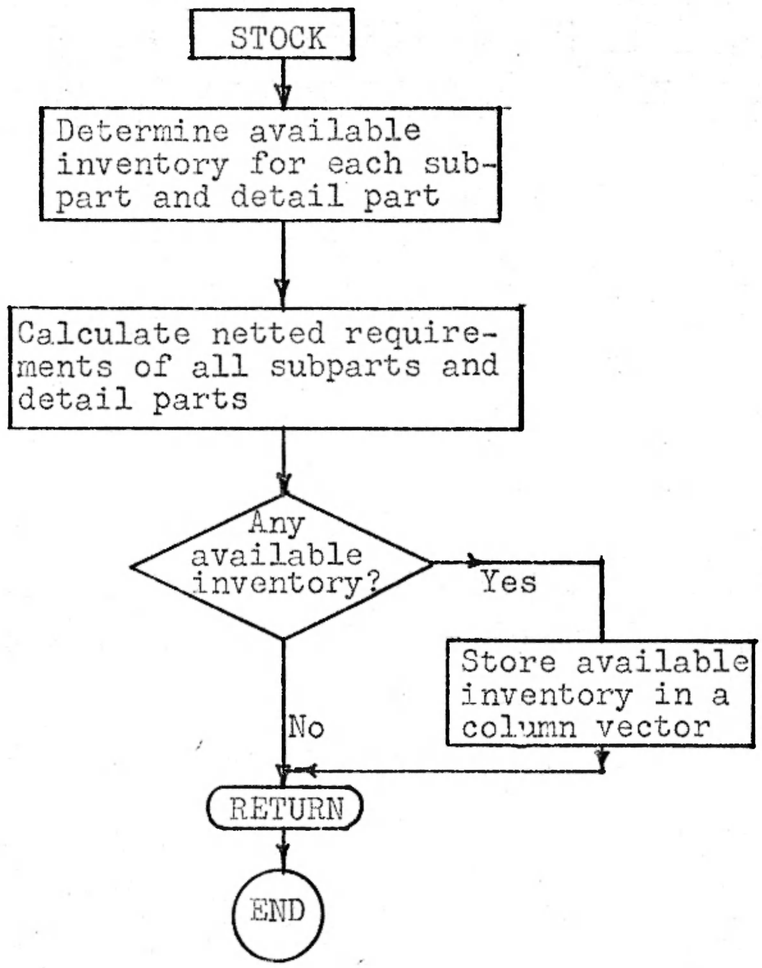


Fig. A-6 Subroutine STOCK

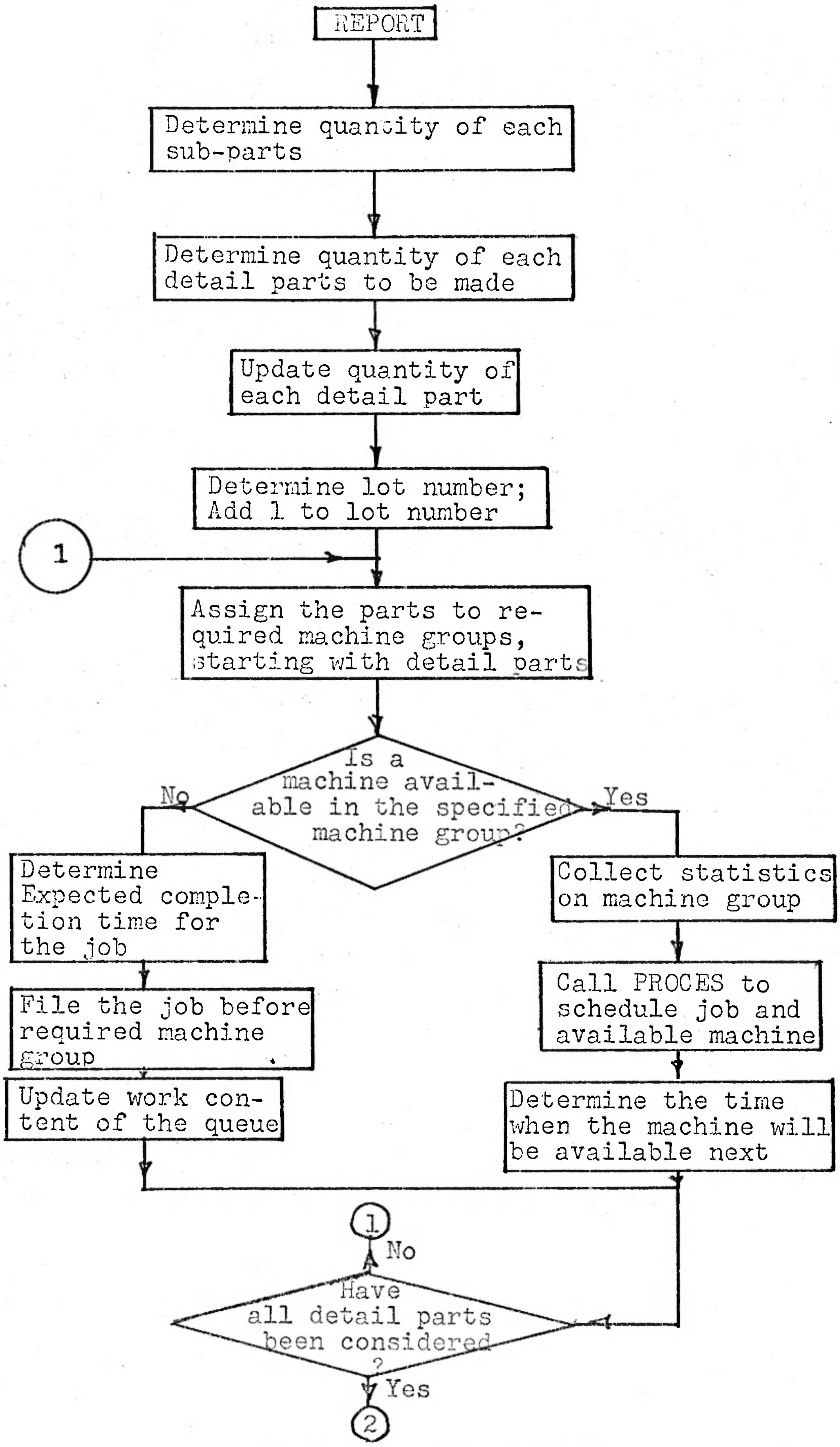


Fig. A-7 Subroutine REPORT

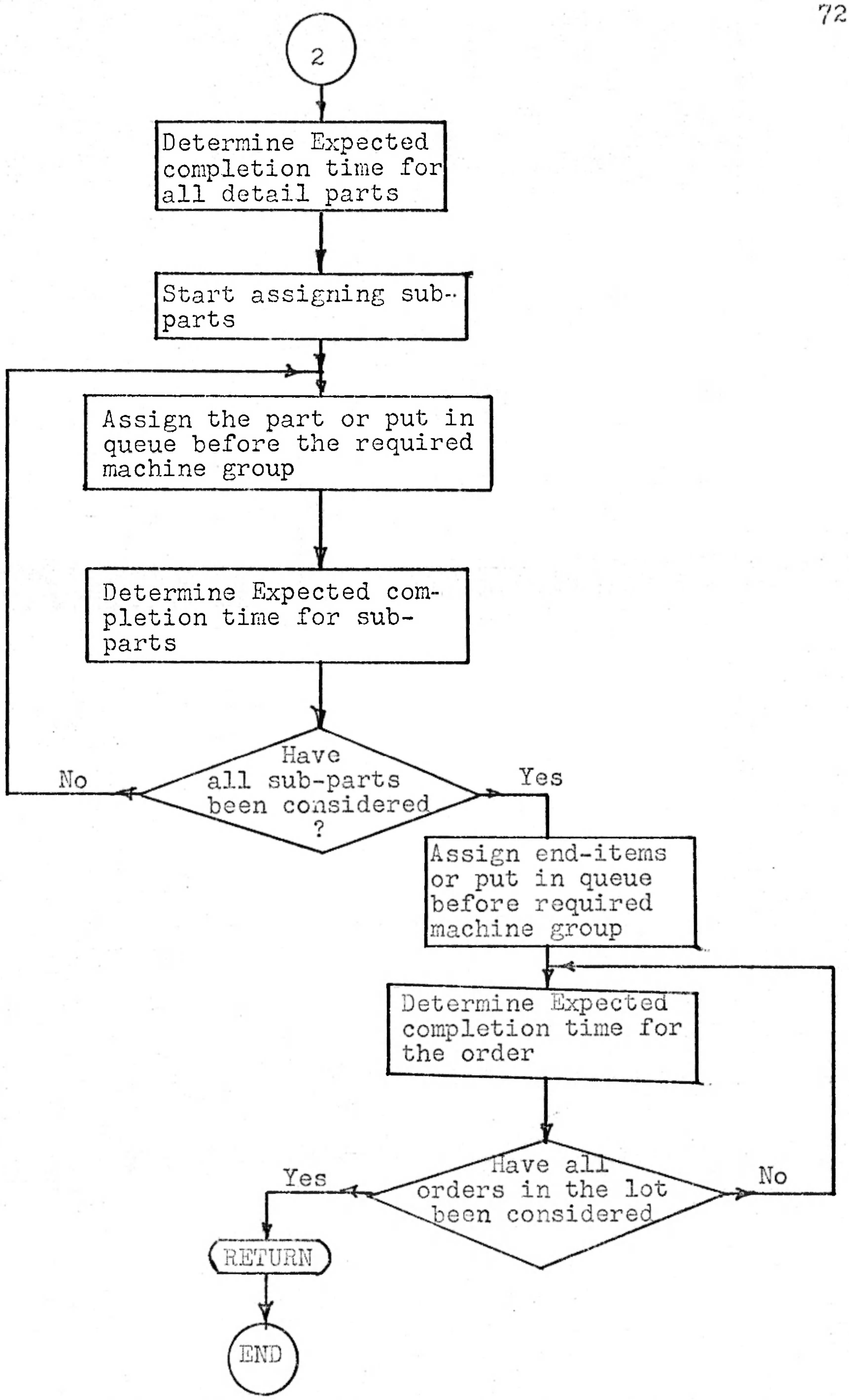


Fig. A-7 Subroutine REPORT (Contd.)

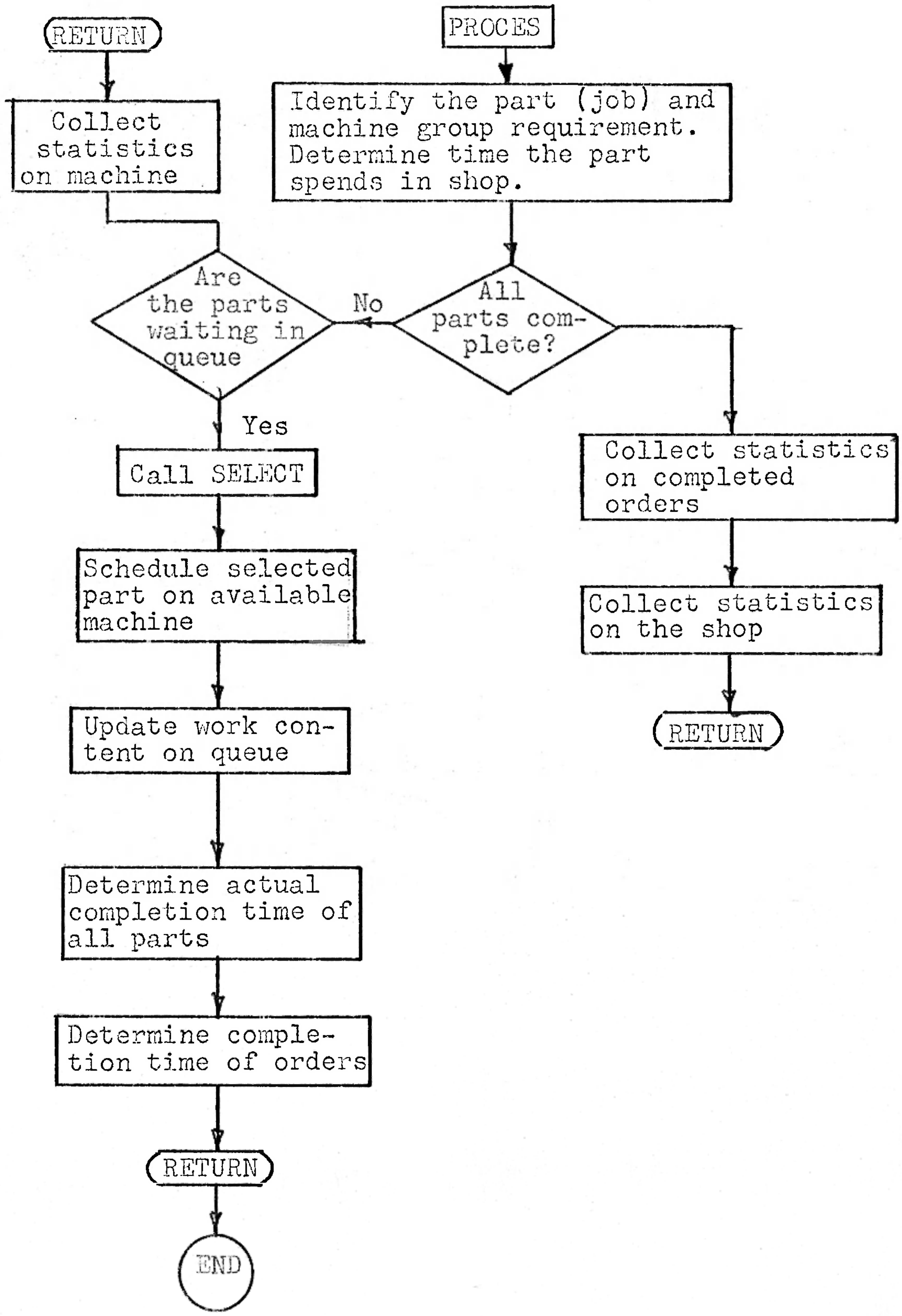


Fig. A-8 Subroutine PROCES

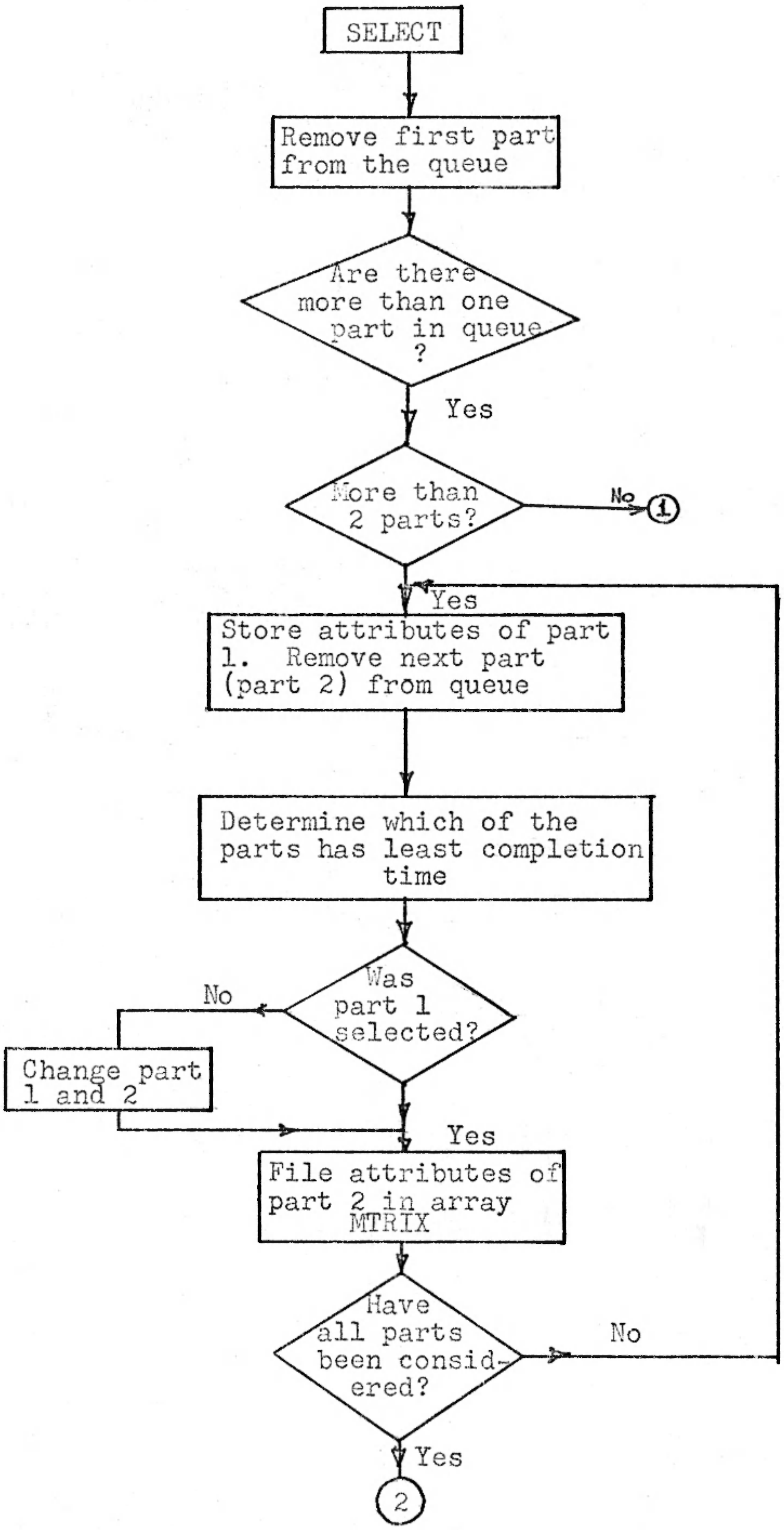
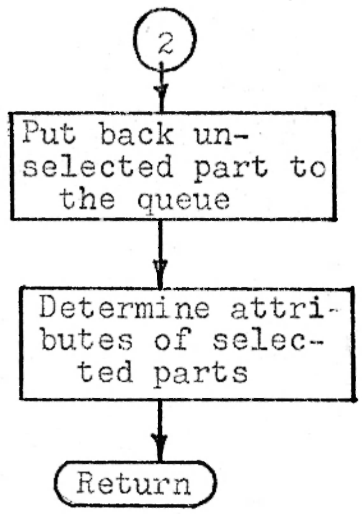


Fig. A-9 Subroutine SELECT



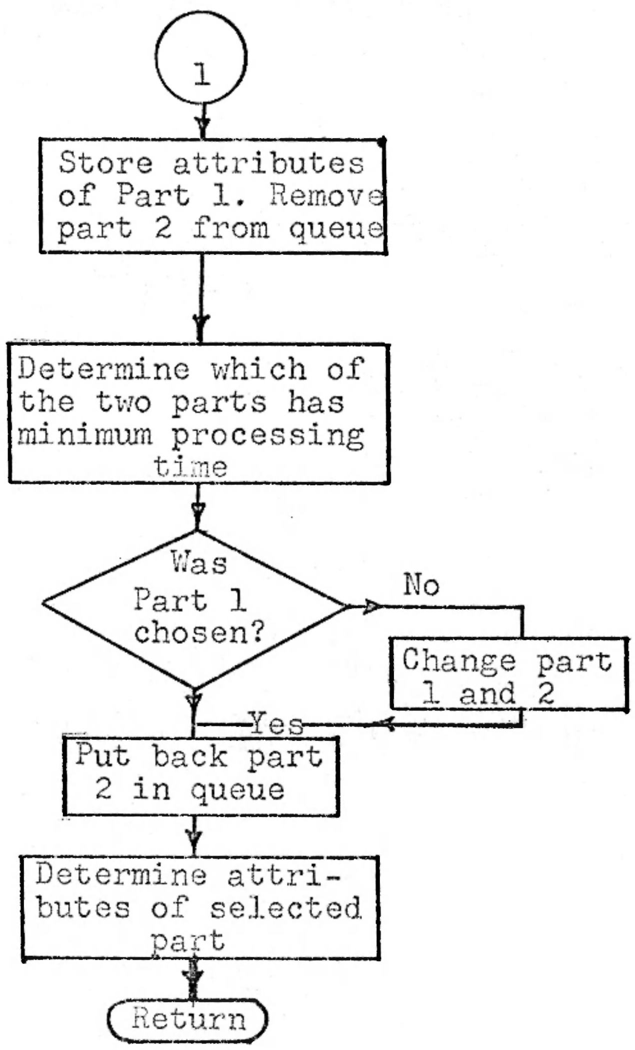


Fig. A-9 Subroutine SELECT (Contd).

SIMULATION OUTPUT SAMPLES

GASP SUMMARY REPORT

GENERATED DATA

CODE	MEAN	STD.DEV.	MIN	MAX.	OBS.
1	155.15	13.36	23.35	199.98	400
2	76.19	26.36	31.23	123.68	400
3	231.23	13.35	98.36	251.63	400
4	-0.45	6.36	-1.23	6.11	400

TIME GENERATED DATA

CODE	MEAN	STD.DEV.	MIN.	MAX.	TOTAL TIME
1	1.93	0.83	1601.19	2.00	1601.150
2	1.78	0.72	1601.19	2.00	1601.150
3	1.36	0.56	1601.19	2.00	1601.150
4	1.76	0.80	1601.19	2.00	1601.150
5	1.89	0.73	1601.19	2.00	1601.150

*** TOTAL UTILIZATION= 87.22

SERVICE PERIOD (HRS.)= 80.00 RUN # 1

QUEUE PRINTOUT, QUEUE NO. 1

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 10.558

MAXIMUM 13

QUEUE CONTENTS

1959.0298	1.0000	1959.0298	454.0000	10.0000	162.0000	454.0000	3.0000	2110.9500	0.0
1960.0000	3.0000	1938.8799	12.2400	10.0000	160.0000	0.0	1.0000	1980.7500	0.0
1962.7200	4.0000	1100.0000	70.0000	15.0000	0.0	41.7900	4.0000	1306.2500	12.0000
1966.3799	4.0000	940.0000	69.0000	18.0000	0.0	69.0000	4.0000	1138.0000	11.0000
1996.2500	4.0000	1820.0000	47.2000	15.0000	0.0	47.2000	2.0000	2027.8298	15.0000
2002.8198	4.0000	1180.0000	89.5100	17.0000	0.0	89.5100	1.0000	1386.2500	9.0000
2007.3599	4.0000	1900.0000	63.2600	15.0000	0.0	33.2300	3.0000	2104.9700	10.0000
2020.0000	2.0000	1900.0000	22.1200	20.0000	0.0	22.1200	1.0000	2108.8799	10.0000
2026.9600	4.0000	1820.0000	106.7600	16.0000	0.0	106.7600	5.0000	2027.8298	15.0000
2037.0000	4.0000	1580.0000	417.0000	12.0000	0.0	417.0000	2.0000	1783.7500	14.0000
2064.9199	4.0000	1740.0000	270.0000	10.0000	0.0	270.0000	1.0000	1937.3098	14.0000
2214.5698	4.0000	940.0000	543.0000	12.0000	0.0	543.0000	3.0000	1060.5000	11.0000

QUEUE PRINTOUT, QUEUE NO. 2

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 3.563

MAXIMUM 10

QUEUE CONTENTS

740.0000	1.0000	700.0000	441.0000	10.0000	0.0	348.0000	1.0000	832.5000	9.0000
1060.0000	1.0000	1020.0000	723.0000	12.0000	0.0	723.0000	1.0000	1222.0598	8.0000
1220.0000	1.0000	1180.0000	416.0000	10.0000	0.0	416.0000	1.0000	1359.5298	9.0000
1220.0000	1.0000	1180.0000	124.2900	15.0000	0.0	124.2900	1.0000	1386.2500	9.0000
1380.0000	1.0000	1340.0000	531.2500	12.0000	0.0	472.0000	1.0000	1540.7500	12.0000
1540.0000	1.0000	1500.0000	631.0000	12.0000	0.0	631.0000	1.0000	1677.1099	13.0000
1940.0000	1.0000	1900.0000	291.0000	12.0000	0.0	291.0000	1.0000	2104.9800	10.0000
1940.0000	1.0000	1900.0000	56.0500	17.0000	0.0	56.0500	1.0000	2104.9800	10.0000
1940.0000	1.0000	1900.0000	19.4400	19.0000	0.0	19.4400	1.0000	2104.9800	10.0000
1940.0000	1.0000	1900.0000	22.1200	20.0000	0.0	22.1200	1.0000	2108.8799	10.0000

QUEUE PRINTOUT, QUEUE NO. 3

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 1.299

MAXIMUM 10

QUEUE CONTENTS

1620.0000	1.0000	1580.0000	159.8500	16.0000	0.0	79.8900	2.0000	1783.7500	14.0000
1620.0000	1.0000	1580.0000	26.5800	17.0000	0.0	26.5800	2.0000	1783.7500	14.0000
1620.0000	1.0000	1580.0000	17.5000	19.0000	0.0	17.5000	2.0000	1783.7500	14.0000
1700.0000	1.0000	1660.0000	774.7598	12.0000	0.0	657.0000	2.0000	1862.5598	15.0000
1700.0000	1.0000	1660.0000	45.6400	15.0000	0.0	22.7900	2.0000	1862.5598	15.0000
1860.0000	1.0000	1820.0000	174.0000	10.0000	0.0	87.0000	2.0000	2027.8298	15.0000
1860.0000	1.0000	1820.0000	522.0000	12.0000	0.0	522.0000	2.0000	2027.8298	15.0000
1940.0000	1.0000	1900.0000	501.0000	10.0000	0.0	501.0000	2.0000	2104.9700	10.0000

QUEUE PRINTOUT, QUEUE NO. 4

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 2.233

MAXIMUM 7

QUEUE CONTENTS

1380.0000	1.0000	1340.0000	614.0000	10.0000	0.0	307.0000	3.0000	1540.7500	12.0000
1700.0000	1.0000	1660.0000	37.9200	19.0000	0.0	37.9200	3.0000	1862.6099	15.0000
1700.0000	1.0000	1660.0000	40.6500	20.0000	0.0	40.6500	3.0000	1865.4700	15.0000
1780.0000	1.0000	1740.0000	778.0000	12.0000	0.0	389.0000	3.0000	1937.3098	14.0000
1780.0000	1.0000	1740.0000	21.2600	19.0000	0.0	21.2600	3.0000	1941.3098	14.0000
1860.0000	1.0000	1820.0000	22.0000	20.0000	0.0	22.0000	3.0000	2028.8699	15.0000

QUEUE PRINTOUT, QUEUE NO. 5

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 5.949

MAXIMUM 17

QUEUE CONTENTS

1060.0000	1.0000	1020.0000	1228.0000	10.0000	0.0	1228.0000	4.0000	741.7500	8.0000
820.0000	1.0000	780.0000	353.0000	12.0000	0.0	353.0000	4.0000	984.0298	9.0000
980.0000	1.0000	940.0000	542.0000	10.0000	0.0	542.0000	4.0000	1060.5000	11.0000
900.0000	1.0000	860.0000	654.0000	10.0000	0.0	654.0000	4.0000	1062.5398	10.0000
900.0000	1.0000	860.0000	516.7500	12.0000	0.0	333.0000	4.0000	1062.5398	10.0000
980.0000	1.0000	940.0000	45.5000	15.0000	0.0	45.5000	4.0000	1138.0000	11.0000
1940.0000	1.0000	1900.0000	91.0500	16.0000	0.0	81.1700	4.0000	2104.9800	10.0000
1940.0000	1.0000	1900.0000	58.1400	18.0000	0.0	58.1400	4.0000	2104.9800	10.0000

QUEUE PRINTOUT, QUEUE NO. 6

AVERAGE NO. OF ITEMS IN THE QUEUE WAS 3.384

MAXIMUM 13

QUEUE CONTENTS

1300.0000	1.0000	1260.0000	250.0000	12.0000	0.0	125.0000	5.0000	1424.1599	11.0000
1460.0000	1.0000	1420.0000	999.7898	10.0000	0.0	963.0000	5.0000	1620.2000	9.0000
1460.0000	1.0000	1420.0000	83.7500	15.0000	0.0	41.8500	5.0000	1627.3599	9.0000
1460.0000	1.0000	1420.0000	126.7900	17.0000	0.0	63.3800	5.0000	1627.3599	9.0000
1460.0000	1.0000	1420.0000	100.1300	18.0000	0.0	100.1300	5.0000	1627.3599	9.0000
1460.0000	1.0000	1420.0000	19.0600	20.0000	0.0	18.8500	5.0000	1628.4600	9.0000
1540.0000	1.0000	1500.0000	46.2500	17.0000	0.0	46.2500	5.0000	1707.1299	13.0000
1700.0000	1.0000	1660.0000	421.8499	10.0000	0.0	337.0000	5.0000	1862.5898	15.0000
1700.0000	1.0000	1660.0000	119.5800	18.0000	0.0	59.7700	5.0000	1862.5898	15.0000
1780.0000	1.0000	1740.0000	82.9400	18.0000	0.0	82.9400	5.0000	1941.2898	14.0000

COMPUTER PROGRAM

```
DESCRIPTION OF VARIABLES
**************************************************

LI= LEVEL 1 PART#
L21- L22= LEVEL 2 PART#
L31-L32= LEVEL 3 PART#
NSUMP1= NO OF JOBS FOR INITIAL RUN
NSUMP2= NO OF JCBS IN THE SAMPLE
MCG= MACHINE GR. #;1 -5
NMCEG= NO OF MACHINES IN EACH MACHINE GR.
NOPART= PART#
PROD(I,J)= PRODUCT STORAGE ARRAY FOR ANY PRODUCT I
MACH(I)= NO OF MACHINES IN MACHINE GR.I
MPROD= PRODUCT #
NPRDCT= NO OF ORDERS DURING SERVICE PERIOD
NPLATE= NO OF ORDERS LATE
ASM= SERVICE INTERVAL
PRDUCE= FACTOR TO MULTIPLY PROCESSING TIME FOR ESTIMATING DUE-DATE
JKR= RUN #
BN(I,J)= REQUIREMENT MATRIX FOR I TH PRODUCT
ANX(NOPART)= ARRAY FOR STORING QUANTITY OF NOPART
JBPRT(LEV)= NO OF JOBS IN ANY LEVEL PART
IST(NOPART)= ON-HAND INVENTORY OF NOPART
ISTF(NOPART)= INVENTORY OF NOPART CARRIED OVER TO THE NEXT SERVICE PERIOD
```

```
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMCN ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMCN AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMCN ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPROCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,2500)
      JKR=1
1     READ 600,L1,L21,L22,L31,L32,MCG,NMCEG
      READ 600,NSUMP1,NSUMP2
      DO 100 I= 1,L1
      ISTF(I)=0
      IST(I)=0
100   ANX(I)= 0.
       DO 201 I= 1,MCG
      MACH(I)= NMCEG
      NAVL(I)= NMCEG
      DO 202 J= 1,4
202   SAV(J,I)= 0.
201   CONTINUE
      DO 702 J= L31,L1
      DO 702 I= 1,10
702   PROD(J,I)= 0.
      MPROD=0
      NPROCT= 0
      KTPRD= 0
      NPLATE= 0
      ST=0.
      CT=0.
      IKTT=0
      READ 600,IX1,IX2,IX3,IX4,IX5,IX6,IX7
600   FORMAT(7I6)
      READ 1969,ASM
      READ 1969,PRDUCE
1969  FORMAT(5X,F5.1)
      CALL GASP(NSET)
      IF(JKR .EQ.3) GO TO  601
      JKR= JKR+1
      GO TO 1
601   CALL EXIT
      END
```

```
  SUBROUTINE PRODCT(NSET)
  PURPOSE :: THIS SUBROUTINE GENERATES THE ATTRIBUTES OF THE END--
  ITEM ORDER; IT CALLS SUBROUTINE GENRTE TO DETERMINE THE PART--
  STRUCTURE OF THE END-ITEM
  COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
 1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
 2OP,MXX
  COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
 1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
 25),IX(8),SUMA(25,5),ATTRIB(10)
  COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
 1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
 240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IO(40)
 3,LO(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
 44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
 525),IJOB(40,20),TSHOP(40)
  COMMON ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
 1,NOPART,NPROCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
 2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
  DIMENSION NSET(12,1)
C ATTRIB(1)= SCHEDULED TIME OF NEXT EVENT
C ATTRIB(2)= EVENT CODE(1-ARRIVAL;2-REPORT;3-STOCK;4-PROCES(END OF SERVICE))
C ATTRIB(3)= ARRIVAL TIME OF PRODUCT
C ATTRIB(4)= PROCESSING TIME /PART
C ATTRIB(5)= PART #
C ATTRIB(6)= PRODUCT #
C ATTRIB(7)= QTY/PART
C ATTRIB(8)= MACHINE GR. REQUIREMENT
C ATTRIB(9)= DUE DATE
C ATTRIB(10)= LOT #
  TTIME1=0.
  TTIME2= 0.
  TTIME3=0.
  AT= ERLANG(1,1)
  ATTRIB(1)= AT+ST
  ATTRIB(3)= ATTRIB(1)
  ST= ATTRIB(1)
  ATTRIB(2)=1.
   MPROD= MPROD+1
  IX1= IX1 + 2
  IX2= IX2+ 4
  IX3= IX3+ 4
  IX7= IX7+ 4
  CALL GENRTE(NSET)
  DO 101 K= L31,L1
  PROD(K,5)= K
  ATIME(K)= ERLANG(2,2)
  PTIME(K)= ATIME(K)*AN(K)
  PROD(K,4)= PTIME(K)
  PROD(K,6)= MPROD
```

```
      PROD(K,8)= MG(K)
      NG= PROD(K,8)
      DO 199 J=1,3
101   CONTINUE
      IF(AN(L1).LE.0.) GO TO 177
      TTIME1= AN(L1)*ATIME(L1)
199   PROD(K,J)= ATTRIB(J)
177   DO 102 K= L21,L22
      IF(AN(K).LE.0.) GO TO 178
102   TTIME2= 2.*AN(K)*ATIME(K) + TTIME2
178   DO 103 K= L31,L32
      IF(AN(K))103,103,180
180   TTIME3= 4.*AN(K)*ATIME(K) + TTIME3
103   CONTINUE
      DO 700 KKK= L31,L1
      IF(KKK.GE.L1) GO TO 310
      IF(KKK.GE.L21.AND.KKK.LE.L22) GO TO 320
      DDATE= ATTRIB(1) + 160. -TTIME1 -TTIME2
      GO TO 450
310   DDATE= ATTRIB(1) + 160.
      GO TO 450
320   DDATE= ATTRIB(1) +160.- TTIME1
450   PROD(KKK,9)= DDATE
      PROD(KKK,10)=0.
700   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GENRTE(NSET)
      THIS SUBROUTINE GENERATES PART-STRUCTURE OF THE END-ITEM
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IO(40)
     3,LQ(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMON ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPRDCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      DO 800 I= L31,L1
      DO 800 J=1,6
  800 BN(I,J)=0.
      BN(L1,1)= RANDU(IX1)*5.+ 2.
      J1= RANDU(IX7)*2+5
      I1= RANDU(IX7)*4+1
      JJ1= RANDU(IX7)*2+1
      JF= J1+2
      DO 100 J=1,J1
      NPART2(L1,J)= RANDU(IX1)*9+ 10
      K23= NPART2(L1,J)
      LQ(K23)= (RANDU(IX2)*3+2)*BN(L1,1)
      BN(K23,1)= LQ(K23)
  100 CONTINUE
      DO 901 JK= L31,L1
      AN(JK)=0.
      DO 902 IK=1,6
      AN(JK)= AN(JK)+BN(JK,IK)
  902 CONTINUE
  901 CONTINUE
C     COUNTER
      M=0
      DO 998 JKL= L31,L1
      IF(AN(JKL))998,998,997
  997 M=M+1
  998 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EVENTS(I,NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMON ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPPDCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      GO TO (1,2,3,4),I
    1 CALL ARRVL(NSET)
      RETURN
    2 CALL REPORT(NSET)
      RETURN
    3 CALL STOCK(NSET)
      RETURN
    4 CALL PROCES(NSET)
      RETURN
      END
```

```
      SUBROUTINE REPORT(NSET)
C     THIS IS AN EVENT SUBROUTINE; THIS IS CALLED BY GASP AT THE SERVICING
C       PERIOD .  IT DETERMINES QUANTITY OF EACH PART TO BE  RELEASED TO
C      THE SHOP. IF THE REQUIRED MACHINE IS AVAILABLE IT SCHEDULES THE
C      PART,CTHERWISE IT PUTS THE PART IN A QUEUE BEFORE THE REQUIRED
C      MACHINE GROUP.
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMCN ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     225),IX(8),SUMA(25,5),ATTRIB(10)
      COMMCN AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMCN ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGO
     1,NOPART,NPROCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      DO 100 K=1,40
      DO 100 I=1,20
100   ANB(K,I)= 0.
      DO 150 NOPART= L31,L1
150   ANX(NOPART)= PROD(NOPART,7)
      DO 500 NOPART= L21,L22
      IF(ANX(NOPART).GT.0.)GO TO 165
      GO TO 500
165   DO 501 I=1,3
      NPART4(NOPART,I)= RANDU(IX2)*4+10
      K42=  NPART4(NOPART,I)
      LQ(K42)= (RANDU(IX3)*1+2)*ANX(NOPART)
      NPART5= NOPART-14
      ANB(K42,NPART5)= LQ(K42)
501   CONTINUE
 500  CONTINUE
      DO 600 K= L31,L32
      ANC(K)= 0.
      DO 601 I=1,10
 601  ANC(K)= ANC(K)+ ANB(K,I)
 600  CONTINUE
      DO 700 I= L31,L32
700   ANX(I)= ANC(I)+ ANX(I)
      DO 701 I= L31,L1
      PROD(I,7)= ANX(I)
      PROD(I,4)= ANX(I)*ATIME(I)
      IF(ANX(I)) 148,149,149
 148  ISTF(I)= -ANX(I)
```

```
      GO TO 701
  149 ISTF(I)= 0
  701 CONTINUE
      DO 1678 I= L31,L1
      IF(ANX(I))1679,1679,1680
 1679 PROD(I,7)=0.
      PROD(I,4)=0.
      GO TO 1678
 1680 PROD(I,7)= PROD(I,7)
      PROD(I,4)= PROD(I,4)
 1678 CONTINUE
      IKTT= 0
      DO 1900 I= L31,L1
      IF(ANX(I))1900,1900,1902
 1902 IKTT=IKTT+1
 1900 CONTINUE
      DETERMINE THE LOT #
      NPRDCT= NPRDCT+1
      LEV= NPRDCT
      PR1= 0.
      DO 2100 NOPART= L31,L1
      PR1= PR1+ PROD(NOPART,4)
 2100 CONTINUE
      PR2= PR1/5.
      BNX(LEV)=  ATTRIB(1) - .5*ASM + PRDUCE *PR2
  12  PRINT 1200
 1200 FORMAT('  CHECK ***')
      JBPRT(LEV)= IKTT
      IZ1(LEV)= MPROD
      LEV1= LEV-1
      IF(LEV.EQ.1) IZ2(LEV)= MPROD
      IZ2(LEV)= IZ1(LEV)- IZ1(LEV1)
      IQ(LEV)= 0
      NOPART= L31
 1000 ATTRIB(5)= NOPART
      ATTRIB(10)= NPRDCT
      ATTRIB(4)= PROD(NOPART,4)
      PROD(NOPART,10)= NPRDCT
      NG= MG(NOPART)
      IF(PROD(NOPART,4).LE.0.) GO TO 99
      ATTRIB(3)= TNOW- .5*ASM
       ATTRIB(8)= NG
      ATTRIB(7)= PROD(NOPART,7)
      ATTRIB(9)= PROD(NOPART,9)
C     SCHEDULE THE PARTS ON MACHINES
      IF(NAVL(NG))12,30,25
 25   XPROC= MACH(NG)-NAVL(NG)
      CALL TMSTAT(XPROC,TNOW,NG,NSET)
      NAVL(NG)= NAVL(NG)-1
      ATTRIB(1)= TNOW+ATTRIB(4)
```

```
      ATTRIB(2)= 4.
      CALL FILEM(1,NSET)
      K= MACH(NG)
C     STORAGE FOR PART COMPLETION TIME
       DO 44 IK=1,K
      IF(TNOW-SAV(IK,NG))44,43,43
 44   CONTINUE
      GO TO 12
 43   TFMCA(NG)= ATTRIB(1)
      SAV(IK,NG)= ATTRIB(1)
      DO 45 IJ= 1,K
      IF(SAV(IJ,NG)-TFMCA(NG))46,45,45
 46   TFMCA(NG)= SAV(IJ,NG)
45    CONTINUE
      CTIME(LEV,NOPART)= ATTRIB(1)
      GO TO 65
C     PARTS MUST WAIT FOR AVAILABLE MACHINE
 30   ATTRIB(1)= TNOW
      ATTRIB(2)= 1.
      NGQ= NG+1
      CALL FILEM(NGQ,NSET)
      CALCULATE EXPECTED COMPLETION TIME FOR THE PART
      CTIME(LEV,NOPART)= TFMCA(NG)+ ATTRIB(4)
 65   IF(CTIME(LEV,NOPART).GT.IC(LEV))IC(LEV)= CTIME(LEV,NOPART)
 99   IF(NOPART.EQ.L1) GO TO 1001
      NOPART= NOPART+1
      GO TO 1000
1001  ATTRIB(1)= TNOW +ASM
      ATTRIB(2)= 2.
      CALL FILEM(1,NSET)
      DO 1333 NOPART= L31,L1
      PROD(NOPART,4)= 0.
1333  PROD(NOPART,7)= 0.
      RETURN
      END
```

```
      SUBROUTINE ARRVL(NSET)
C      SUBROUTINE FOR ARRIVAL OF PRODUCT EVENT.
      THIS IS AN EVENT SUBROUTINE. IT CALL SUBROUTINE PROCT TO DETERMINE
      ATTRIBUTES OF THE ORDER. IT STORES THE ATTRIBUTES IN EVENT FILE ,
      AND SCHEDULE ARRIVAL OF NEXT ORDER.
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(10),MG(35),MAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TEMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMON ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPROCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      CALL PRODCT(NSET)
      DO 200 NOPART= L21,L22
      IF(AN(NOPART)) 195,195,196
195   AN(NOPART)= RANDU(IX4)*2. +2.
      GO TO 200
196   AN(NOPART)= AN(NOPART)
200   CONTINUE
      DETERMINE CUM. PART REQUIREMENTS FOR END-ITEM
      DO 105 NOPART= L31,L1
      ATIME(NOPART)= ERLANG(2,2)
      PROD(NOPART,7)= PROD(NOPART,7)+ AN(NOPART)
      PROD(NOPART,4)= PROD(NOPART,7)*ATIME(NOPART)
105   CONTINUE
      DO 106 NOPART= L31,L1
      DO 107 I= 1,10
107   ATTRIB(I)= PROD(NOPART,I)
106   CONTINUE
      FILE ATTRIBUTES OF THE ORDER IN EVENT FILE, SCHDULE ARRIVAL OF NEXT ORDER
      DO 676 I=1,10
676   ATTRIB(I)= PROD(10,I)
      CALL FILEM(1,NSET)
      RETURN
      END
```

```
      SUBRCUTINE STOCK(NSET)
      COMMCN ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMCN ENO(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     225),IX(8),SUMA(25,5),ATTRIB(10)
      COMMCN AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40.12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IO(40)
     3,LQ(40),MACH(10),MC(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMCN ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPRDCT,NPLATE,PR1,PR2,PRDUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      DO 500 I= L31,L32
      IST(I)= PROD(I,7)
      IST(I)= IST(I)+ ISTF(I)
  500 CONTINUE
      DO 600 NOPART= L31,L1
      PROD(NOPART,7)= PROD(NOPART,7)-IST(NOPART)
  600 CONTINUE
      DO 601 J= L31,L1
      DO 602 I= 1,10
  602 ATTRIB(I)= PROD(J,I)
  601 CONTINUE
      PROD(10,1)= TNOW+ 40.
      PROD(10,2)=3.
      DO 676 I=1,10
  676 ATTRIB(I)= PROD(10,I)
      CALL FILEM(1,NSET)
      RETURN
      END
```

```
      SUBROUTINE SELECT(NSET)
C     THIS SUBROUTINE SELECTS THE NEXT PART TO BE PROCESSED BY ANY MACHINE WHEN
C     MORE THAN ONE PARTS ARE WAITING IN THE QUEUE. ALL PARTS ARE ARRANGED
C     IN THE EVENT FILE AS PER DUE-DATE. IT SELECTS THE PART WITH EARLIET DUE-
C     DATE. IN CASE OF TIES, IT SELECTS THE PART WITH SHORTEST PROCESSING TIME.
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLF(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(30),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TEMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMON ASM,BF,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPRDCT,NPLATE,PR1,PR2,PROUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      DO 668 I=1,10
  668 PROC(NOPART,I)= ATTRIB(I)
       NGQ= NG+1
      MFE1= MFE(NGQ)
C     REMOVE THE PART WITH EARLIEST DUE-DATE
      CALL REMOVE(MFE1,NGQ,NSET)
      IF(NQ(NGQ)-1)49,44,40
C     TWO PARTS WAITING IN THE QUEUE
   44 DO 71 I=1,10
   71 STORE(I)= ATTRIB(I)
      NO1= ATTRIB(5)
      PT1= ATTRIB(4)
      MFE2= MFE(NGQ)
      CALL REMOVE(MFE2,NGQ,NSET)
      NO2= ATTRIB(5)
      PT2= ATTRIB(4)
C     DETERMINE WHICH PART HAS THE LEAST NO OF JOBS
      IF(PT1 - PT2)72,72,70
   70  DO 41 I=1,10
      SAVE(I)= ATTRIB(I)
   41  ATTRIB(I)= STORE(I)
      CALL FILEM(NGQ,NSET)
      DO 43 I=1,10
   43  ATTRIB(I)=SAVE(I)
      RETURN
   72 CALL FILEM(NGQ,NSET)
      DO 73 I=1,10
   73 ATTRIB(I)= STORE(I)
      RETURN
```

```
C     MORE THAN TWO PARTS WAITING IN QUEUE
 40   NO1= ATTRIB(5)
      PT1= ATTRIB(4)
      DO 201 I= 1,10
 201  STORE(I)= ATTRIB(I)
      K=NQ(NGQ)
      DO 202 J=1,K
      MFE3= MFE(NGQ)
      CALL REMOVE(MFE3,NGQ,NSET)
      NO2= ATTRIB(5)
      PT2= ATTRIB(4)
C     DETERMINE WHICH PART TAKES MIN TIME FOR COMPLETION
      IF(PT1 - PT2)203,203,204
      STORE PART NOT TO PUT ON MACHINE NOW
 203  DO 75 L=1,10
  75  CN(J,L)= ATTRIB(L)
      GO TO 202
 20   DO 206 JK=1,10
      CN(J,JK)= STORE(JK)
  206 STORE(JK)= ATTRIB(JK)
      NO1= NO2
 202  CONTINUE
C     PUT REMAINING PARTS IN PROPER MACHINE QUEUE
      DO 207 J= 1,K
      DO 76 KIM=1,10
   76 ATTRIB(KIM)= CN(J,KIM)
  207 CALL FILEM(NGQ,NSET)
      DO 208 I= 1,10
 208  ATTRIB(I)= STORE(I)
 49   RETURN
      END
```

```
      SUBROUTINE PROCES(NSET)
C     THIS SUBROUTINE SCHEDULES THE PARTS AS PER MACHINE AVAILABILITY. IT CALLS
C     SUBROUTINE SELECT TO SELECT A PART FROM A QEUE. IT MARKS END OF PROCESSING
C     FOR ANY PART(ORDER). IT STARTS COLLECTING STATISTICS ON THE PROCESSED
C     ORDERS. IT ALSO CALLS SUMMARY TO SUMMARISE THE STATISTICS UPTO A PRE-
C     DETERMINED NO. OF END-ITEM ORDERS(400).
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ENQ(25),INN(25),JCELLS(25,25),KRANK(25),MAXNQ(25),MFE(25),
     1MLC(25),MLE(25),NCELLS(25),NQ(25),PARAMS(20,4),QTIME(25),SSUMA(25,
     25),IX(8),SUMA(25,5),ATTRIB(10)
      COMMON AN(40),ANB(40,20),ANC(40),ANX(40),ATIME(40),BJOB(40),BNX(40
     1),BN(40,12),CN(30,12),CTIME(40,40),JBPRT(40),INV(40),IST(40),ISTF(
     240),IZ1(25),IZ2(25),IZ3(25),IZ4(25),IZ5(25),IZ6(25),IZ7(25),IQ(40)
     3,LQ(40),MACH(10),MG(35),NAVL(30),NPART2(40,10),NPART3(40,10),NPART
     44(40,10),PROD(40,10),PTIME(40),SAV(5,25),SAVE(10),STORE(10),TFMCA(
     525),TJOB(40,20),TSHOP(40)
      COMMON ASM,BT,CT,DDATE,IKTT,IMM,ISS,JFILE,JKR,KTPRD,M,MPROD,NG,NGQ
     1,NOPART,NPRDCT,NPLATE,PR1,PR2,PRODUCE,ST,TISYS,TLATE,XPROC,IX1,IX2,
     2IX3,IX4,IX5,IX6,IX7,L1,L21,L22,L31,L32,MCG,NMCEG,NSUMP1,NSUMP2
      DIMENSION NSET(12,1)
      NOPART= ATTRIB(5)
      NG= ATTRIB(8)
      NPRDCT= ATTRIB(10)
      LEV= NPRDCT
      TISYS= ATTRIB(1)-ATTRIB(3)
      TSHOP(LEV)= TSHOP(LEV)+TISYS
      JBPRT(LEV)= JBPRT(LEV)-1
      SEEE IF ALL PARTS OF THE LEVEL ARE COMPLETE
      IF(JBPRT(LEV))10,10,20
      CALCULATE A STATISTICS ON COMPLETED LEVEL
   10 DIFF= TNOW - IQ(LEV)
      CT= CT+ DIFF
      IQ(LEV)= 0
      TSHOP(LEV)= TSHOP(LEV)/IZ2(LEV)
      CALL COLECT(TISYS,1,NSET)
      CALL COLECT(TSHOP(LEV),2,NSET)
      CT= TISYS- TSHOP(LEV)
      CALL COLECT(CT,3,NSET)
      TLATE= TNOW- BNX(LEV)
      CALL COLECT(TLATE,4,NSET)
      CT=0.
      KTPRD= KTPRD+1
      PRINT 150,KTPRD
  150 FORMAT('    KTPRD=',I5)
      IF(KTPRD.EQ.NSUMP2) GO TO 3
      GO TO 20
3     DO 24 NG=1,NSTAT
      XPROC= MACH(NG)-NAVL(NG)
```

```
24    CALL TMSTAT(XPROC,TNOW,NG,NSET)
      NEP=1
      MSTOP= -1
      NORPT=0
      RETURN
  20  NGQ= NG+1
      IF(NQ(NGQ))30,30,40
  30  XPROC= MACH(NG) -NAVL(NG)
      CALL TMSTAT(XPROC,TNOW,NG,NSET)
      NAVL(NG)= NAVL(NG)+ 1
      WT= ATTRIB(1)- (ATTRIB(3)+ ASM*.5)
      RETURN
C     SELECT NEXT PART TO BE SCHEDULED
 40   CALL SELECT(NSET)
      NOPART= ATTRIB(5)
C     SCHEDULE NEXT PART
      ATTRIB(1)= TNOW+ ATTRIB(4)
      ATTRIB(2)= 4.
      CALL FILEM(1,NSET)
      K= MACH(NG)
C     STORAGE FOR PART COMPLETION TIME
      DO 44 IK= 1,K
      IF(TNOW-SAV(IK,NG))44,43,41
  44  CONTINUE
  41  CALL ERROR(4,NSET)
43    TFMCA(NG)= ATTRIB(1)
      SAV(IK,NG)= ATTRIB(1)
C     UPDATE TIME FIRST MACHINE AVAILABLE
      DO 45 IJ= 1,K
      IF(SAV(IJ,NG)-TFMCA(NG))46,45,45
46    TFMCA(NG)= SAV(IJ,NG)
45    CONTINUE
      CTIME(LEV,NOPART)= ATTRIB(1)
C     UPDATE EXPECTED COMPLETION TIME OF LEVEL
      IF(CTIME(LEV,NOPART).GT.IQ(LEV))IQ(LEV)= CTIME(LEV,NOPART)
      RETURN
      END
```

AN INVESTIGATION ON THE EFFECTS OF RELEASE DATE IN AN MULTI-PRODUCT ASSEMBLY SHOP

by

BASUDEB SARKAR

B. Tech. (Hons. Electrical), Indian Institute of Technology
Kharagpur, India, 1965

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1969

The objective of this thesis was to simulate an assembly shop manufacturing a variety of end-items, requiring usage of certain sub-parts and detail parts in an inconsistent fashion. The simulation model was programmed in FORTRAN IV utilising GASP II (General Activity Simulation Program) to study the effects of various release dates of orders to the shop on the overall performance of the system.

The simulations were performed assuming an exponential arrival of end-item orders. The common parts required by orders were identified and stored in a part file till the release date, at the end of which orders were released to the shop. They were processed with an exponential service time distribution by a group of five machines. The experiment was performed using one priority dispatching rule: "process the part with the earliest due date; in case of ties, chose the job with shortest operation time." The investigation was performed for different values of release dates. Under the present experimental set up a release date of 30 hours was found to give optimum system performance.

Further investigations should be directed to study the effects of various interacting parameters, viz., arrival and processing time distribution, level of shop utilization and choice of priority dispatching rule in determining suitable release date.